Superpy User Manual

Introduction
Welcome to Superpy, your all-in-one inventory and sales management tool. Superpy helps you effortlessly manage your inventory, track sales, and generate insightful reports for better decision-making. This user manual provides a comprehensive guide on how to use the various features of Superpy

Getting Started

Installation
Superpy does not require a formal installation process. Ensure that you have Python3 installed on your system. Download the Superpy script files and save them in a directory of your choice. Superpy does however use some non standard packages that need to be downloaded to enable the full Superpy experience. These are Rich and Reportlab. If you do not have these installed you can use pip install to downlaod them via your terminal.
Commands:
pip install rich
pip install reportlab

Running Superpy
Open a terminal of your preference or command prompt, navigate to the directory where Superpy is saved, and execute the following command:

```
python3 main.py
```

This will launch the Superpy application. After this command nothing visible will happen. In order to have your first visible interaction with Superpy execute the help command, which is the following:

```
python3 main.py -h
```

The -h command is also handy to keep in for later. If you get stuck after selecting a command you can add -h to get more information. In the next section we'll discuss the commands and operations that Superpy is able to perform.

Commands and Operations

Superpy supports three main types of commands: report, operation, and time. Each command type has a number sub-commands and options. Below you can find a quick overview on how to execute the different commands and operations. [] means it is not mandatory to use/specify. Before executing any of these commands you should always call Superpy by typing `python3 main.py`.

Reports
- Inventory Report: `report inventory [--date DATE]`
- Sales Report: `report sales [--date DATE]`
- Product Report: `report product`
- Expired Report: `report expired`
- Revenue Report: `report revenue [--date DATE]`
- Profit Report: `report profit [--date DATE]`
- Monthly Report: `report monthly --period PERIOD`

Operations
- Buying Inventory: `operation buy --product_name NAME --buy_price PRICE --expiration_date DATE`
- Selling Inventory: `operation sell --product_name NAME --sell_price PRICE`
- Changing Inventory Data: `operation change --id ID --product_name NAME --buy_date DATE --buy_price PRICE --expiration_date DATE`
- Deleting Inventory: `operation delete --id ID`
- Removing Obsolete Inventory: `operation obsolete`

Time
- Setting Time: `time set --date DATE`
- Advancing Time: `time advance --days DAYS`
- Current Time: `time current`

Lets dive a little deeper into the different operations each main command can provide.
Hint : most arguments used in theses operations also have an abbreviated version.
reports:
date: --date or -d
period: --period or -p
operations:
product_name: --product_name or -pn
buy_price: --buy_price or -bp
sell_price: --sell_price or -sp
expiration_date: --expiration_date or -ed
id: --id
buy_date: --buy_date or -bd
time:
date: --date or -d
days: --days or -ds

<div align="center">

Reports

</div>

Inventory Report
Generate a report of the current inventory or items in the inventory up to a specific date.

```
python3 main.py report inventory --date 2023-11-03
```

The --date is optional, if used it has to have the following format YYYY-MM-DD. If left out you will get a report on the entire inventory ledger.

Sales Report
Generate a report of sales for the entire sales ledger or items in the sales ledger up to a specific date.

```
python3 main.py report sales --date 2023-11-03
```

The --date is optional, if used it has to have the following format YYYY-MM-DD. If left out you will get a report on the entire sales ledger.

Product Report
Get a report of the quantity of each product in the current inventory.

```
python3 main.py report product
```

Expired Report
Get a report of items in the inventory that have expired.

```
python3 main.py report expired
```

Revenue Report
Generate a report of total revenue form the entire sales ledger or a specific date.

```
python3 main.py report revenue --date 2023-11-03
```

--date is optional, if used it has to have the following format YYYY-MM-DD. If left out you will get a report on the total revenue.

Profit Report
Generate a report of gross profit form the entire sales ledger or a specific date.

```
python3 main.py report profit --date 2023-11-03
```

--date is optional, if used it has to have the following format YYYY-MM-DD. If left out you will get a report on the total revenue.

Monthly Report
Generate a detailed monthly report including revenue, profit, and other key metrics.

```
python3 main.py report monthly --period 2023-11
```

The period should be in the following format YYYY-MM.

Operations

Buying Inventory
Buy a new product and add it to the inventory.

```
python3 main.py operation buy --product_name apple --buy_price 1 --expiration_date 2023-11-31
```

Expiration date should be in the following format YYYY-MM-DD.

Selling Inventory
Sell a product from the inventory.

```
python3 main.py operation sell --product_name apple --sell_price 2
```

Changing Inventory Data
Modify the data of a specific inventory item.

```
python3 main.py operation change --id 14 --product_name apple --buy_date 2023-11-14 --buy_price 2 --expiration_date 2023-11-31
```

Make sure to write the dates in the following format YYYY-MM-DD.

Deleting Inventory
Delete a specific item from the inventory based on the inventory id.

```
python3 main.py operation delete --id 1
```

Removing Obsolete Inventory
Move expired products to the obsolete inventory ledger. This will ensure that the buying costs are included in the monthly costs and thus profit calculation. Next it makes sure the inventory is only contains sellable products.

```
python3 main.py operation obsolete
```

Time Operations

Setting Time
Set the internal date of Superpy to a specific date.

```
python3 main.py time set --date 2023-11-03
```

Date should be in the following format YYYY-MM-DD.

Advancing Time
Advance the internal date of Superpy by a specified number of days.

```
python3 main.py time advance --days 7
```

Current Time

Get the current internal date of Superpy.

```
python3 main.py time current
```

Conclusion

Congratulations! You have completed the Superpy user manual. Explore the various reports and operations to efficiently manage your inventory and sales. If you have any questions or encounter issues, refer to this manual.