COM709 FUNDAMENTALS 2020 Assignment

AE1 Part 1

andy September 17, 2020

1. **Write an online data management application**

# Instructions

* + - You must follow the functional requirement specifications
    - You must follow the non-functional requirement specifications
    - You must deliver the product on time for the deadline
    - You must deliver all parts of the product
    - You must deliver the product in the specified form

# Delivery date

* + - Fri October 2nd 2020

# Functional requirements

* + - Program is to create an online data retrieval system for stock keeping and prices
      * Items would be like:

Name ID Price In Stock

|  |  |  |  |
| --- | --- | --- | --- |
| Soap | 0012367345 | 1.99 | 100 |
| Chocolate | 1998700374 | 0.99 | 40 |

Batteries 6476398475 2.50 60

* + - Program must:
      * Implement a CRUD protocol for text based data store
      * Program must show start screen/menu
    - Must expose menu driven functions to:
      * Help (explain how to use the app)
      * Create (add new) new entry
      * Retrieve (and show) one or all records
      * Update an existing record
      * Delete a record
    - Data records will contain the following information:
      * A product name: A string of up to 20 characters, limited to upper case ASCII
      * Product code: A string representing a padded integer of 10 digits from 0000000000 to 9999999999
      * A price: A positive float
      * Number in stock: A positive integer

# Data store requirements

* + - You may choose ANY of the following storage methods
      * A csv file
      * a flat text file of your own format
      * a SQLite database
    - When first launched the program should look in its directory for an existing data store, (this will be called data.csv, or data.sql or data.txt as examples). If a data file does not exist then your program should create a new data store in its directory.

# Optional requirements (extra marks)

* + - You can implement a search for item names or product codes if you want Otherwise just print them **all** in a list
    - You can make the program network enabled - accessible via a socket if you like. Use port 8080 of localhost. Otherwise interaction via command line menus will do.

# Non Functional (constraints)

* + - Program should be docstring commented and examinable via pydoc
    - Program is to be written in Python >= v.3.20
    - Program must ONLY use specified libraries
      * socket, sqlite3
      * buitins (os, sys, csv etc)
    - Program must execute from the command line
    - Program must be self-contained (apart from the data store)
      * a single python file
      * it should not take any command line options
      * no other config files or dependencies necessary

# Delivery requirements

* + - The program must be named **datamanager.py**
    - it should be the only .py file in a **.zip** archive
    - you can also include an example data file
    - the .zip file must be named as your student number
      * for example: Q1234567.zip

# Marking Criteria

* + - Is in ZIP file with correct student number
    - Folder structure is correct
    - Well commented source code
    - A sample data set is provided
    - Program executes
    - Program presents initial interface screen/banner
    - Help menu works and is meaningful
    - Add item works and does proper data validation
      * bounds checking
      * existing item name/ID
    - List items works
      * All items listed
      * BONUS: Items can be searched by name/product ID
    - Items can be updated (with validated data)
    - Items can be deleted
    - Clean program Exit
    - BONUS FEATURE: Network interface on port 8080

# Example Usage Session

$ python3 datamanager.py

> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * \*\*\* WELCOME TO DATAMANAGER \*\*\*   > \*\*\* Menu: \*\*\* | | | | | | |
| > \* | 1 | List All | 4 | Delete | ID | \* |
| > \* | 2 | List ID | 5 | Update | ID | \* |
| > \* | 3 | List Name | 6 | Add |  | \* |
| > \* | 7 | Exit |  |  |  | \* |
| > \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  > | | | | | | |
| > 1 | |  | |  |  | |
| * Peanuts | | 4763527385 | | 2.49 | 30 | |
| * Bananas | | 3452948757 | | 0.30 | 63 | |
| * Beans | | 3564958777 | | 0.78 | 34 | |
| > 2 | |  | |  |  | |
| * ID? | | 3452948757 | |  |  | |
| * Bananas | | 3452948757 | | 0.30 | 63 | |
| > 6 | |  | |  |  | |
| * Name? Coconut   > ID? 4763527385 | | | | | | |

>

>

* ERROR ID EXISTS

> ID? 3423413567

> PRICE? 1.00

* ADD STOCK? 10
* OK
* 3
* NAME? Coconut
* Coconut 4763527385 1.00 10
* 5
* ID? 4763527385
* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* \*\*\* Update Menu: \*\*\*
* \* 1 Name 3 In Stock \*
* \* 2 Price 4 Exit \*
* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* 4
* Update aborted
* 7
* Bye

$