Name: Cody Luhmann

Title: SkateShop

Project Summary: Customers will be able to order/create their own skateboard from a list of all kinds of skate parts that will be held in the inventory class. A customer can select one type of board per order, and can then select the kind of skate deck they want as well as their wheels and trucks. Then once they are done with their order they can checkout and receive their board.

UI Mockup:

I will be using the command line for inputs into my application.

For example for a new order I will type: newOrder, and this should prompt me to first ask what kind of board I want and the screen will display the different types of boards available. Once I choose 1 of the 3, the next line will prompt me to choose my wheels, trucks, and deck and you can display the inventory for said items by either saying displayBoards, displayWheels, and display trucks. The user will only be able to choose one for each order. Then you will be able to check your order, delete your order, checkout, or start a new order

Requirements:

Requirements ID:	Requirements:	Responsibilities:
RQ-01	Customer can place an order	System will keep track of orders per customer, and will only allow one type of deck, wheels, trucks, and board per order
RQ-02	Customer can browse inventory	System should be able to display all the items for boards, wheels, trucks, and decks so that the customer can add the part (s)he wants.
RQ-03	Customer can checkout	System will show what the customer has ordered so far and can delete the order if they want. Otherwise the customer gets their board!
RQ-04	Easy to read inventory and the attributes associated with each main part, such as seeing the hardness or color of wheels, or be able to know what design or color is on the deck.	System will print out each item with all its attributes so that the customer can make an informed decision on what they want.

RQ-05	Once a customer chooses a board, it will limit is options for parts specifically tailored for that board type	System must set flags when a customer chooses his board so that it limits the amount of options for that board.
RQ-06	As a Customer, you will be able check what you have ordered at any time.	Method in order will display the current order so far.

Non-Functional ID:	Requirements:	Responsibilities:
NF - 01	System runs on most computers	Code should run on most computers
NF – 02	Will be able to run worldwide	Self-Explanatory
NF - 03	System will handle large amounts of traffic to website	System must be able to run with thousands of users ordering at the same time
NF-04	A small pre determined database will hold the inventory class and its items so that a customer can pick and choose what they want	-

Class Diagram:

Customer

str: name
int: ID
str: Address

Wants to place order

Order

board: str
deck: str
trucks: str
wheels: str
orderNum: int

orderDeck()
orderTrucks()
orderWheels()
orderBoard()
checkout()

Ready to checkout

Checkout

board: str
deck: str
trucks: str
wheels: str

purchase()
deleteOrder()

Browse Inventory

Inventory

board: List<Board>
deck: List<Deck>
trucks: List<trucks>
wheels: List<wheels>

displayAll(board)
displayAll(deck)
displayAll(trucks)
displayAll(wheels)

Select Skate Parts

Board

length: int
width: int

Deck

Type: Str
length: int
width: int
color: str
design: str
griptape: str

Trucks

width: int
kingpin/reverse: bool
Tightness: int

Wheels

Size: int
Hardness: int
color: str

SkateBoard

StreetCredit: int
Name: str

PennyBoard

plasticColor: str
Name: str

LongBoard

dropThrough: Bool
Name: str