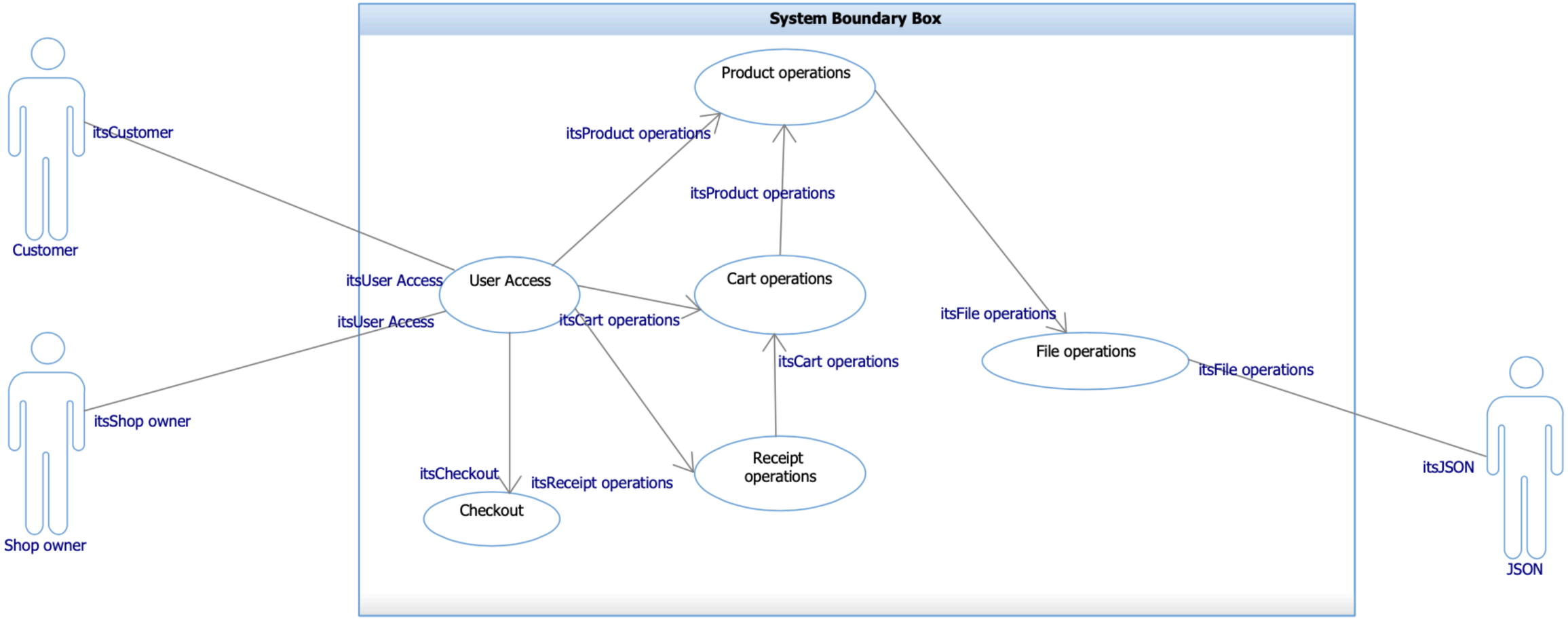
Walkthrough

Shopping application

ROBY VARGHESE 16-12-2021

Use case diagram

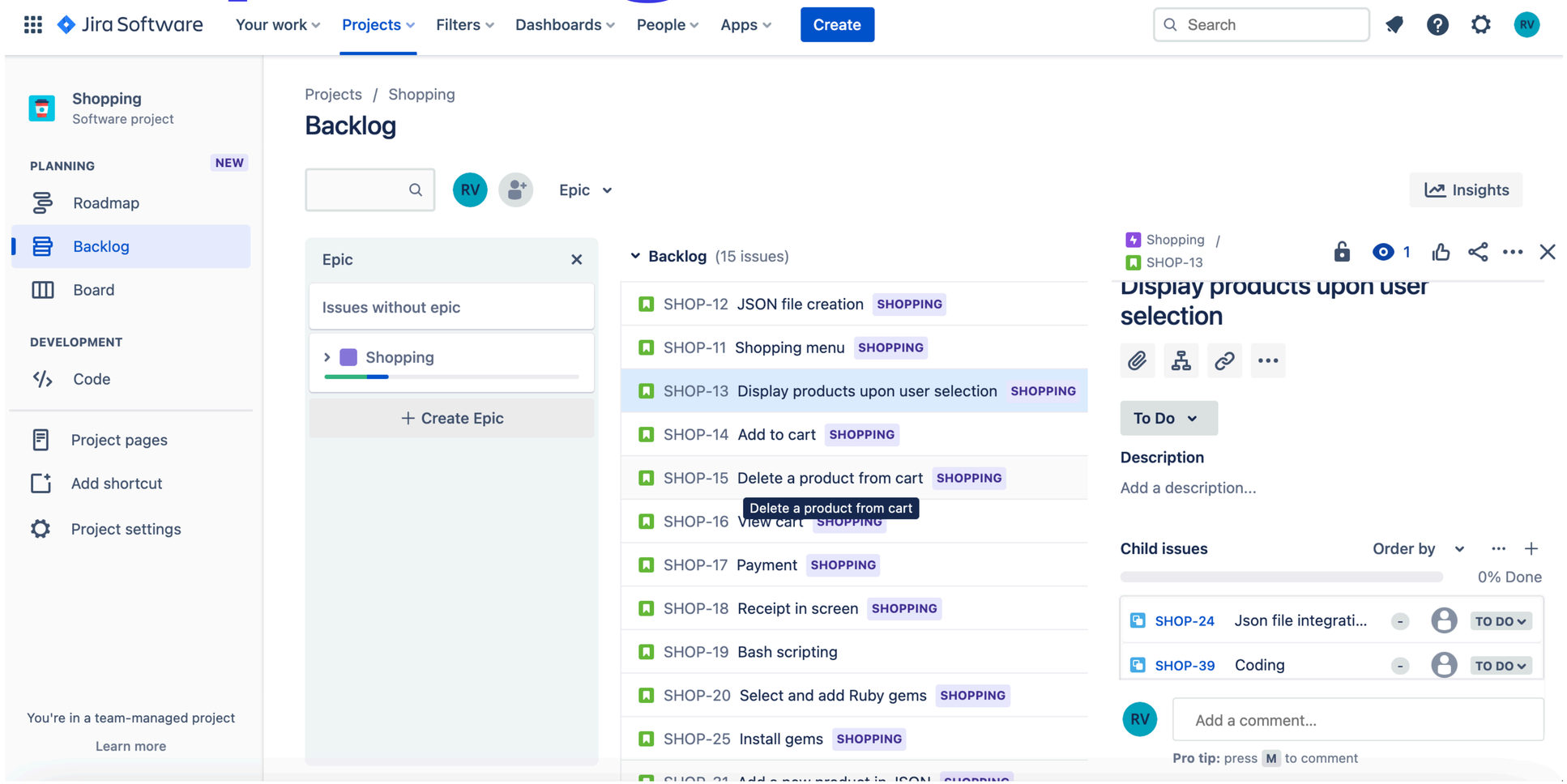


Features

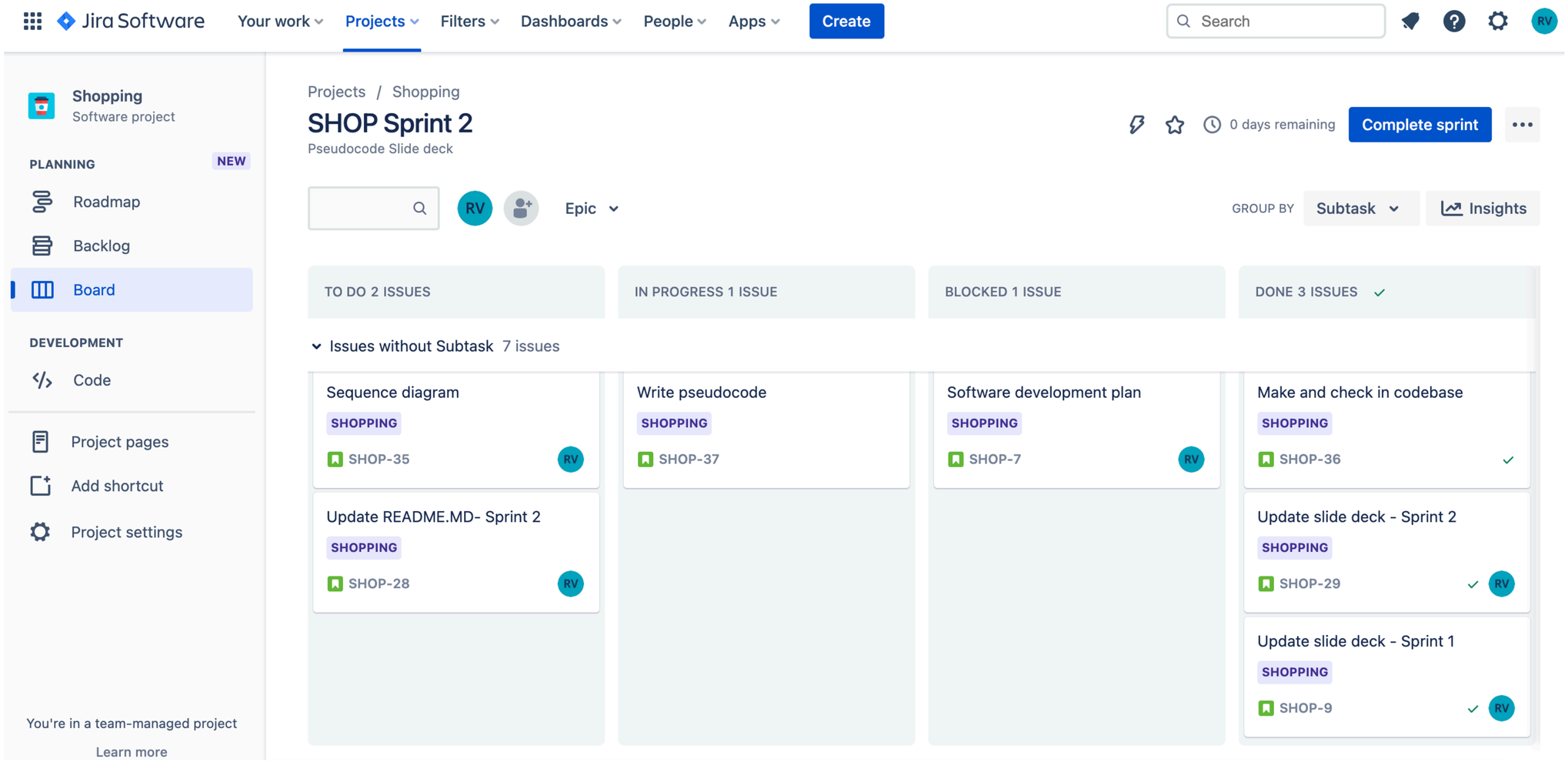
This terminal application can be used by a customer through a command line menu and would be able to do the following actions.

•Show available products, prices and quantities available
•Able to select and add to cart
•Should be able to delete from cart
•Should be able to view the cart
•Should be able to view the receipt in screen
•Should be able to do payment.
•Updating JSON file for quantities and new products
•Two modes should be there one for shop owner and another for customers
•Errors like wrong input and product not in stock should be displayed to the user

The planning



The planning

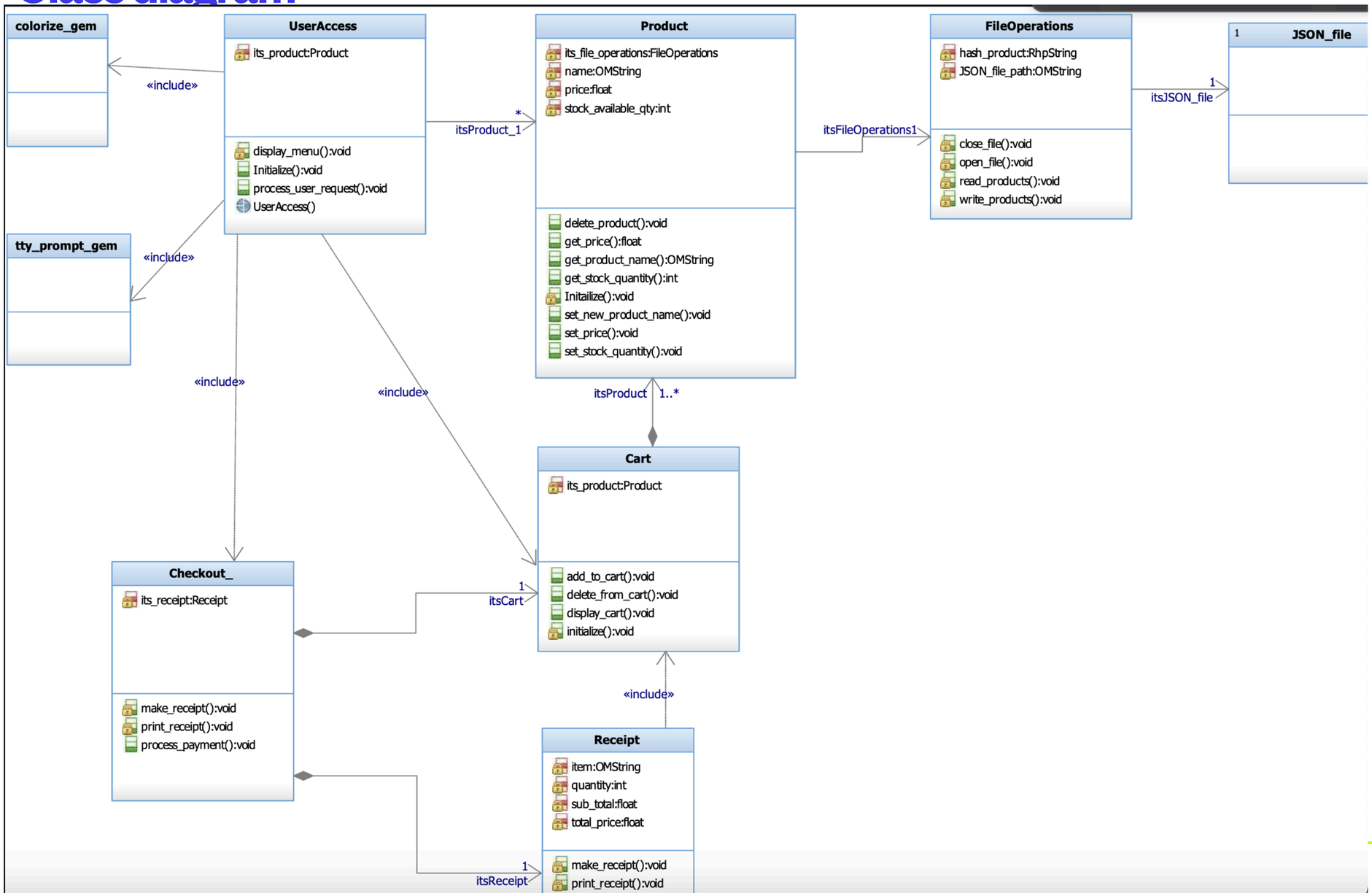


Class analysis

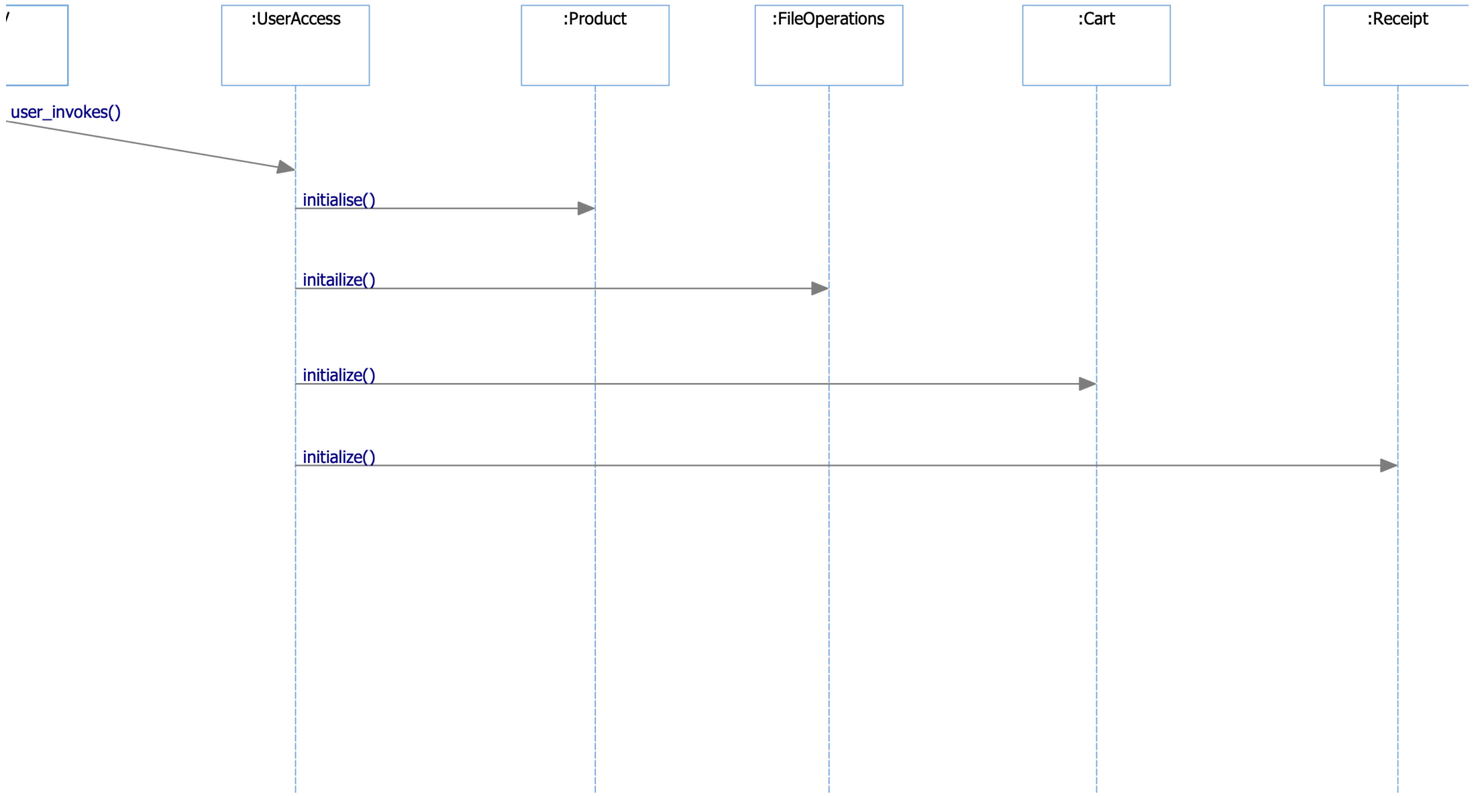
Identified several Classes

- UserAccess
- Product
- Cart
- Checkout
- Receipt
- FileOperations

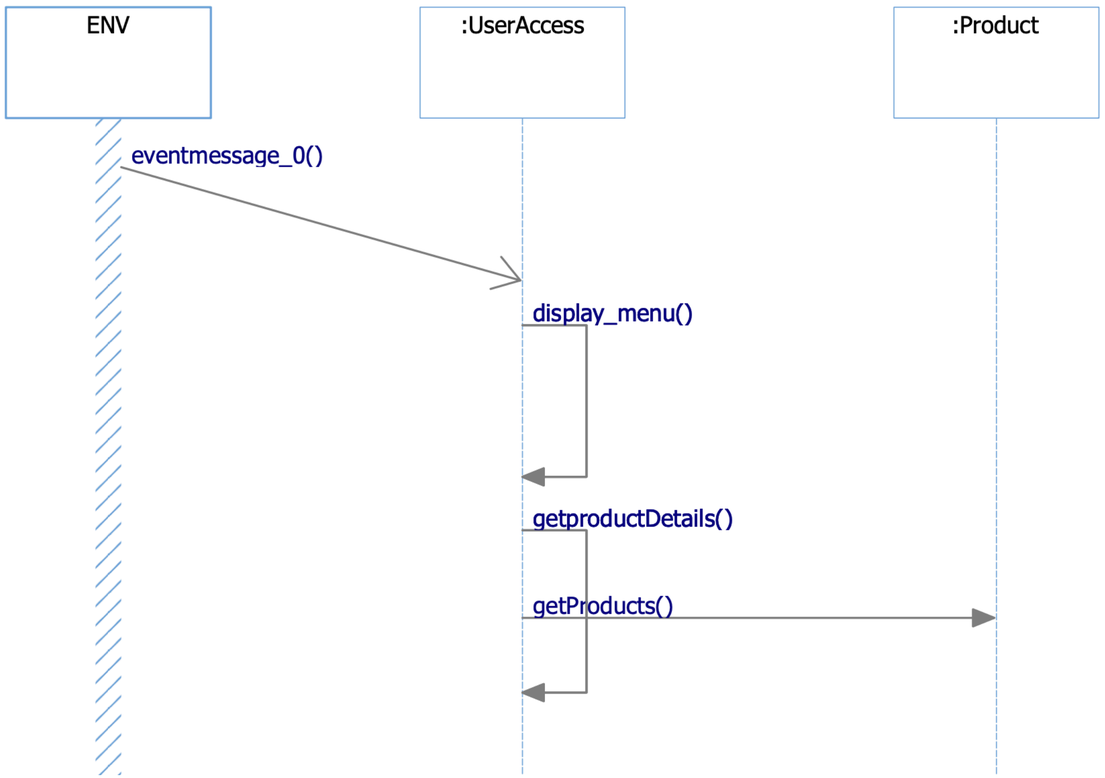
Class diagram



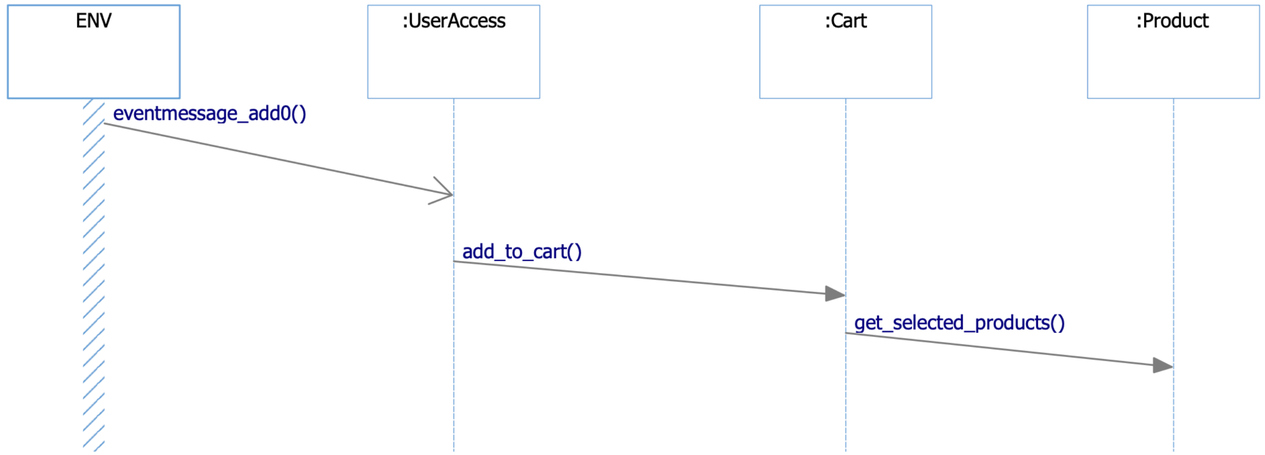
Sequence diagram - initialise



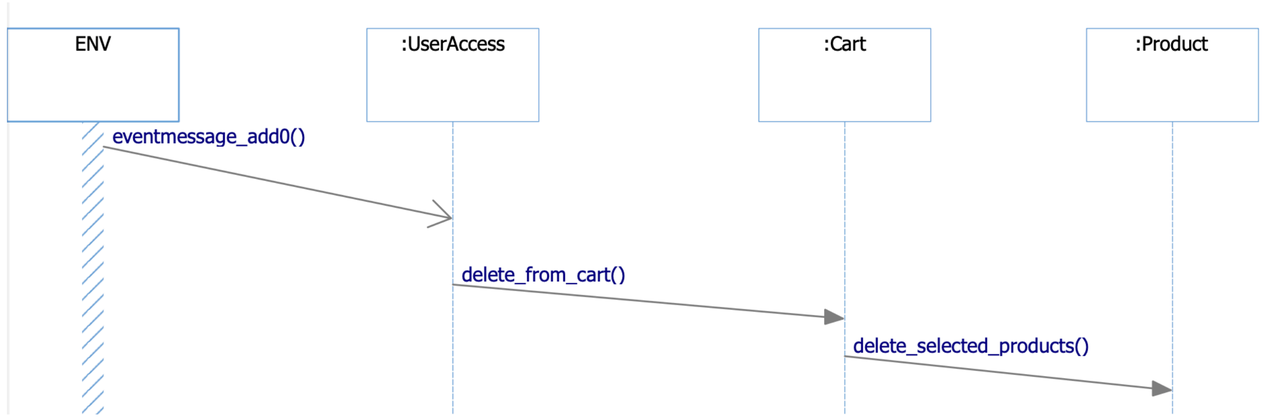
Sequence -look up products



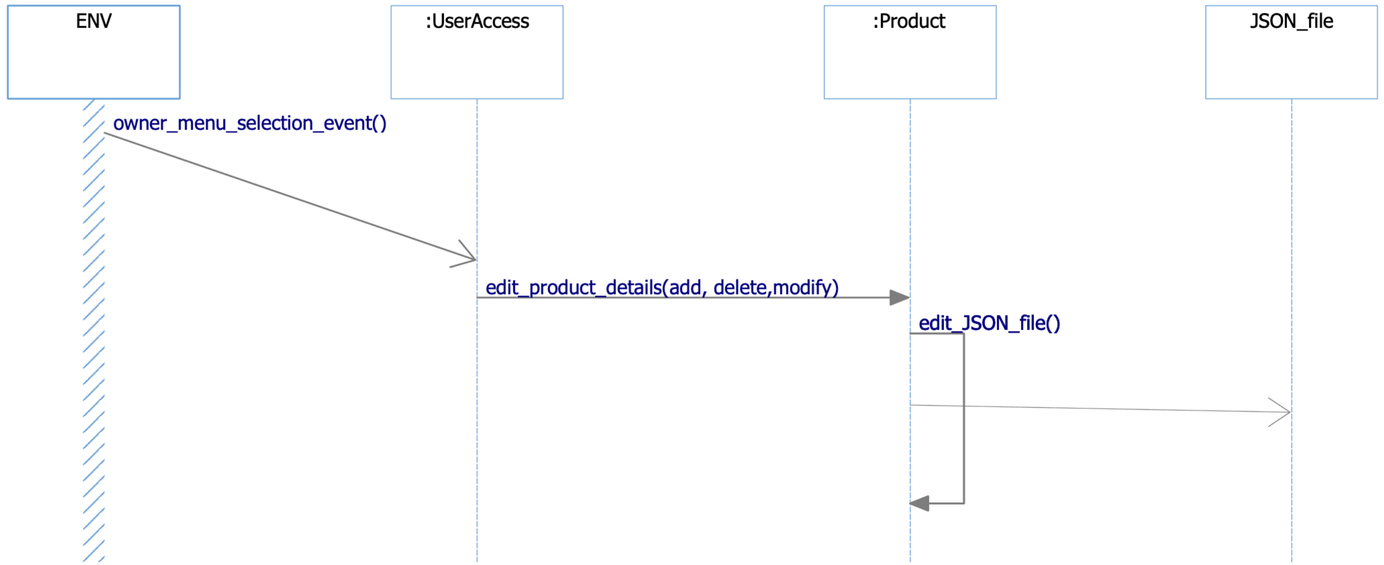
Add a product to cart



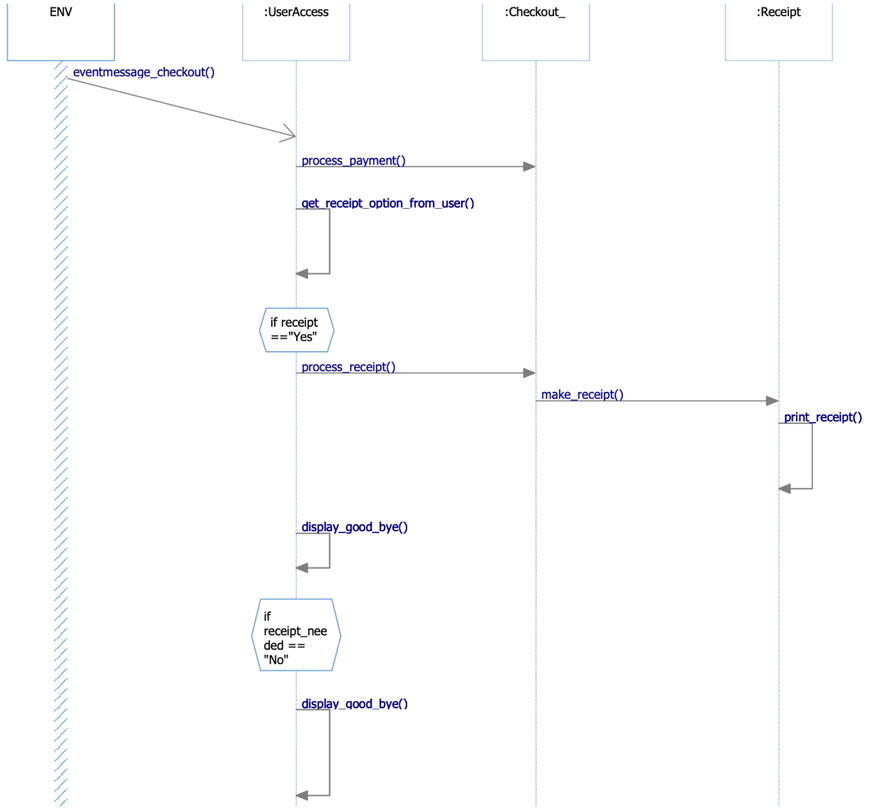
Delete item from cart



Shopping owner edit



Checkout


ENV
:UserAccess
:Checkout_
:Receipt
eventmessage_checkout()
process_payment()
get_receipt_option_from_user()
if receipt
=="Yes"
process_receipt()
make_receipt()
print_receipt()
display_good_bye()
if
receipt_nee
ded ==
"No"
display_good_bye()

Logic -Pseudocode

```
Require gem colorise
Require gem tty-prompt
#Start of Class UserAccess
attr_reader is_customer
  #attributes
  public:
  @@is_customer = false

  #methods
  def checkUser
        #start of the program
        puts (“Are you a customer or Shop owner)
        If customer Then
            set customer is_customer = true
        elsif Shop_owner
            is_customer = false
        else
            throw error message to exception handler
            retry in exception handler
        End
  end
#End of Class UserAccess
```

Pseudo code - initial

```
#Main program
start
# Create new UserAccess object
  create objUserAccess

# create and initialize Product object
  objProduct.initialize(iobjUserAccess.is_customer)
end
```

Pseudo code - initial

```
#Start of Class Product

hashmapmain  products[]
hashmap products_selected[]
    #method initialise()
 def initialise(is_customer)
    {
        if (is_customer)
            {
                display menu for customer
            }
        else
            {
                display menu for shop owner
            }
            display_products()
            display_product_selection_options()
    }
    # method initialize() end
```

Pseudocode

```
    #method display_product_selection_options()
    if (is_customer)
        1."Select 1 for add a specific product to
cart"
        2. "Delete a product from the cart"
        if selected == 1
            puts ("Enter quatity")
            add_to_cart(product_id, quantity)
        elsif selected == 2
            puts ("Enter quatity to be deleted")
            delete_from_cart(product_id,
quantity)
        else
        end
    end
```

Pseudocode

```
    #if shop owner
    elsif (!is_customer)
        1. "Add a new product"
        2. "Delete a product"
        3. "Change price of a product"
        4. "Change stock quantity of a product"
    end
```

Pseudocode

```
#Start of class FileOperatios
#attributes
its_JSON_file
#methods
open_JSON_file()
read_and_map_JSON_to_hash()
write_new_product_to_hash(product array)
close_JSON_file()

#End of class FileOperations
```

Pseudocode

```
#Start of Class Cart
#attributes
itsCheckoutObj
#methods
add_to_cart()
delete_from_cart()
display_checkout_options()
#End of Class Cart
```

Pseudocode

```
#Start of Class Checkout

#attributes
itsReceiptObj
#methods
process_payment()
ask_user_for_receipt()

def process_payment
    "Enter name"
    "Enter card number"
    check_card_type_from_number()
    "Enter anount"
    "Payment succesful"
    ask_user_for_receipt()
end
def check_card_type_from_number
    if cardNUmber firt 4 numbers == 123
        card_type = "Mastercard"
    elsif cardNUmber == 456
        card_type == "Visa"
    else
        "Card not recognised"
    end
def ask_user_for_receipt
    "Do you want a receipt"
    if required
        itsReceiptObj.make_receipt
    else
        "Good bye"
    end
end
#End of Class Checkout
```

Challenges

- Did get trapped in ‘planning’ loop
- Did get trapped in ‘Design fine tuning’ loop

Improvements

- Need to focus on iterative development
- Test based development
- And to code without the fear of failure.