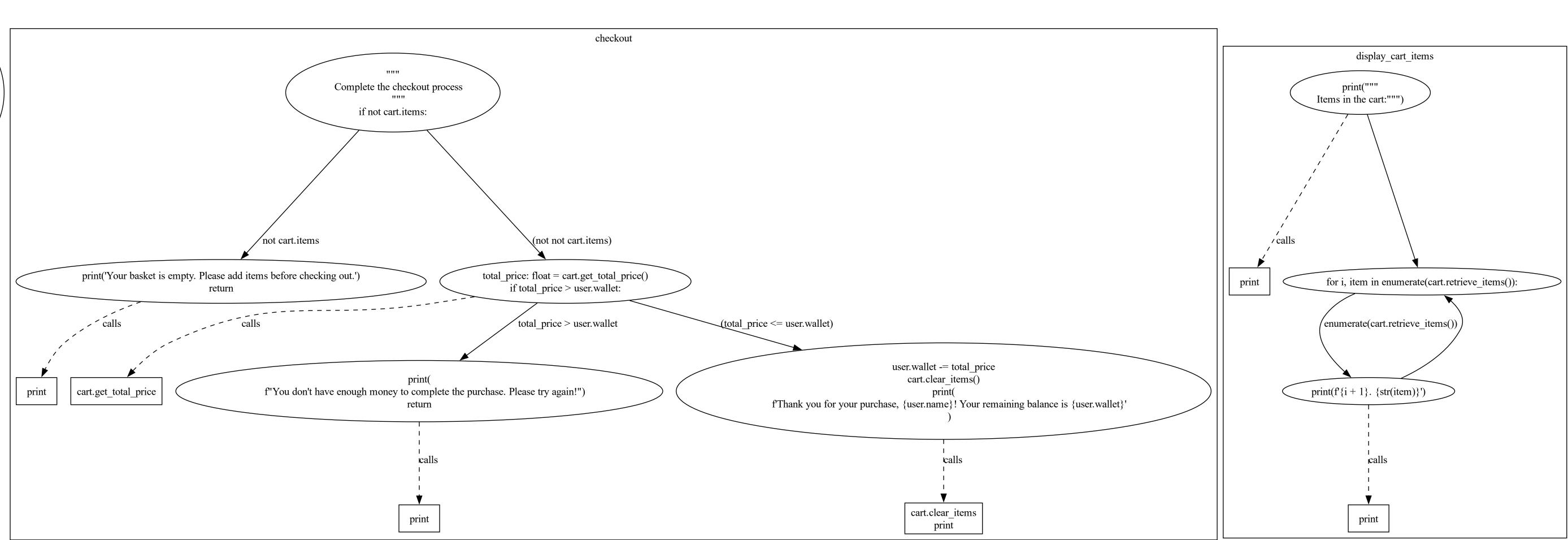
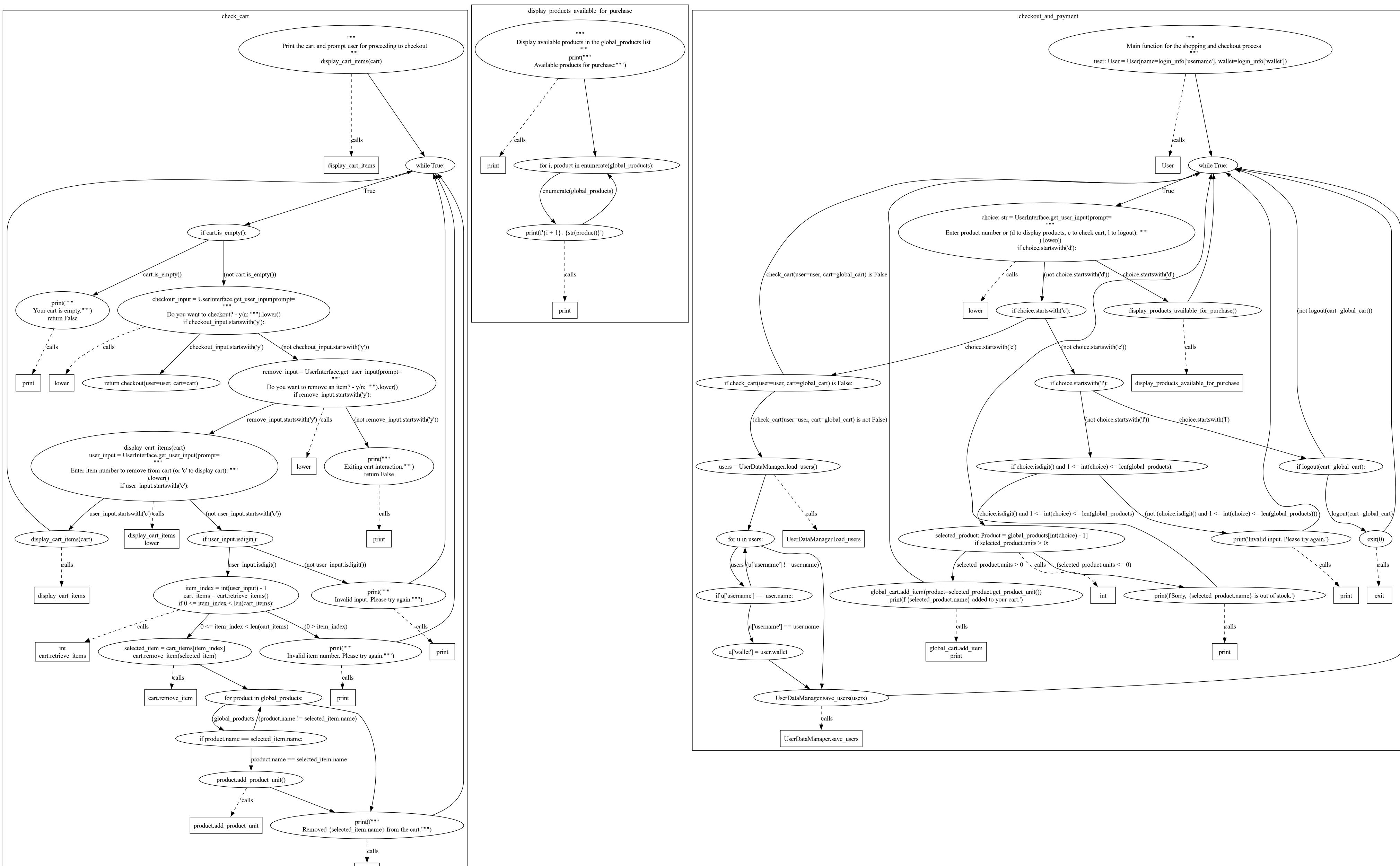
from online\_shopping\_cart.checkout.shopping\_cart import ShoppingCart from online\_shopping\_cart.product.product\_data import get\_products from online\_shopping\_cart.user.user\_interface import UserInterface from online\_shopping\_cart.product.product import Product from online\_shopping\_cart.user.user\_logout import logout from online\_shopping\_cart.user.user import User from online shopping cart.user.user data import UserDataManager global\_products: list[Product] = get\_products() global\_cart: ShoppingCart = ShoppingCart() def checkout(user, cart) ->None:... def display\_cart\_items(cart) ->None:... def check cart(user, cart) ->(None | bool):...

> def display\_products\_available\_for\_purchase() ->None:... def checkout\_and\_payment(login\_info) ->None:...

> > get\_products
> > ShoppingCart





checkout\_process

Main function for the shopping and checkout process

user: User = User(name=login\_info['username'], wallet=login\_info['wallet'])

display\_products\_available\_for\_purchase()

display\_products\_available\_for\_purchase

(not choice.startswith('l'))

choice.startswith('l')

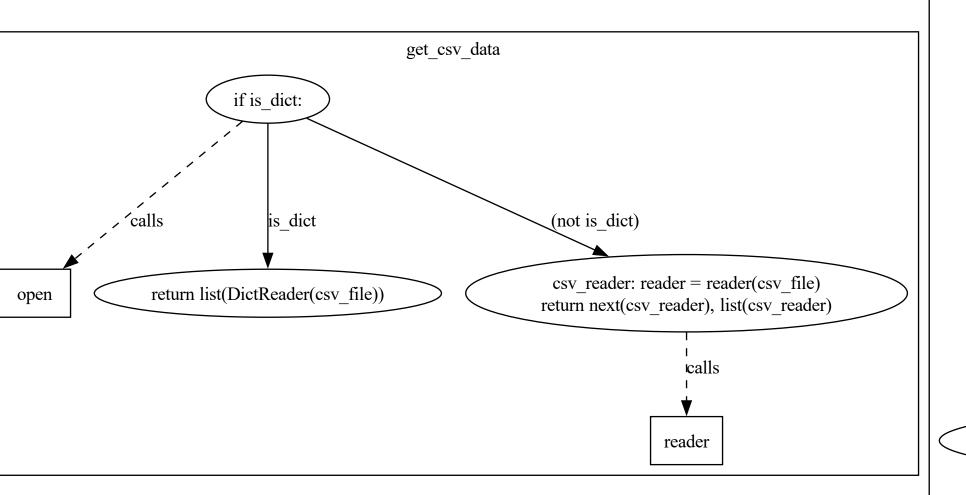
print(f'Sorry, {selected\_product.name} is out of stock.')

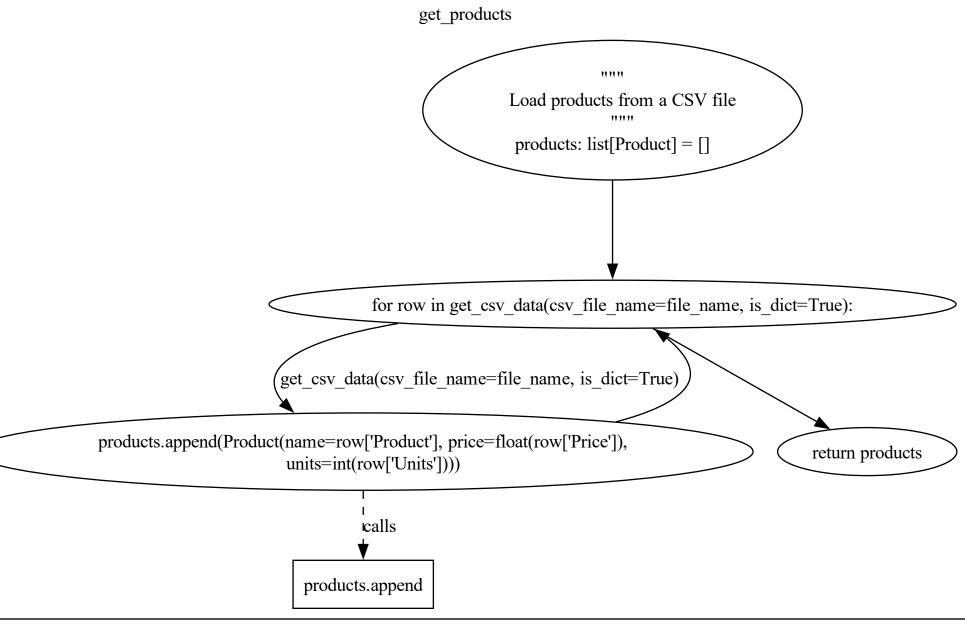
(not logout(cart=global\_cart))

if logout(cart=global\_cart):

print('Invalid input. Please try again.')

from online\_shopping\_cart.product.product import Product
from csv import DictReader, reader
PRODUCTS\_FILE\_PATHNAME: str = './files/products.csv'
def get\_csv\_data(csv\_file\_name=PRODUCTS\_FILE\_PATHNAME, is\_dict=False) ->(list...
def get\_products(file\_name=PRODUCTS\_FILE\_PATHNAME) ->list[Product]:...

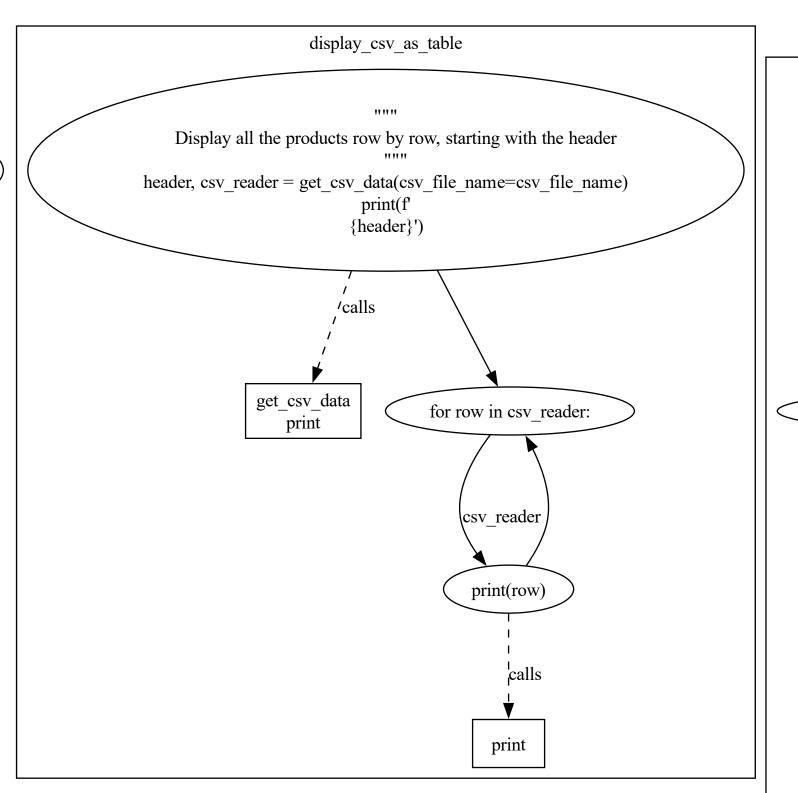


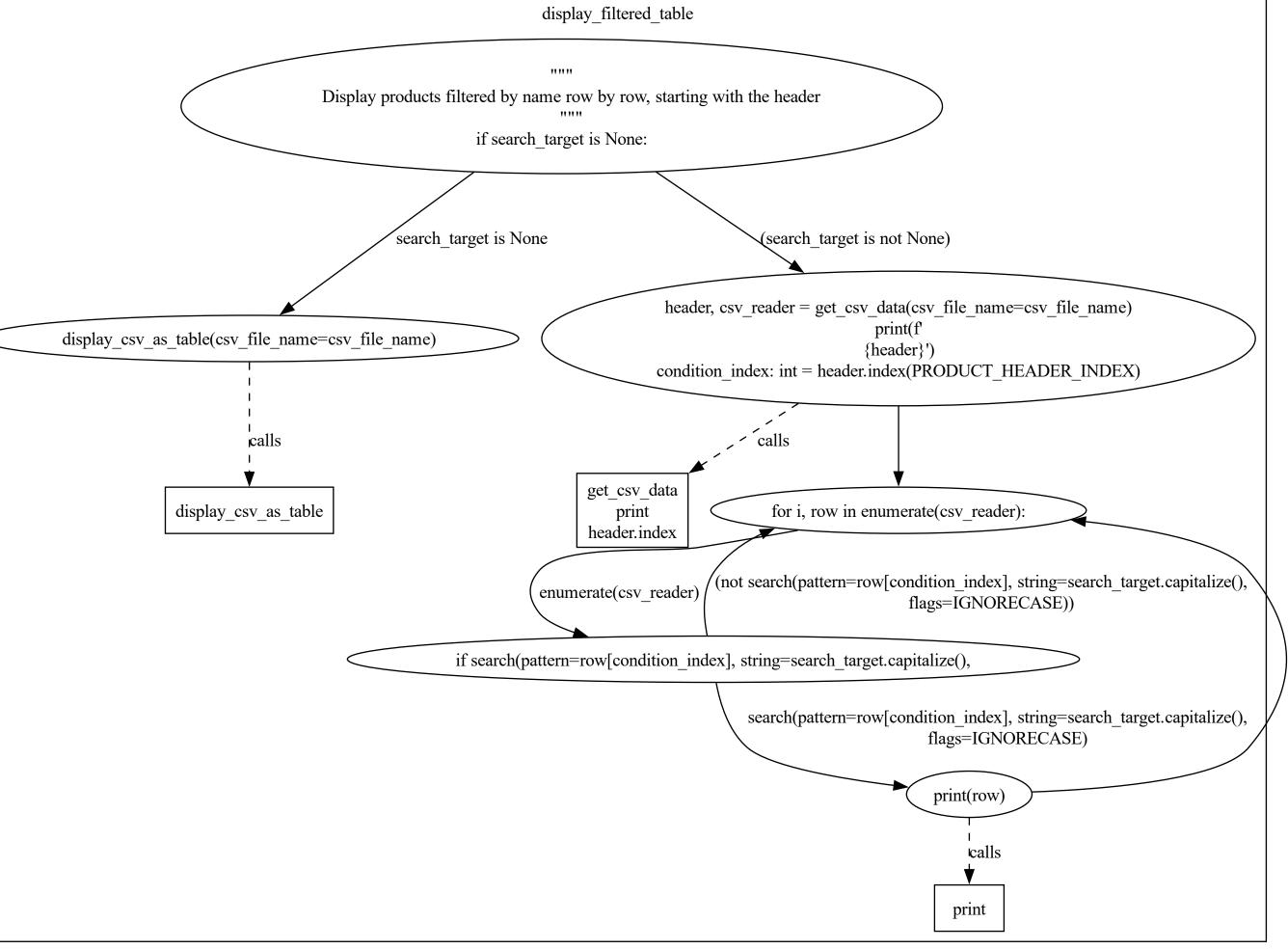


product data

from online\_shopping\_cart.product\_product\_data import get\_csv\_data, PRODUCTS\_FILE\_PATHNAME from re import search, IGNORECASE

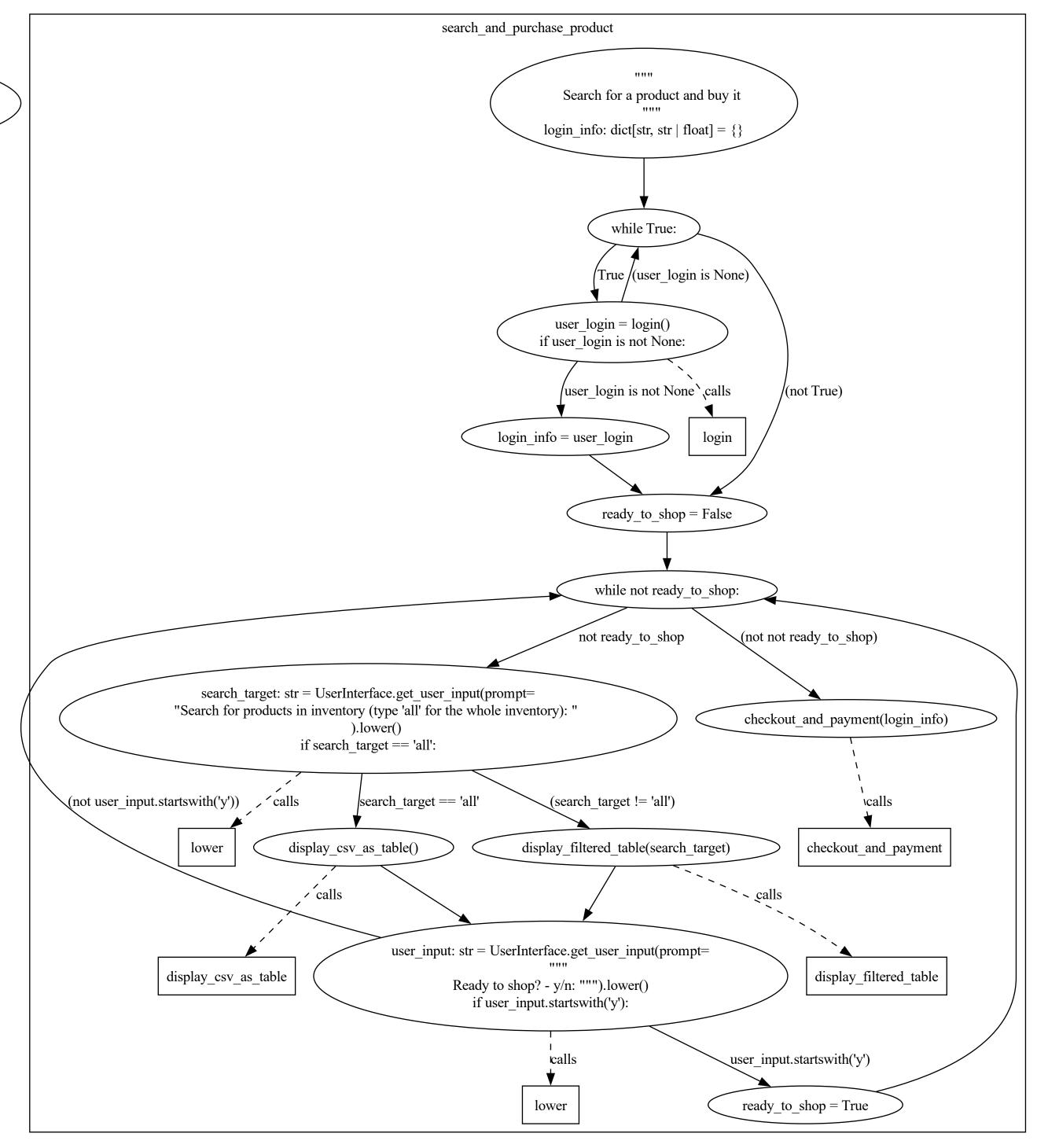
PRODUCT\_HEADER\_INDEX: str = 'Product'
def display\_csv\_as\_table(csv\_file\_name=PRODUCTS\_FILE\_PATHNAME) ->None:...
def display\_filtered\_table(csv\_file\_name=PRODUCTS\_FILE\_PATHNAME,...





product search

from online\_shopping\_cart.product\_search import display\_csv\_as\_table, display\_filtered\_table from online\_shopping\_cart.checkout.checkout\_process import checkout\_and\_payment from online\_shopping\_cart.user.user\_interface import UserInterface from online\_shopping\_cart.user.user\_login import login def search\_and\_purchase\_product() ->None:...



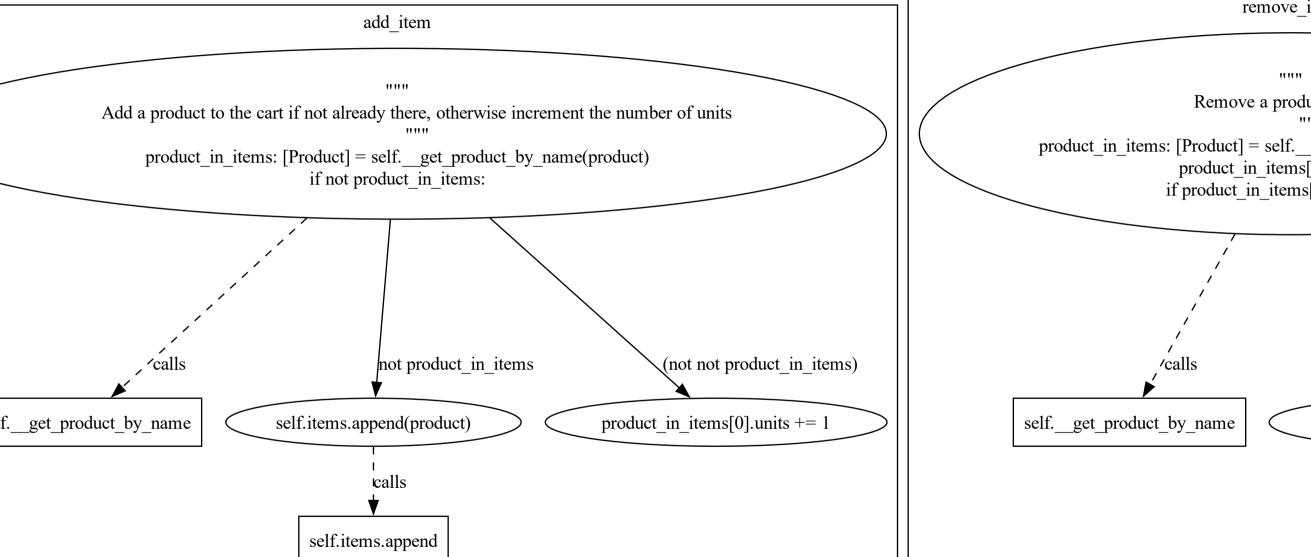
shop\_search\_and\_purchase

from online\_shopping\_cart.product.product import Product ShoppingCart class to represent the user's shopping cart def \_\_init\_\_(self) ->None:... def \_\_get\_product\_by\_name(self, product\_search: Product) ->[Product]:... def add item(self, product) ->None:... def remove item(self, product: Product) -> None:... def retrieve\_items(self) ->list[Product]:... def clear items(self) ->None:... def is\_empty(self) ->bool:... def get\_total\_price(self) ->float:...

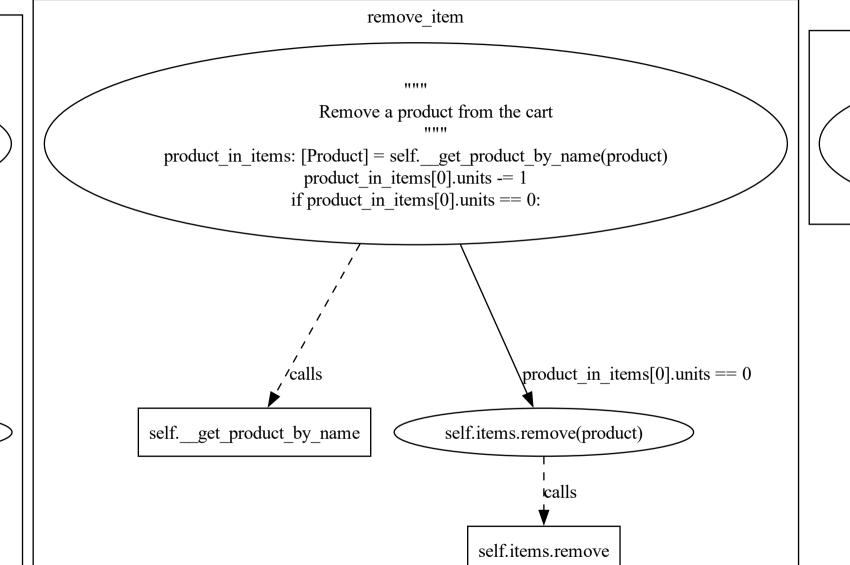
self.items: list[Product] = list()

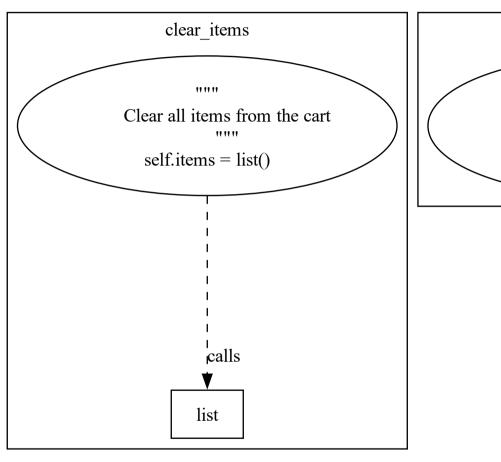
return [product\_i for product\_i in self.items if product\_i.name == product\_search.name]

\_\_get\_product\_by\_name



shopping\_cart





retrieve\_items

11 11 11

return self.items

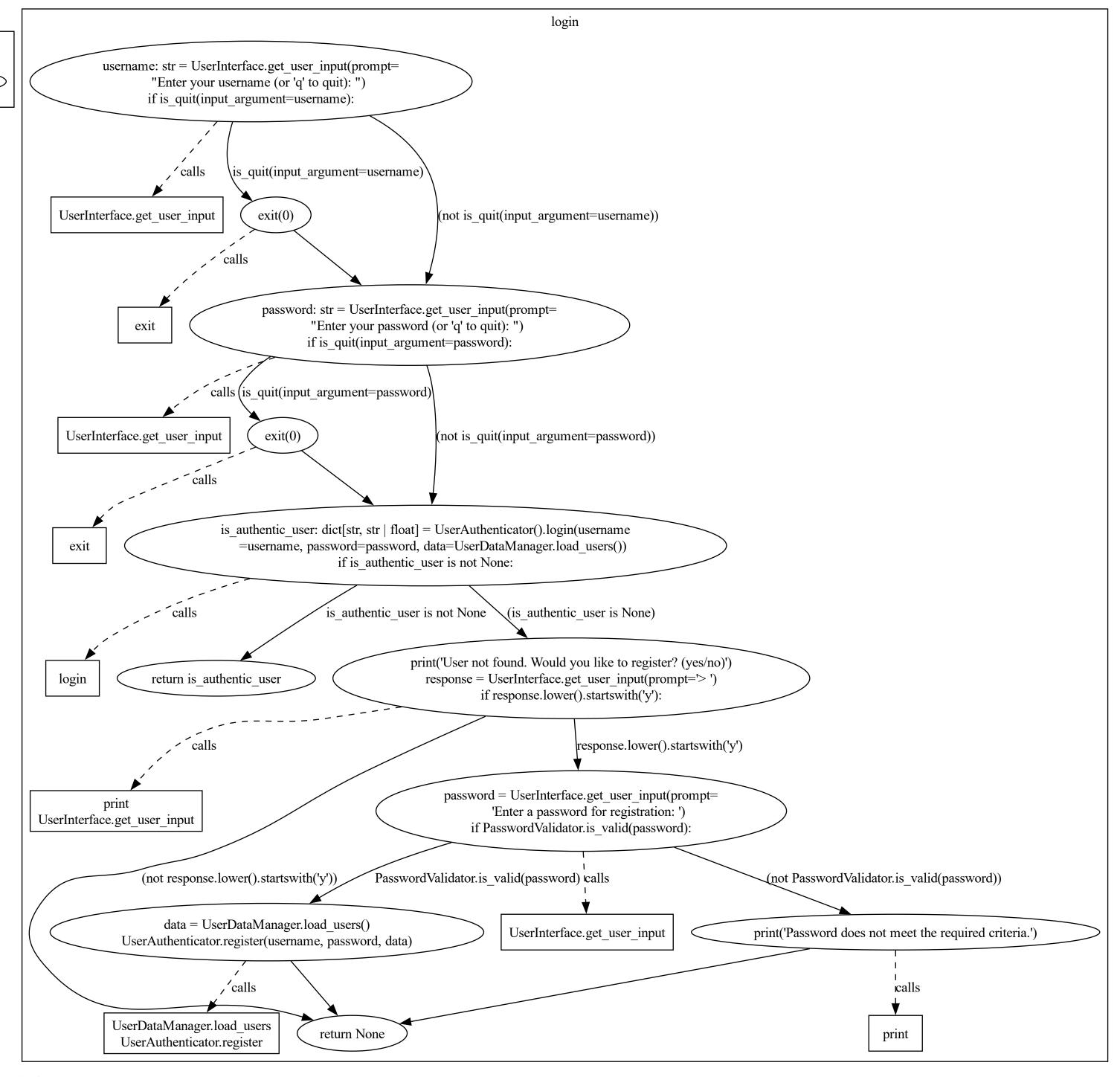
Retrieve the items in the cart

get\_total\_price is\_empty 11 11 11 11 11 11 Calculate the total price of items in the cart Checks if the cart is empty return self.items == [] return sum(item.price \* item.units for item in self.items) from online\_shopping\_cart.user.user\_authentication import UserAuthenticator, PasswordValidator from online\_shopping\_cart.user.user\_interface import UserInterface from online\_shopping\_cart.user.user\_data import UserDataManager def is\_quit(input\_argument: str) ->bool:...

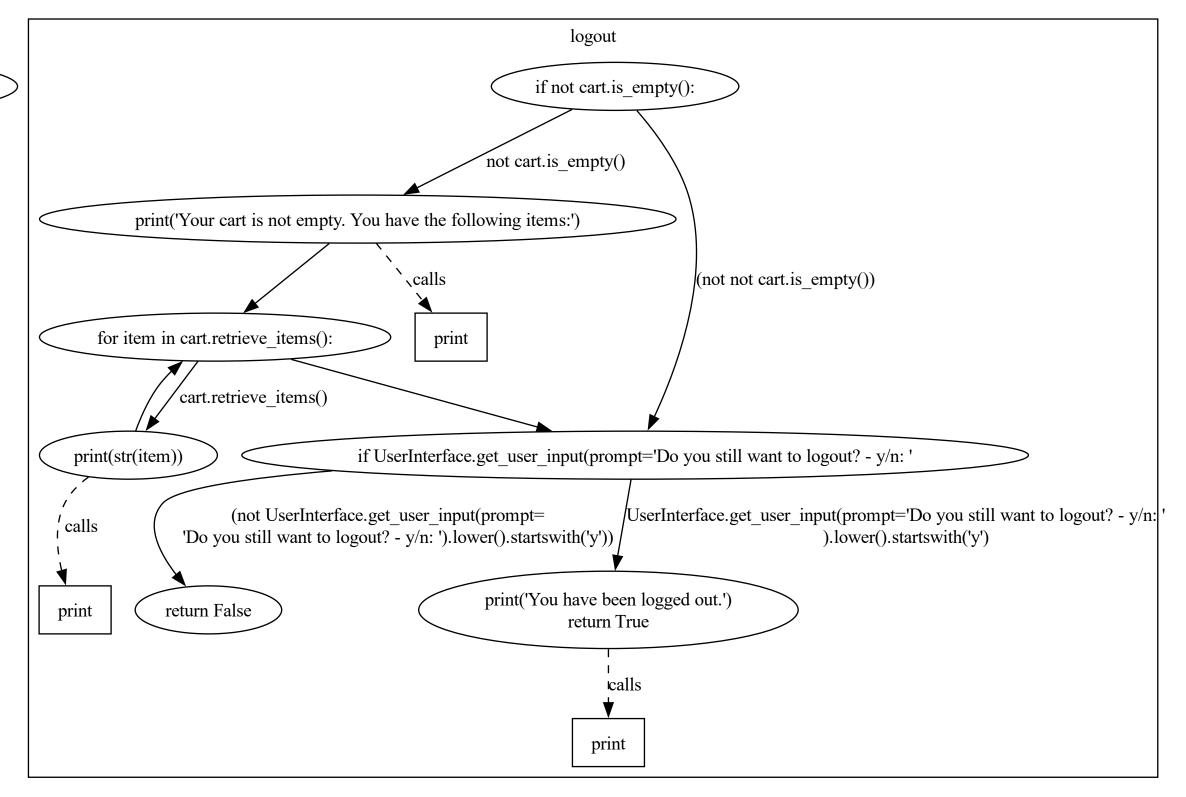
def login() ->(dict[str, str | float] | None):...

is\_quit

return input\_argument.lower() == 'q'



from online\_shopping\_cart.user.user\_interface import UserInterface def logout(cart) ->bool:...



user\_logout