TERMINAL APPLICATION

T1A3

CREATOR MICHELLE OHANNESSIAN
JULY 2022

CONTENTS:

MY WARDROBE

HAVE YOU EVER TRIED TO PURCHASE CLOTHING OR FOOTWEAR ONLINE BUT DON'T REMEMBER WHAT SIZE YOU ARE FOR A PARTICULAR BRAND?

'MY WARDROBE' ALLOWS THE USER TO RECORD & STORE INFORMATION RELATED TO THEIR CLOTHING & FOOTWEAR ITEMS SO THEY ARE ABLE TO RECALL & VIEW THE ITEM'S DETAILS SUCH AS ITS SIZE AND BRAND, ETC.

MY WARDROBE

Planning, Design and Development

MY WARDROBE
USER STORIES

- *As an aspiring web developer,*

  I want to create an app to showcase my skill set in Python

  Using my terminal app assignment

- *As as online shopper,*

  I want to be able to reference my clothing sizes immediately when shopping online

  To utilise my time spent shopping

- *As an online shopper,*

  I want to be able to store my sizes somewhere for each brand

  So I can easily reference them when I am placing an order

MY WARDROBE REVIEW

Classes:

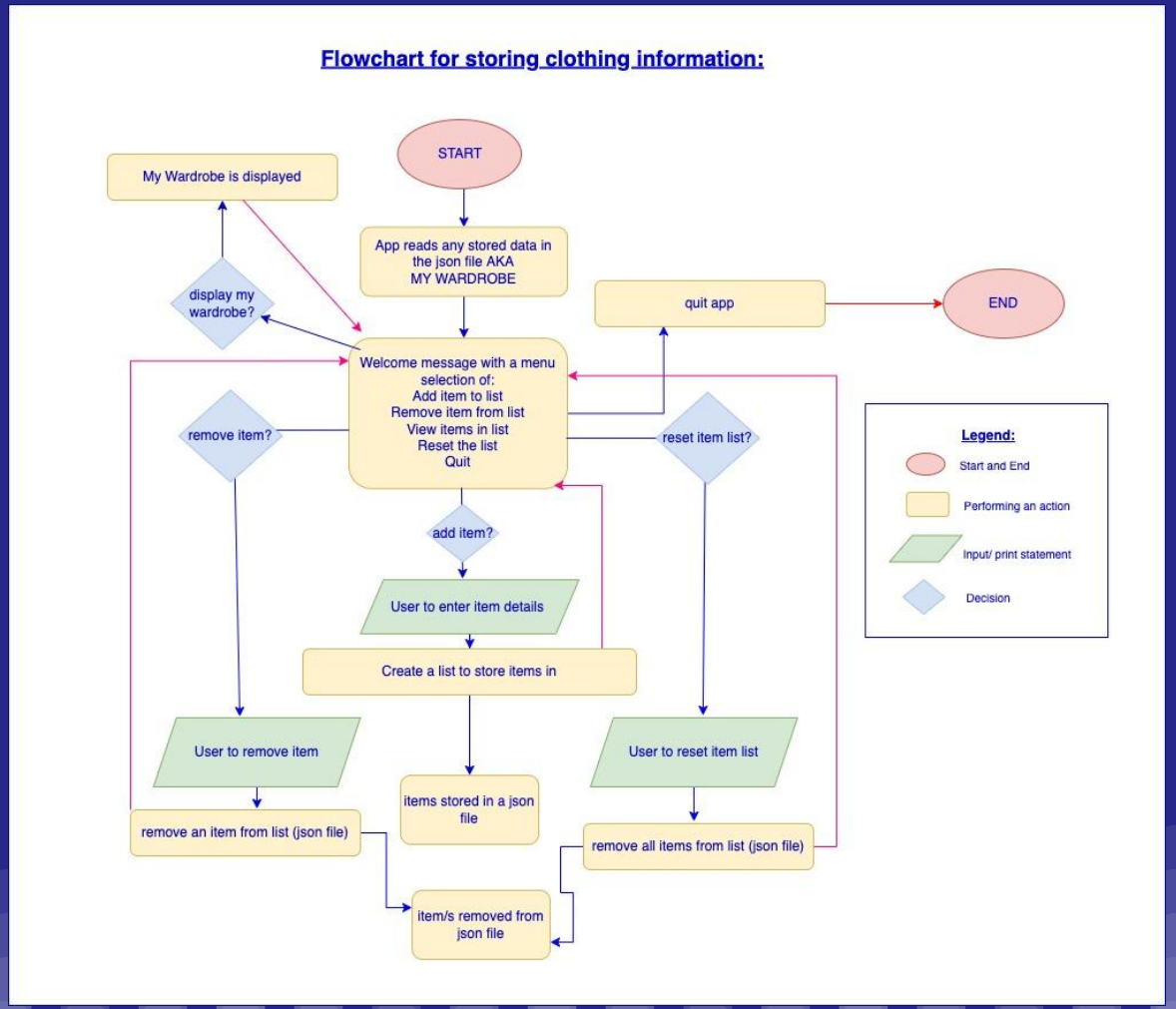
Main Class to run app

InputOutput Class

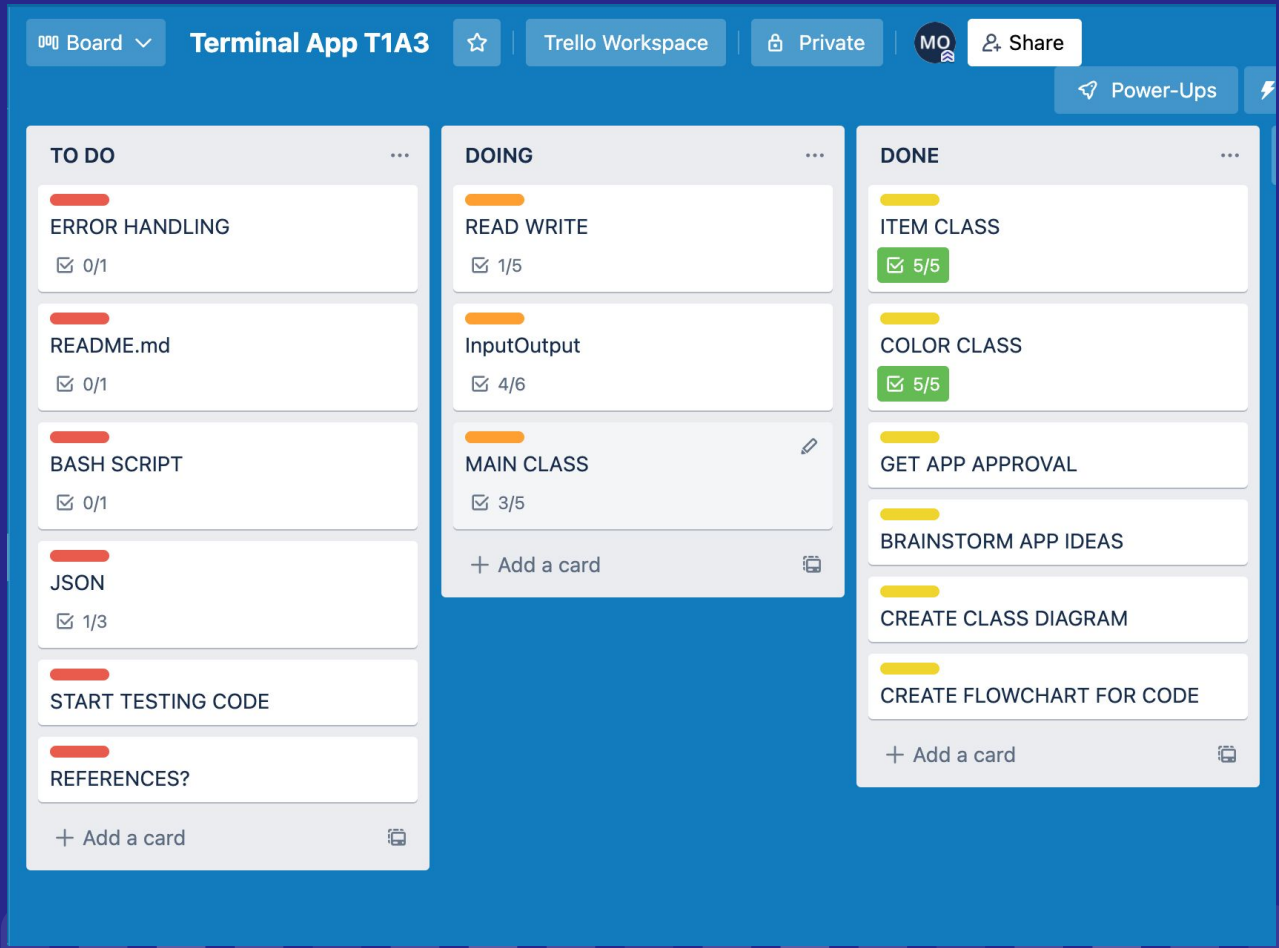
ReadWrite class to store User input

Color Class to colour text

MY WARDROBE LOGIC



MY WARDROBE CHECKLIST IN TRELLO



WELCOME TO:

MY WARDROBE

APP FEATURES:

1.MAIN MENU

2.ADDING ITEMS

3.REMOVING ITEMS

4.APP OUTPUT

5.STORING INFORMATION IN A SEPARATE FILE

MY WARDROBE
WALK-THROUGH TERMINAL APP

```
→  T1A3 git:(main) python3 main.py
Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
```

- Run MY WARDROBE in terminal
- Python3 is required for this application to run
- Welcome and main menu displays

MY WARDROBE
WALK-THROUGH TERMINAL APP

```
Please enter a title for your item:  comfy jeans
Please enter the item description:  boyfriend
Please enter the item style:  pant
Please enter the item size on tag:  12
Please enter the item brand:  Nobody
Please enter the item price:  299
Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
```

- User selects 1
- The app prints a list of questions, one by one for the user to input their data and the app saves their input as a list using the separate json file (list.json)
- Once finished, the main menu runs again

MY WARDROBE
WALK-THROUGH TERMINAL APP

```
Title: comfy jeans, Description: boyfriend, Style: pant, Size:12, Brand:Nobody, Price: $299


Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
```

- The user selects 3 to view their items
- The list saved in the app displays for their viewing
- Main menu then displays again

MY WARDROBE
WALK-THROUGH TERMINAL APP

```
Please enter a title for your item:  Fav jacket
Please enter the item description:  bomber
Please enter the item style:  jacket
Please enter the item size on tag:  12
Please enter the item brand:  Nobody
Please enter the item price:  199
Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
```

- User selects 1 again to add item
- The app prints a list of questions for the user's input and saves the data they enter as a list.
- Once finished, the main menu runs again

MY WARDROBE
WALK-THROUGH TERMINAL APP

```
Title: comfy jeans, Description: boyfriend, Style: pant, Size:12, Brand:Nobody, Price: $299


Title: Fav jacket, Description: bomber, Style: jacket, Size:12, Brand:Nobody, Price: $199


Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
```

- The user selects 3 to view items
- The list saved in the app displays for their viewing (2 items now in the list)
- Main menu then displays again

MY WARDROBE
WALK-THROUGH TERMINAL APP

```
Which item do you want to remove by title?:
Fav jacket
Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
```

- User selects 2 to remove an item from the list
- App asks what item by title the user would like to remove
- User input “Fav jacket” and it is removed
- Main menu runs

MY WARDROBE
WALK-THROUGH TERMINAL APP

```
Title: comfy jeans, Description: boyfriend, Style: pant, Size: 12, Brand: Nobody, Price: $299


Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
4
```

- The user selects 3 to view items saved
- The list saved in the app(json file) displays for their viewing (1 items now in the list)
- Main menu then displays again
- User selects 4 to reset the list

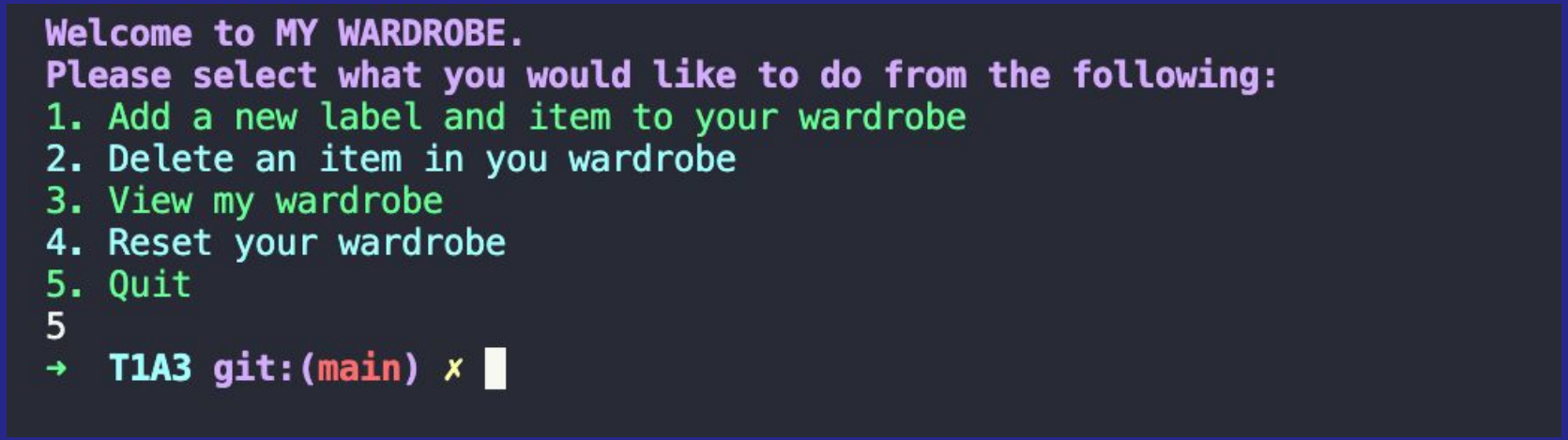
MY WARDROBE
WALK-THROUGH TERMINAL APP

```
Are you sure you want to reset your wardrobe(Y/N)?

y
Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
```

- A question is printed asking user to reconfirm that they would like to reset their wardrobe:
- If Y, list is cleared and reset in json file, main menu displays
- If N, main menu displays

MY WARDROBE
WALK-THROUGH TERMINAL APP



- User selects 5 to quit
- User exits MY WARDROBE APP

MY WARDROBE
WALK-THROUGH TERMINAL APP

```
→  T1A3 git:(main) ✗ python3 main.py
Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
3
```

- If you execute the app again
- User enters 3 to view their wardrobe

MY WARDROBE
WALK-THROUGH TERMINAL APP

```
Title: comfy jeans, Description: boyfriend, Style: pant, Size: 12, Brand: Nobody, Price: $299


Welcome to MY WARDROBE.
Please select what you would like to do from the following:
1. Add a new label and item to your wardrobe
2. Delete an item in you wardrobe
3. View my wardrobe
4. Reset your wardrobe
5. Quit
```

- Wardrobe has been stored and saved in json file so that is can be recalled anytime:

MY WARDROBE CODE

MY WARDROBE APP CODE

Main where app is run from

main.py

```python
from read_write import ReadWrite
import os
from input_output import InputOutput


class Main:

    def __init__(self, input_output, read_write):
        self.input_output = input_output
        self.read_write = read_write
        self.running = True

    def app(self):
        while self.running:
            self.input_output.welcome_menu()
        # class and method then save it to a variable
        # method waits for input from user and I capture it menu_selection and store it
            menu_selection = self.input_output.user_input()
            if menu_selection == "1":  # Add a new label and item to your wardrobe
                os.system('clear')
                # store info in dictionary, save to json and create an item list
                item = self.input_output.create_item()
                self.read_write.write_item_to_json(item)
            elif menu_selection == "2":
                os.system('clear')
                self.input_output.delete_item_question()
                delete_title = self.input_output.user_input()
                json_list = self.read_write.get_json_list()
                for item in json_list:
                    # checking if the items name of this iteration is equal to the name we receive
                    if item["title"] == delete_title:
                        # access to the list and remove the item
                        json_list.remove(item)
                        break
```

MY WARDROBE APP CODE

main.py

Used dependency injection with the InputOutput class and ReadWrite class

```
35                 self.read_write.write_list_to_json(json_list)
36             elif menu_selection == "3":
37                 os.system('clear')
38                 json_list = self.read_write.get_json_list()
39                 self.input_output.display_wardrobe(json_list)
40             elif menu_selection == "4":
41                 os.system('clear')
42                 self.input_output.reset_wardrobe_question()
43                 delete_wardrobe = self.input_output.user_input()
44                 # for wardrobe_reset_question in self.items:
45                 # checking if the items name of this iteration is equal to the name we receive
46                 if delete_wardrobe.lower() == "y":
47                     # access wardrobe and remove the it
48                     self.read_write.reset_json()
49                 else:
50                     os.system('clear')
51                     self.input_output.not_reset_wardrobe()
52             elif menu_selection == "5":
53                 self.running = False
54
55
56 # dependance injection - data base class - InputOutput
57 main = Main(InputOutput(), ReadWrite(('list.json')))
58 main.app()
```

MY WARDROBE APP CODE

Json file

list.json

MY WARDROBE APP CODE

InputOutput.py

Welcome_menu output for user selection

Create_item function print questions for user input

```python
import os
from color import Colors


class InputOutput:

    def welcome_menu(self):
        print(Colors.BOLD + Colors.PURPLE +
              "Welcome to MY WARDROBE. \nPlease select what you would like to do from the following:   " + Colors.END)
        print(Colors.GREEN +
              "1. Add a new label and item to your wardrobe" + Colors.END)
        print(Colors.BLUE + "2. Delete an item in you wardrobe" + Colors.END)
        print(Colors.GREEN + "3. View my wardrobe" + Colors.END)
        print(Colors.BLUE + "4. Reset your wardrobe" + Colors.END)
        print(Colors.GREEN + "5. Quit" + Colors.END)

    def create_item(self):
        os.system('clear')
        title = self.user_input(
            Colors.YELLOW + f"Please enter a title for your item:  " + Colors.END)
        description = self.user_input(
            Colors.RED + f"Please enter the item description:  " + Colors.END)
        style = self.user_input(
            Colors.GREEN + f"Please enter the item style:  " + Colors.END)
        size = self.user_input(
            Colors.BLUE + f"Please enter the item size on tag:  " + Colors.END)
        brand = self.user_input(
            Colors.PURPLE + f"Please enter the item brand:  " + Colors.END)
        price = self.user_input(
            Colors.YELLOW + f"Please enter the item price:  " + Colors.END)
```

MY WARDROBE APP CODE

Dictionary of item info

Functions for input and output

InputOutput.py

```
31          return {
32              "title": title,
33              "description": description,
34              "style": style,
35              "size": size,
36              "brand": brand,
37              "price": price,
38          }
39
40      def display_wardrobe(self, items):
41          os.system('clear')
42          for item in items:
43              print(f"Title: {item['title']}, Description: {item['description']}, Style: {item['style']}, Size: {item['size']}, Brand: {item['brand']}, Price: ${i
44            # get attribute from object
45
46  # wrapper method we are in control of user_input
47      def user_input(self, message=""):
48          return input(message)
49
50      def delete_item_question(self):
51          print(Colors.BOLD + Colors.YELLOW +
52                "Which item do you want to remove by title?:   \n" + Colors.END)
53
54      def reset_wardrobe_question(self):
55          print(Colors.BOLD + Colors.RED +
56                "Are you sure you want to reset your wardrobe(Y/N)?  \n" + Colors.END)
57
58      def not_reset_wardrobe(self):
59          print(Colors.BOLD + Colors.BLUE +
60                "Your wardrobe has not been reset.\n" + Colors.END)
```

MY WARDROBE APP CODE

ReadWrite stores, resets, reads, writes to the json file

read_write.py

```python
import json

# this class will read and write to a json file (store data)


class ReadWrite:
    def __init__(self, file_path):
        self.file_path = file_path

    def get_json_list(self):
        input_file = open(self.file_path)
        json_list = json.load(input_file)
        return json_list

    def write_item_to_json(self, item):
        input_file = open(self.file_path)
        json_list = json.load(input_file)
        json_list.append(item)
        with open(self.file_path, 'w') as my_file:
            json.dump(json_list, my_file)

    def write_list_to_json(self, list):
        with open(self.file_path, 'w') as my_file:
            json.dump(list, my_file)

    def reset_json(self):
        with open(self.file_path, 'w') as my_file:
            json.dump([], my_file)
```

MY WARDROBE
APP CODE

Read_write.py
With Error handling

```python
import json

# this class will read and write to a json file (store data)


class ReadWrite:
    def __init__(self, file_path):
        self.file_path = file_path

    def get_json_list(self):
        try:
            input_file = open(self.file_path)
            json_list = json.load(input_file)
            return json_list
        except:
            raise IOError(
                "There was an issue reading your list.json file. Please make sure it exists and is in the correct format.")

    def write_item_to_json(self, item):
        try:
            input_file = open(self.file_path)
            json_list = json.load(input_file)
            json_list.append(item)
            with open(self.file_path, 'w') as my_file:
                json.dump(json_list, my_file)
        except:
            raise IOError(
                "There was an issue reading your list.json file. Please make sure it exists and is in the correct format.")

    def write_list_to_json(self, list):
        try:
            with open(self.file_path, 'w') as my_file:
                json.dump(list, my_file)
```

Read_write.py
With Error handling

```
34          except:
35              raise IOError(
36                  "There was an issue reading your list.json file. Please make sure it exists and is in the correct format.")
37
38      def reset_json(self):
39          try:
40              with open(self.file_path, 'w') as my_file:
41                  json.dump([], my_file)
42          except:
43              raise IOError(
44                  "There was an issue reading your list.json file. Please make sure it exists and is in the correct format.")
```

- IOError - if json file is missing from folder this Error message will run and app will close.

MY WARDROBE
REVIEW

Challenges:

1. Where do I start :-)
2. How to store data so that it will be saved when user runs app again.
3. Keep it simple!
4. Error issues working out what to include here

Favourite Parts:

1. Creating my own App
2. Developing my coding skills in Python.
3. Using classes
4. Successfully running my app
5. Resolving error issues
6. Making the app have more user appeal with the use of colored text


THE END