System Implementation (Readme)

Course: MSc Computer Science

Module: Object-Oriented Information Systems

Assignment: System Implementation

Date: Saturday 17th July 2021

Student ID: 12685395

Contents

Description of Solution Implemented:

This project serves as a basic technical implementation of the 'Online Store' brief provided on the learning platform. A key emphasis has been placed on Object-Oriented design, focusing on core concepts including Inheritance, Composition, Polymorphism and Encapsulation.

Following the UML diagrams produced in earlier assignments, ten classes have been implemented: Customer, DeliveryDetail, GiftVoucher, Order, Product, PromoCode, Seller, StoredPaymentDetails, User & Main (New class to represent the systems UI). Inheritance is demonstrated between the User and Customer/Seller classes, with various functions and attributes being taken from the parent class. Composition is demonstrated throughout the Order class, with Products, Delivery Details, Promo Codes and Gift Vouchers all being available. Finally, encapsulation has been shown within the Product class, with a series of 'getter'/'setter' methods to provide an interface to access private/protected attributes.

In a production environment, this solution would be implemented as a web-based solution, using an Object-Oriented MVC framework such as Django, using a SQL database for information storage/retrieval, as opposed to the array/dictionary stores currently in use.

Throughout the development of this system, testing has been a core focus, with every function being thoroughly tested before the development of the next one begins. Further down in this document, a thorough set of test cases have been

provided, demonstrating that the usability/functionality of the system performs as expected.

The code deliverables for this project have been produced in line with the Python PEP8 Style Guide (Van Rossum et al., 2001) to ensure that code is easily readable and maintainable by other developers. Furthermore, in line with the Python PEP257 conventions (Goodger et al., 2001), detail for each method is included as part of a multi-line docstring. In cases whereby further explanation is needed, additional multi-line comments are included in the appropriate sections of the code.

How to Execute Code (Windows):

1) For development purposes, I have chosen to use the VirtualEnvironment functionality within python to isolate my code and packages from others that may be installed on the system. Detailed steps are available within the Python Documentation (Python, n.d.). **Python 3.x must be installed to use this project.**

2) Navigate to the code directory and follow the steps in the above link to create and enter the Python virtual environment. For example, in Windows you would execute the following commands

   python -m venv env

```
C:\Users\kiero\Downloads\python-development>python -m venv env

C:\Users\kiero\Downloads\python-development>
```

   .\env\Scripts\activate

```
C:\Users\kiero\Downloads\python-development>.\env\Scripts\activate

(env) C:\Users\kiero\Downloads\python-development>
```

3) Once you are in the Virtual Environment, please execute the following command in order to install all external packages:

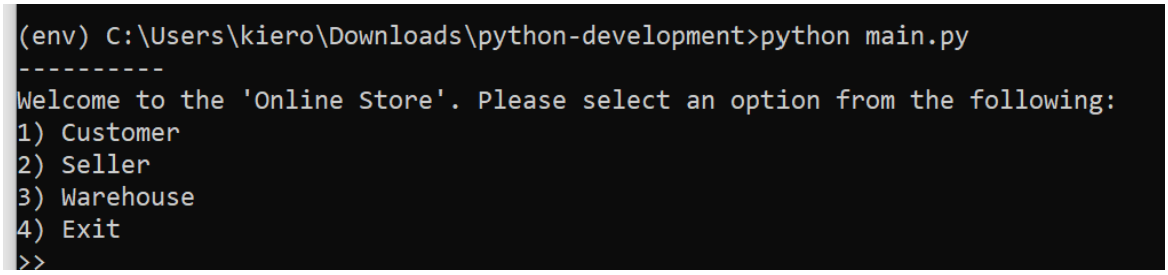
   pip install -r requirements

```
(env) C:\Users\kiero\Downloads\python-development>pip install -r requirements
Collecting autopep8==1.5.7
  Using cached autopep8-1.5.7-py2.py3-none-any.whl (45 kB)
Collecting prettytable==2.1.0
  Using cached prettytable-2.1.0-py3-none-any.whl (22 kB)
Collecting pycodestyle==2.7.0
  Using cached pycodestyle-2.7.0-py2.py3-none-any.whl (41 kB)
Collecting toml==0.10.2
  Using cached toml-0.10.2-py2.py3-none-any.whl (16 kB)
Collecting wcwidth==0.2.5
  Using cached wcwidth-0.2.5-py2.py3-none-any.whl (30 kB)
Installing collected packages: toml, pycodestyle, autopep8, wcwidth, prettytable
Successfully installed autopep8-1.5.7 prettytable-2.1.0 pycodestyle-2.7.0 toml-0.10.2 wcwidth-0.2.5
WARNING: You are using pip version 20.2.3; however, version 21.1.3 is available.
You should consider upgrading via the 'c:\users\kiero\downloads\python-development\env\scripts\python.exe -m pip install
 --upgrade pip' command.

(env) C:\Users\kiero\Downloads\python-development>
```

4) Once this is complete, you can execute the code within the Python Virtual Environment by entering the following command:

python main.py

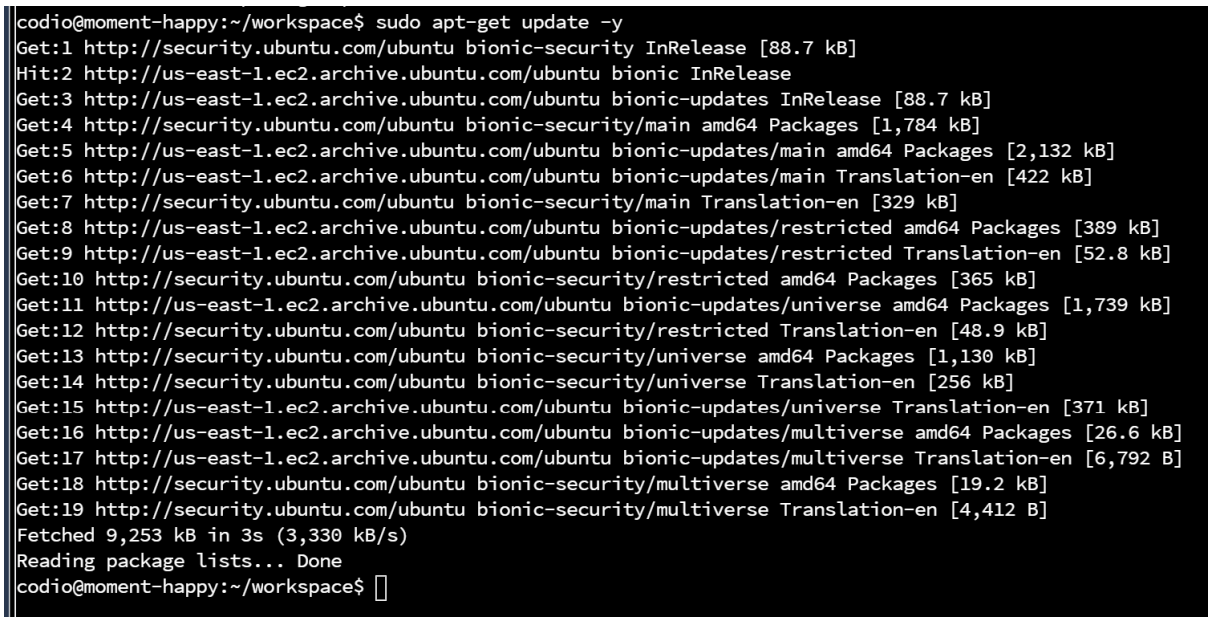


How to execute code (Codio):

Unfortunately, due to the Operating System image builds within the Codio environment, a few additional steps are needed on a fresh ‘box’. If you have already entered the pre-configured Codio environment, you can begin at the “source ./env/bin/activate” stage of Step 3.

1) Update the package repository within the Ubuntu instance using the following command:

   sudo apt-get update -y



2) Once this is complete, you will need to install the python3-venv package using the following command:

   sudo apt-get install python3-venv -y

```
codio@moment-happy:~/workspace$ sudo apt-get install python3-venv -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libpython3.6 libpython3.6-dev libpython3.6-minimal libpython3.6-stdlib python3.6 python3.6-dev python3.6-minimal python3.6-venv
Suggested packages:
  python3.6-doc binfmt-support
The following NEW packages will be installed:
  python3-venv python3.6-venv
The following packages will be upgraded:
  libpython3.6 libpython3.6-dev libpython3.6-minimal libpython3.6-stdlib python3.6 python3.6-dev python3.6-minimal
7 upgraded, 2 newly installed, 0 to remove and 182 not upgraded.
Need to get 50.9 MB of archives.
After this operation, 109 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3.6-dev amd64 3.6.9-1~18.04ubuntu1.4 [508 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython3.6-dev amd64 3.6.9-1~18.04ubuntu1.4 [44.9 MB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython3.6 amd64 3.6.9-1~18.04ubuntu1.4 [1,414 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3.6 amd64 3.6.9-1~18.04ubuntu1.4 [203 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython3.6-stdlib amd64 3.6.9-1~18.04ubuntu1.4 [1,712
 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3.6-minimal amd64 3.6.9-1~18.04ubuntu1.4 [1,610 k
B]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython3.6-minimal amd64 3.6.9-1~18.04ubuntu1.4 [534
kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 python3.6-venv amd64 3.6.9-1~18.04ubuntu1.4 [6,188
B]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 python3-venv amd64 3.6.7-1~18.04 [1,208 B]
Fetched 50.9 MB in 2s (28.6 MB/s)
(Reading database ... 36883 files and directories currently installed.)
Preparing to unpack .../0-python3.6-dev_3.6.9-1~18.04ubuntu1.4_amd64.deb ...
Unpacking python3.6-dev (3.6.9-1~18.04ubuntu1.4) over (3.6.8-1~18.04.3) ...
Preparing to unpack .../1-libpython3.6-dev_3.6.9-1~18.04ubuntu1.4_amd64.deb ...
Unpacking libpython3.6-dev:amd64 (3.6.9-1~18.04ubuntu1.4) over (3.6.8-1~18.04.3) ...
Preparing to unpack .../2-libpython3.6_3.6.9-1~18.04ubuntu1.4_amd64.deb ...
Unpacking libpython3.6:amd64 (3.6.9-1~18.04ubuntu1.4) over (3.6.8-1~18.04.3) ...
Preparing to unpack .../3-python3.6_3.6.9-1~18.04ubuntu1.4_amd64.deb ...
Unpacking python3.6 (3.6.9-1~18.04ubuntu1.4) over (3.6.8-1~18.04.3) ...
Preparing to unpack .../4-libpython3.6-stdlib_3.6.9-1~18.04ubuntu1.4_amd64.deb ...
Unpacking libpython3.6-stdlib:amd64 (3.6.9-1~18.04ubuntu1.4) over (3.6.8-1~18.04.3) ...
Preparing to unpack .../5-python3.6-minimal_3.6.9-1~18.04ubuntu1.4_amd64.deb ...
Unpacking python3.6-minimal (3.6.9-1~18.04ubuntu1.4) over (3.6.8-1~18.04.3) ...
Preparing to unpack .../6-libpython3.6-minimal_3.6.9-1~18.04ubuntu1.4_amd64.deb ...
Unpacking libpython3.6-minimal:amd64 (3.6.9-1~18.04ubuntu1.4) over (3.6.8-1~18.04.3) ...
Selecting previously unselected package python3.6-venv.
Preparing to unpack .../7-python3.6-venv_3.6.9-1~18.04ubuntu1.4_amd64.deb ...
Unpacking python3.6-venv (3.6.9-1~18.04ubuntu1.4) ...
Selecting previously unselected package python3-venv.
Preparing to unpack .../8-python3-venv_3.6.7-1~18.04_amd64.deb ...
Unpacking python3-venv (3.6.7-1~18.04) ...
Setting up libpython3.6-minimal:amd64 (3.6.9-1~18.04ubuntu1.4) ...
Setting up libpython3.6-stdlib:amd64 (3.6.9-1~18.04ubuntu1.4) ...
```

```
Setting up python3.6 (3.6.9-1~18.04ubuntu1.4) ...
Setting up python3.6-venv (3.6.9-1~18.04ubuntu1.4) ...
Setting up libpython3.6-dev:amd64 (3.6.9-1~18.04ubuntu1.4) ...
Setting up python3.6-dev (3.6.9-1~18.04ubuntu1.4) ...
Setting up python3-venv (3.6.7-1~18.04) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
codio@moment-happy:~/workspace$
```

3) You must now create a Python Virtual Environment in the code directory, using the steps available in the Python Documentation (Python, n.d.). Once you have entered the Virtual Environment, your console input prompt will be prefixed with (env).

python3 -m venv env

```
codio@moment-happy:~/workspace$ python3 -m venv env
codio@moment-happy:~/workspace$
```

source ./env/bin/activate

```
codio@moment-happy:~/workspace$ source ./env/bin/activate
(env) codio@moment-happy:~/workspace$
```

4) Once this has been complete, you must install the pre-requisite modules from the Python pip package manager using the following command:

pip install -r requirements

```
(env) codio@moment-happy:~/workspace$ pip install -r requirements
Collecting autopep8==1.5.7 (from -r requirements (line 1))
  Downloading https://files.pythonhosted.org/packages/a7/f6/84070ab117e6b080a87aac0ac9e4d269a66c6f6076ad81509bd0aac828d8/autopep8-1.5.
7-py2.py3-none-any.whl (45kB)
    100% |████████████████████████████████| 51kB 3.4MB/s
Collecting prettytable==2.1.0 (from -r requirements (line 2))
  Downloading https://files.pythonhosted.org/packages/26/1b/42b59a4038bc0442e3a0085bc0de385658131eef8a88946333f870559b09/prettytable-2
.1.0-py3-none-any.whl
Collecting pycodestyle==2.7.0 (from -r requirements (line 3))
  Downloading https://files.pythonhosted.org/packages/de/cc/227251b1471f129bc35e966bb0fceb005969023926d744139642d847b7ae/pycodestyle-2
.7.0-py2.py3-none-any.whl (41kB)
    100% |████████████████████████████████| 51kB 7.5MB/s
Collecting toml==0.10.2 (from -r requirements (line 4))
  Downloading https://files.pythonhosted.org/packages/44/6f/7120676b6d73228c96e17f1f794d8ab046fc910d781c8d151120c3f1569e/toml-0.10.2-p
y2.py3-none-any.whl
Collecting wcwidth==0.2.5 (from -r requirements (line 5))
  Downloading https://files.pythonhosted.org/packages/59/7c/e39aca596badaf1b78e8f547c807b04dae603a433d3e7a7e04d67f2ef3e5/wcwidth-0.2.5
-py2.py3-none-any.whl
Collecting importlib-metadata; python_version < "3.8" (from prettytable==2.1.0->-r requirements (line 2))
  Downloading https://files.pythonhosted.org/packages/3f/e1/e5bba549a033adf77448699a34ecafc7a32adaeeb4369396b35f56d5cc3e/importlib_met
adata-4.6.1-py3-none-any.whl
Collecting zipp>=0.5 (from importlib-metadata; python_version < "3.8"->prettytable==2.1.0->-r requirements (line 2))
  Downloading https://files.pythonhosted.org/packages/92/d9/89f433969fb8dc5b9cbdd4b4deb587720ec1aeb59a020cf15002b9593eef/zipp-3.5.0-py
3-none-any.whl
Collecting typing-extensions>=3.6.4; python_version < "3.8" (from importlib-metadata; python_version < "3.8"->prettytable==2.1.0->-r r
equirements (line 2))
  Downloading https://files.pythonhosted.org/packages/2e/35/6c4fff5ab443b57116cb1aad46421fb719bed2825664e8fe77d66d99bcbc/typing_extens
ions-3.10.0.0-py3-none-any.whl
Installing collected packages: pycodestyle, toml, autopep8, zipp, typing-extensions, importlib-metadata, wcwidth, prettytable
Successfully installed autopep8-1.5.7 importlib-metadata-4.6.1 prettytable-2.1.0 pycodestyle-2.7.0 toml-0.10.2 typing-extensions-3.10.
0.0 wcwidth-0.2.5 zipp-3.5.0
(env) codio@moment-happy:~/workspace$
```

5) The code can now be executed using the following console command:

python main.py

```
(env) codio@moment-happy:~/workspace$ python main.py
----------
Welcome to the 'Online Store'. Please select an option from the following:
1) Customer
2) Seller
3) Warehouse
4) Exit
>>
```

Test Data

Promo Codes:

Voucher Code	Promo Expiry	Promo Amount
10OFF	2021-07-25	£10.00 (Fixed amount, not percentage)
20OFF	2021-07-25	£20.00 (Fixed amount, not percentage)

Gift Vouchers:

Voucher Code	Voucher Expiry	Voucher Amount
FIFTEENGV	2021-07-25	£15.00
TWENTYY	2021-07-27	£20.00

Products:

The application auto-generates the product ID upon launch. Please use the "View all products" page in the Customer Menu to get the correct information.

Test Cases:

Customer – Viewing Products:

Description:

This Test Case determines whether the "View All Products" function outputs the correct product data to the customer.

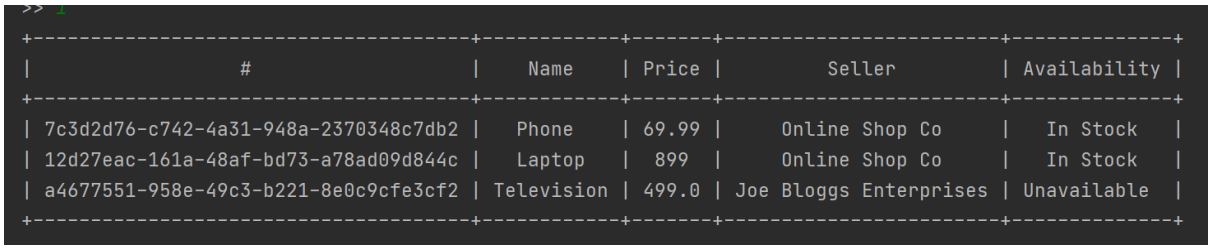
Expected Output:

The three products below should be displayed to the end-user when executing the "View All Products" function within the Customer menu.

Product #	**Product Name**	**Product Price**	**Product Seller**	**Product Stock Level**
Random V4 UUID generated on program execution	Phone	69.99	Online Shop (First Party)	4 (Available)
Random V4 UUID generated on program execution	Laptop	899	Online Shop (First Party)	18 (Available)
Random V4 UUID generated	Television	499	Joe Bloggs Enterprises (Third Party)	0 (Unavailable)

on program execution				

Actual Output:



The above screenshot is the data returned by the application when this option is selected. The products displayed are a 100% match against the expected data.

Customer – Searching Products:

Description:

This Test Case determines whether the application returns the correct results when the customer requests to search for products.

Expected Output:

Searching for "pho":

When searching for "pho", only the Phone product should be displayed to the user.

Product #	**Product Name**	**Product Price**	**Product Seller**	**Product Stock Level**
Random V4 UUID generated on program execution	Phone	69.99	Online Shop (First Party)	4 (Available)

Searching for "p":

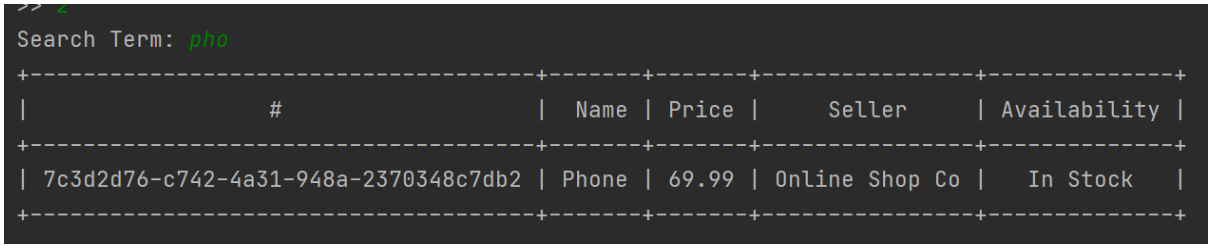
When searching for "p", a total of two products should be displayed to the user, Phone and Laptop.

Product #	**Product Name**	**Product Price**	**Product Seller**	**Product Stock Level**
Random V4 UUID generated on program execution	Phone	69.99	Online Shop (First Party)	4 (Available)

Random V4 UUID generated on program execution	Laptop	899	Online Shop (First Party)	18 (Available)

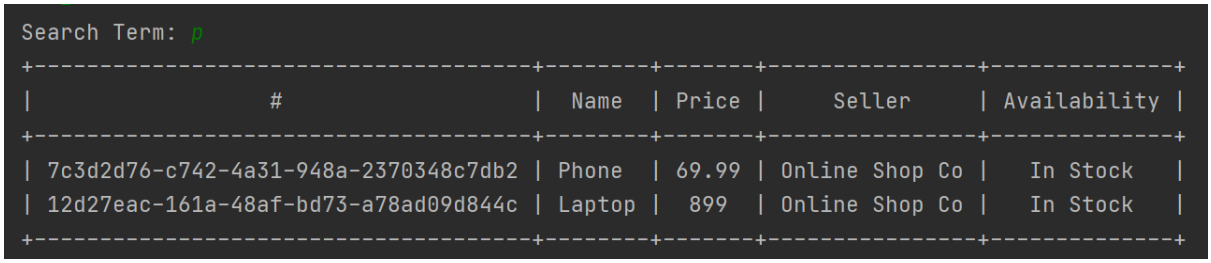
Actual Output:

Searching for "pho":



When searching for "pho", the correct product is displayed to the customer (Phone).

Searching for "p":



When searching for "p", the two expected products are displayed to the customer.

Customer – Adding Products to Cart:

Description:

This Test Case determines whether the "Adding Product To Cart" function works as intended by adding the item to the "products" variable within the Order class.

Expected Output:

Adding a phone to the cart:

When entering the correct product ID for the phone item, the program should respond with "{item} added to basket".

Adding invalid items to the cart:

When attempting to add an invalid item ID to the cart, the program should respond with "No results found for supplied ID".

Actual Output:

Adding a phone to the cart:

```
Product #: 7c3d2d76-c742-4a31-948a-2370348c7db2
Phone added to basket
```

The application successfully adds the Phone item to the cart when the correct ID value is entered.

Adding invalid items to the cart:

```
Product #: xxx
No results found for supplied ID
```

The application correctly returns an error when the user attempts to enter an incorrect item ID into the cart.

Customer – Removing Products from Cart:

Description:

This Test Case determines whether the "Remove Product From Cart" function successfully removes the product from the cart.

Expected Output:

Removing items already in cart:

When removing an item that is already in the cart, the application should respond with "{item} removed from basket".

Removing invalid item ID from the cart:

When attempting to remove an item that doesn't exist, the application should respond with "No results found for supplied ID".

Actual Output:

Removing items already in cart:

```
Product #: 7c3d2d76-c742-4a31-948a-2370348c7db2
Phone removed from basket
```

The application correctly removes the Phone product from the cart.

Removing invalid item ID from the cart:

```
Product #: xxx
No results found for supplied ID
```

The application correctly returns an error message, as the product doesn't exist.

Customer (Checkout) – Shipping Method

Description:

This Test Case establishes whether the In-House shipping method is available for only in-house products (Postal Service for all others).

Expected Output:

Only first-party products in cart:

If only first-party products are in the cart, the in-house shipping method should not error when selected.

Third party items in cart:

If third-party items are in the cart, the Postal System shipping method is the only one that should be usable.

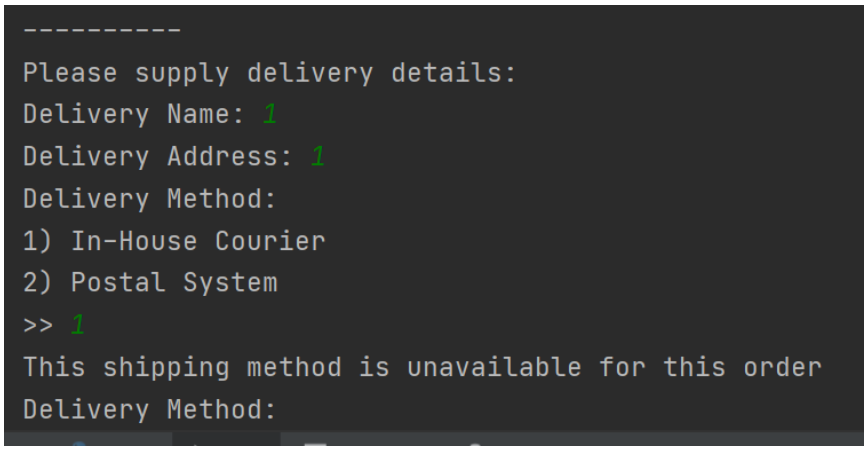
Actual Output:

Only first-party products in cart:

```
----------
Please supply delivery details:
Delivery Name: 1
Delivery Address: 1
Delivery Method:
1) In-House Courier
2) Postal System
>> 1
----------
Order Total: 899.0
```

When only first-party products are in the cart, the application proceeds to the next stage of the checkout process when in-house courier is selected.

Third party items in cart:



When items sold by third party sellers are in the cart, the application returns a “This shipping method is unavailable for this order” error message as expected.

Customer (Checkout) – Gift Code:

Description:

This Test Case establishes whether the Gift Codes are correctly accepted (and debited) from the total remaining balance.

Expected Output:

Valid Gift Code:

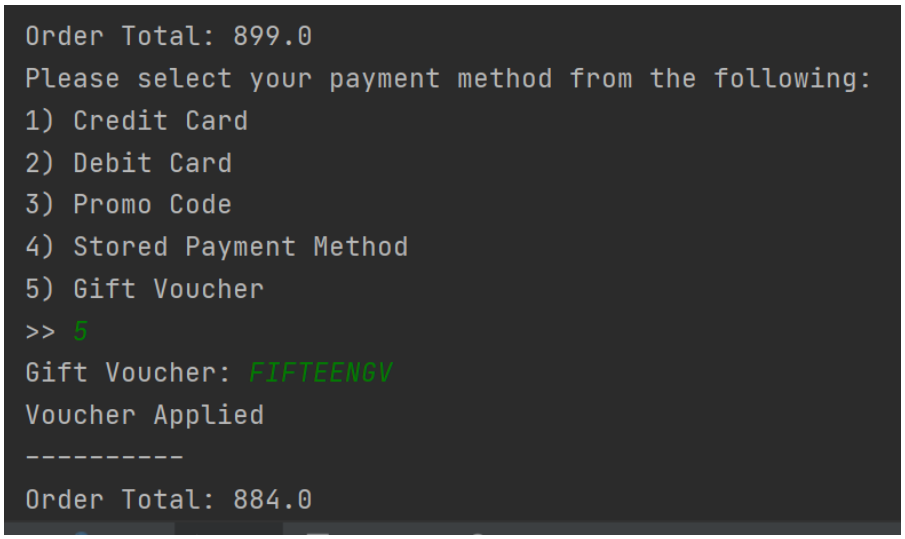
A Valid Gift code should result in the "Voucher Applied" message being displayed to the user, followed by the total amount payable reduced by the gift code's value.

Invalid Gift Code:

An invalid gift code should result in the "Gift Voucher Invalid/Expired" error message being displayed to the user. The total amount payable should remain unchanged.

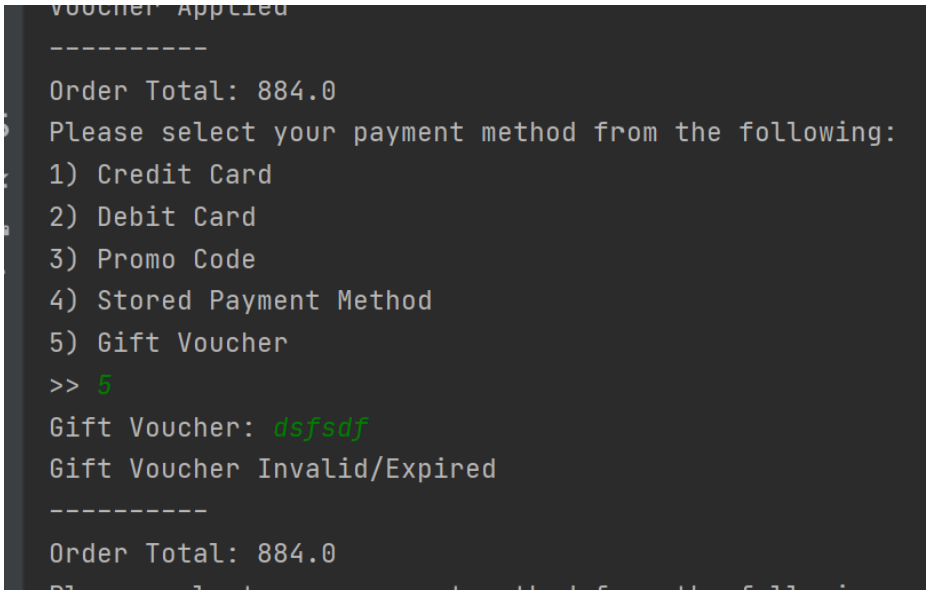
Actual Output:

Valid Gift Code:



The "Voucher Applied" message was displayed to the user as expected. However, the total balance was reduced by £15, leaving £884 payable by the customer.

Invalid Gift Code:



The application returned an error message, as expected. The total amount payable was not reduced from the original sum of £884.

Customer (Checkout) – Promo Code:

Description:

This Test Case establishes whether the Promo Codes are correctly accepted (and debited) from the total remaining balance.

Expected Output:

Valid Promo Code:

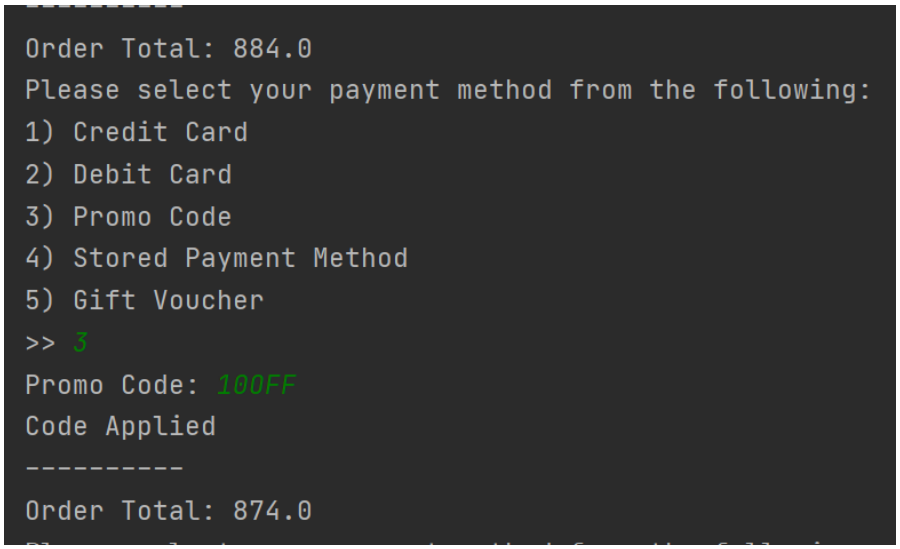
The application should display "Code Applied". The total balance remaining should be reduced by the discount value associated with that specific promo code.

Invalid Promo Code:

The application should display an error message to the user. The total balance should remain unchanged.

Actual Output:

Valid Promo Code:



The application returned the "Code Applied" message as expected. The order total was reduced by £10.

Invalid Promo Code:

```
----------
Order Total: 874.0
Please select your payment method from the following:
1) Credit Card
2) Debit Card
3) Promo Code
4) Stored Payment Method
5) Gift Voucher
>> 3
Promo Code: dfgvs
Promo Code Invalid/Expired
----------
Order Total: 874.0
```

The application displayed an error message as expected. The order total remained unchanged.

Customer (Checkout) – Credit Card/Debit Card:

Description:

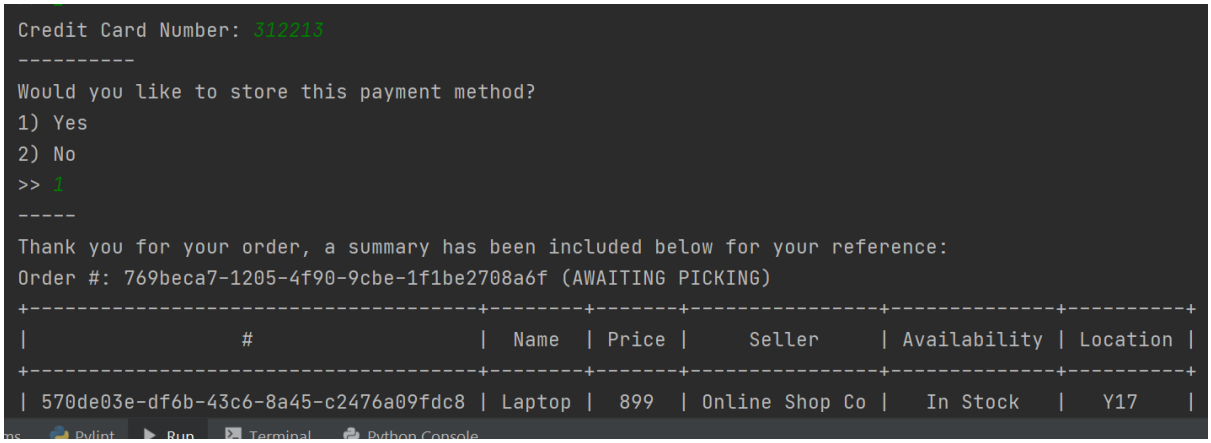
This Test Case demonstrates the ability to take Credit/Debit Card payments using the system and providing those users with the ability to store their payment details for use later in the program.

Expected Output:

The application will prompt the user to enter their Credit/Debit card number. Once this has been completed, the user will be provided with the option to store their payment method for use another time.

At this stage, the status of the order is updated to "Awaiting Picking", and a summary of the completed order is displayed to the user

Actual Output:



The user was provided with the option to save their credit/debit card details. As the order is complete, details of the order are then presented to the user.

Customer (Checkout) – Stored Payment Method:

Description:

This Test Case establishes whether the customer can use a Stored Payment method to complete their order.

Expected Output:

The user should be presented with a list of two stored payment methods. Once a valid method has been selected, the order will be marked as completed ("Awaiting Picking").

Actual Output:

```
Please select a stored payment card:
0) 991999182818
1) 817281728172
1
-----
Thank you for your order, a summary has been included below for your reference:
Order #: fe0eb504-fe10-4e3b-a73c-5bd57bead970 (AWAITING PICKING)
+--------------------------------------+--------+-------+----------------+--------------+----------+
|                  #                   |  Name  | Price |     Seller     | Availability | Location |
+--------------------------------------+--------+-------+----------------+--------------+----------+
| 9eca808e-1d00-4b60-b6d5-6246a417ccbd | Laptop |  899  | Online Shop Co |   In Stock   |   Y17    |
+--------------------------------------+--------+-------+----------------+--------------+----------+
Order Total: 899.0
----------
```

The stored payment method was accepted, and the order was completed. A summary of the order was displayed to the user before redirecting to the main menu.

Seller – View Storefront:

Description:

This Test Case determines whether the application correctly returns the products belonging to the current seller ('The storefront').

Expected Output:

Product #	**Product Name**	**Product Price**	**Product Seller**	**Product Stock Level**
Random V4 UUID generated on program execution	Television	499	Joe Bloggs Enterprises (Third Party)	0 (Unavailable)

The application should only return the "Television" item, as this is the only one belonging to the external seller (Joe Bloggs Enterprises).

Actual Output:

```
Seller Detail: Joe Bloggs Enterprises (joe@bloggs.com)
+--------------------------------------+------------+-------+-----------------------+--------------+
|                  #                   |    Name    | Price |         Seller        | Availability |
+--------------------------------------+------------+-------+-----------------------+--------------+
| a4677551-958e-49c3-b221-8e0c9cfe3cf2 | Television | 499.0 | Joe Bloggs Enterprises | Unavailable  |
+--------------------------------------+------------+-------+-----------------------+--------------+
```

The application returns the correct products belonging to the external seller.

Seller – Edit Product:

Description:

This Test Case determines whether the application allows the seller to edit the pricing of their own items.

Expected Output:

The seller edits their item:

The application should display an "Enter Price" prompt, allowing the seller to specify an updated price for their item.

The seller edits another sellers item:

The application should respond with an "ID does not match seller" error.

Actual Output:

The seller edits their item:

```
>> 2
Search Term: a4677551-958e-49c3-b221-8e0c9cfe3cf2
New Price: 123.12
```

The application correctly identifies that the Phone item belongs to the external seller, prompting the user to enter an updated price.

The seller edits another sellers item:

```
Search Term: 12d27eac-161a-48af-bd73-a78ad09d844c
ID Does not match seller
```

The application correctly identifies that the product ID entered does not belong to the external seller, therefore, returns an error message to the user.

Warehouse – View Orders/Locations:

Description:

This Test Case determines whether the View Orders/Locations function correctly displays a list of items in the customers' order and the warehouse location of the specific item.

Expected Output:

The application outputs a list of items that the customer has added to their in-progress order and the three-digit warehouse location code.

Actual Output:

```
Order #: 5e0f928f-55b2-4db0-b82a-449a1fecd662 (IN PROGRESS)
User Detail: Kieron Holmes (k@h.com)
+--------------------------------------+--------+-------+---------------+-------------+----------+
|                  #                   |  Name  | Price |     Seller    | Availability | Location |
+--------------------------------------+--------+-------+---------------+-------------+----------+
| bfc077e7-f634-46a6-89b5-aa5c9798721e | Laptop |  899  | Online Shop Co |   In Stock   |   Y17    |
+--------------------------------------+--------+-------+---------------+-------------+----------+
```

The application correctly returns the Order ID, the user information, and the details of the products in the order (Including the warehouse location).

Warehouse – Scan Item:

Description:

This Test Case determines whether the warehouse operatives can amend the stock levels by 'scanning' an item (Simulated by entering the product ID).

Expected Output:

Entering valid product ID:

If a valid product ID is entered, the warehouse operative should be prompted to enter an updated stock level for the 'scanned' item.

Entering invalid product ID:

When an invalid item ID is entered, the "No results found for supplied ID" error message should be displayed.

Actual Output:

Entering valid product ID:

```
Product #: a4677551-958e-49c3-b221-8e0c9cfe3cf2
New Stock Level: 1
Stock updated to 1
```

The application successfully returns the "Stock updated to 1" message.

Entering invalid product ID:

```
Product #: sdfsdf
No results found for supplied ID
```

The application correctly displays an error message when an invalid product ID is entered.

Warehouse – Check Shipping Method & Mark Order as Shipped:

Description:

This Test Case determines whether the shipping method/details and order information are correctly displayed on the "Check Shipping Method & Mark Order as Shipped" submenu.

Expected Output:

An order that is "Awaiting Picking":

In previous stages, the customers' details have supplied the Shipping Method, Name and Delivery Address. These should be displayed to the user, followed by the status being updated to "Shipped".

An order that is not "Awaiting Picking" (Checkout incomplete):

An error message should be displayed to the user informing them that the order's status is not "Awaiting Picking".

Actual Output:

An order that is "Awaiting Picking":

```
Shipping Method: 123 Fake Street
Delivery Name: 1
Delivery Address: IN HOUSE COURIER
Order #2205b1ae-d2e9-4c7b-9382-fafda1275b03 was marked as shipped!
```

The system correctly updated the status to Shipped and displayed all relevant information about the order to warehouse staff.

An order that is not "Awaiting Picking" (Checkout incomplete):

```
Order status is not awaiting picking
```

An error is correctly displayed to the warehouse staff as the order status is not set to "Awaiting Picking".

References:

Python. (n.d.) Installing packages using pip and virtual environments. Available From: https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/#installing-virtualenv [Accessed Monday 19th July 2021].

Van Rossum et al. (2001) PEP 8 – Style Guide for Python Code. Available From: https://www.python.org/dev/peps/pep-0008/ [Accessed 18th July 2021].

Goodger et al. (2001) PEP 257 – Docstring Conventions. Available From: https://www.python.org/dev/peps/pep-0257/ [Accessed 18th July 2021].