THE UNIVERSITY OF BRITISH COLUMBIA, OKANAGAN

FACULTY OF SCIENCE

COSC 310 Software Engineering

Dr. Shan Du

Lab Section L01, Team 5

Group Assignment 3 Programming Project

Report by:

Saksham Rana

Paul Unger

Ken Woon

14th November 2022

Table of Contents

Project Description **3**

Data Flow Diagrams **4**

Level 0 DFD 4

Level 1 DFD 4

Limitations **5**

API Feature List **5**

Software Outputs **6**

Example 1: Cloud Backup 6

Example 2: Change Password 7

Example 3: Inventory Analysis 8

Example 4: Savestate and encryption 9

Project Description

Our project is an inventory management system where product inventories will be tracked in terms of how fast they are being sold and how much time it takes to order more. Store managers will be notified when the system detects a product running out of stock and needs replenishment. The project is hosted on GitHub with our team account here.

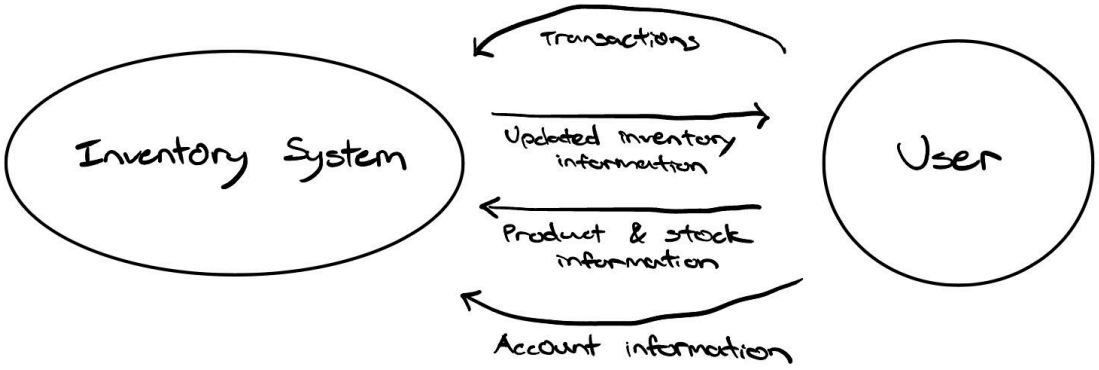
The motivation for this topic comes from the Mentcare case study provided in the case study list from Assignment 1. In the document, patient data was stored in a database where clinic staff could add and edit information related to patients, as well as view historical data for the patient. The information in the database was regularly monitored to ensure that all patients were still attending their appointments and taking their prescribed medication. Additionally, the system would generate monthly reports showing patients, their prescribed medicines, costs related to drugs, and more. Because the patient information is all sensitive and protected by the UK Data Protection Act, data was protected and only authorized users could access the data.

Several of these traits are similar to how our proposed inventory system will need to be designed. Our inventory system must allow for data entry and historical data viewing to help facilitate a manager's decision to order more stock of a particular product. Additionally, the database will have to be regularly monitored to check if the inventory trends of a specific product are going down too fast, indicating we need to warn the manager that we need more stock of that product. Because the database will have to be scanned regularly to generate these warnings and reports, the system must be fast and reliable. However, unlike the Mentcare case study, we are unlikely to have multiple users accessing the database simultaneously, so we don't need to worry about concurrency with our design. Like the Mentcare case study, our system will have to have the data protected as we don't necessarily want an employee who's just working the cash register to have access to all the analytical information related to the inventory. Instead, that employee should be able to see just the current inventory stock information to answer a customer's question about if an item is in stock. Finally, like Mentcare, our system will likely be used by users who may not be familiar with these types of plans, meaning ease of use and user accessibility is something we need to keep in mind when designing the user interface.

Data Flow Diagrams

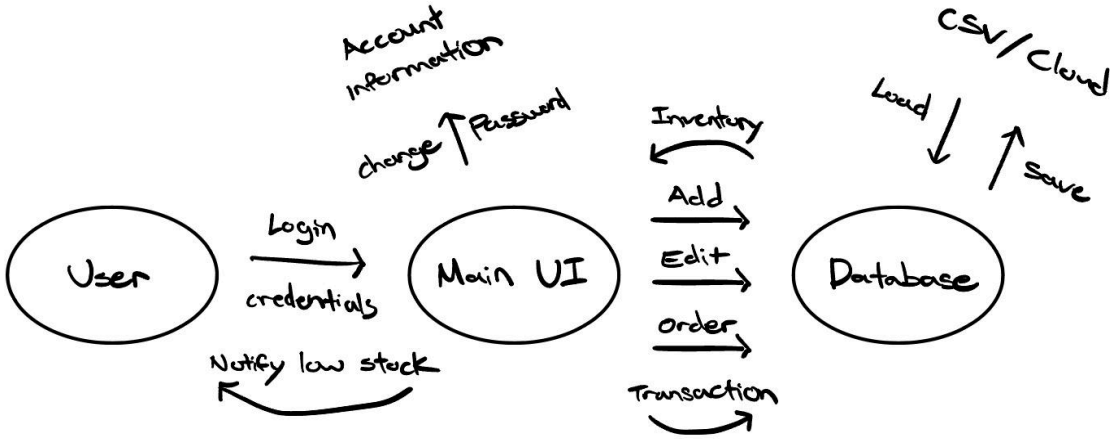
Level 0 DFD

The level 0 data flow diagram is an abstract view of the system shown as single processes with its relationship to external entities. The user of the system is first required to enter their account information in order to log in to have access to the database. The user then enters the product and stock information manually or by importing an external dataset. The inventory system will ultimately show the inventory information to the user and will be updated as the user enters transaction records into the system.



Level 1 DFD

The level 1 data flow diagram highlights the main functions of the system and breakdowns the high-level process of 0-level DFD into subprocesses. After the user enters the login credentials, they will be brought to the Main UI of the inventory system, which will display the current working dataset. From this UI, the user will be able to add, edit, order, and make a transaction, which will update the database. The database can be loaded from and saved as a CSV file or to the cloud. The account information can be edited from the Main UI and also notifies the user when the stock of an item becomes low.



Limitations

- Because the savestate file is stored in a static location, by deleting it users can gain access to the system's default state (with dummy data) - although crucial data is stored in the savestate, so it's not necessarily a security risk
- The database is able to be stored in a public cloud but without any security. This means that the data can be accessed by anyone with the link.

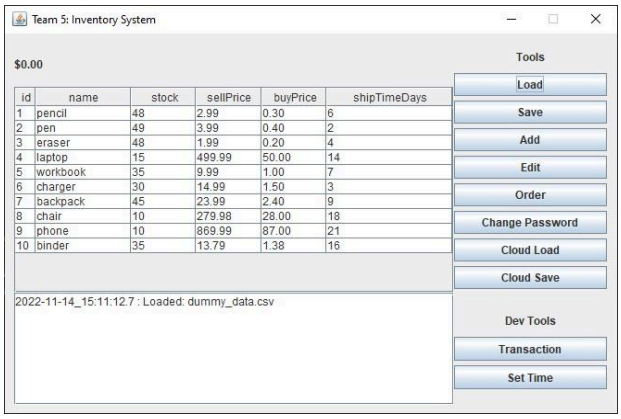
API Feature List

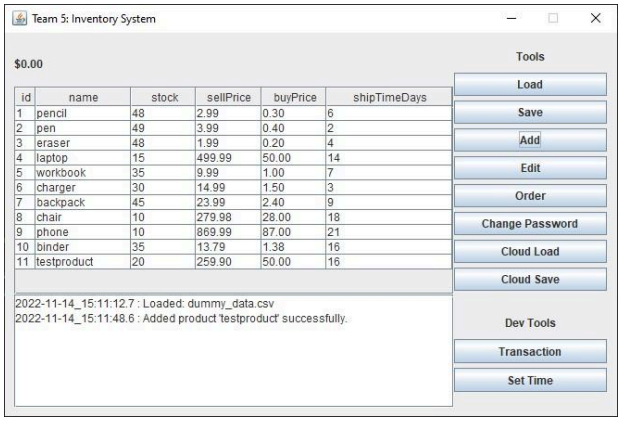
- Encryption/decryption of any generic text file functionality
- Extendable savestate functionality
- Cloud backup functionality
- Database, Product, and Order backend classes
- Some re-usable complex GUI elements such as the pop-up dialogues for checking the password, adding a product to the database, etc

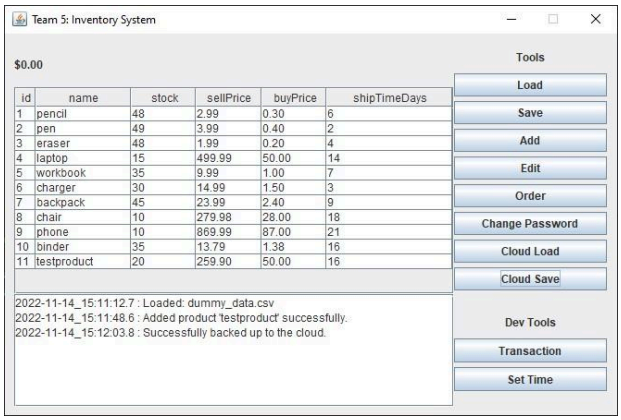
Software Outputs

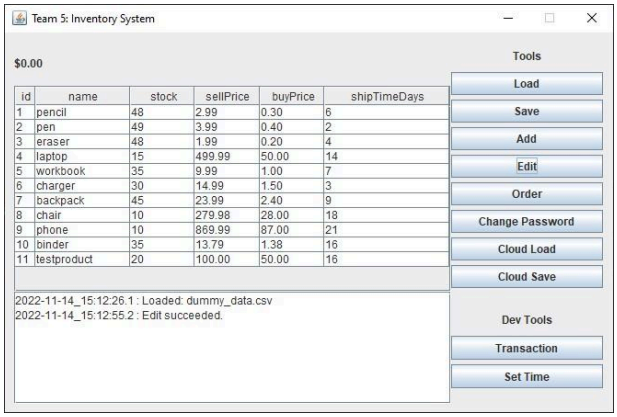
Example 1: Cloud Backup

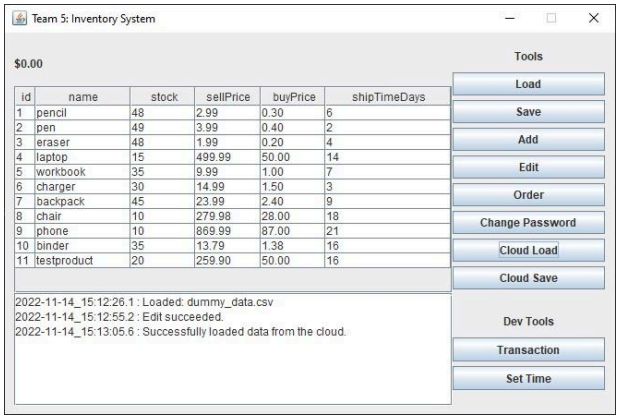
These 5 screenshots show the process of adding a test product, backing it up to the cloud, closing the program, re-opening the program, editing a product to have a much lower sellPrice of $100, then loading data from the cloud backup. The cloud can be accessed here.





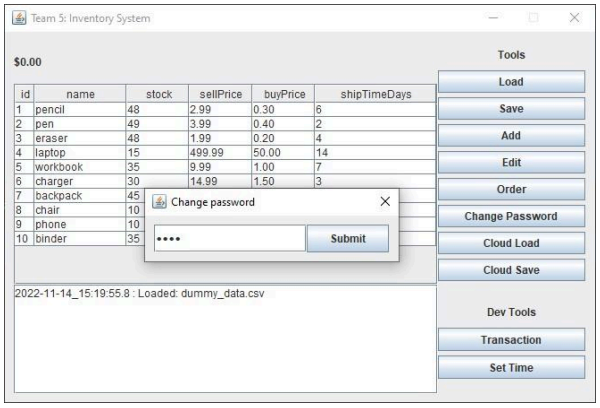


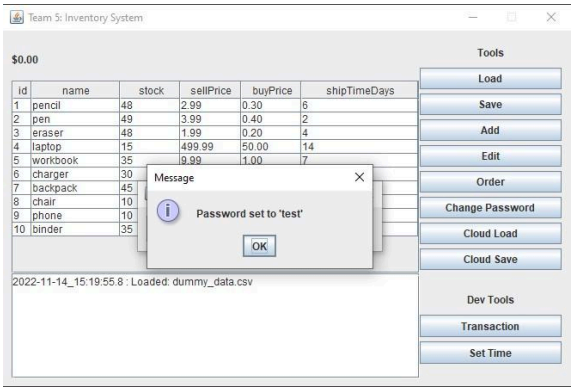


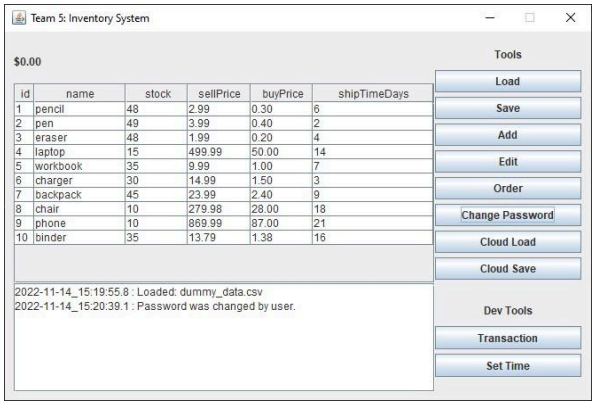


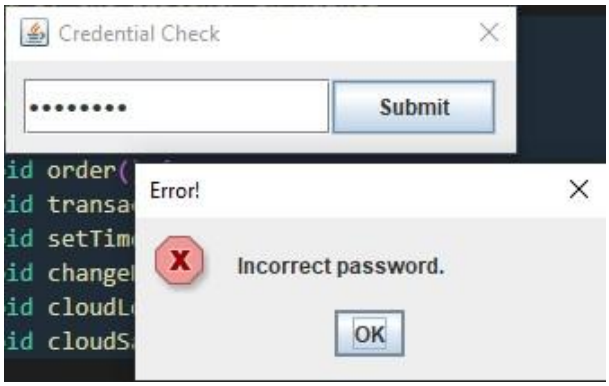
Example 2: Change Password

These 4 screenshots show the process of clicking the change password button, typing ‘test’, closing the program, re-opening the program, and attempting to login using the old default password of ‘password’.



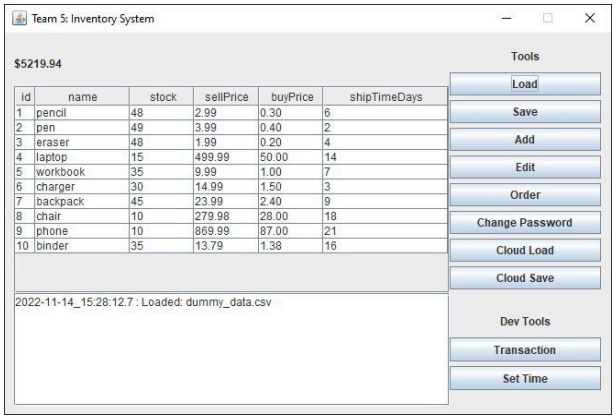


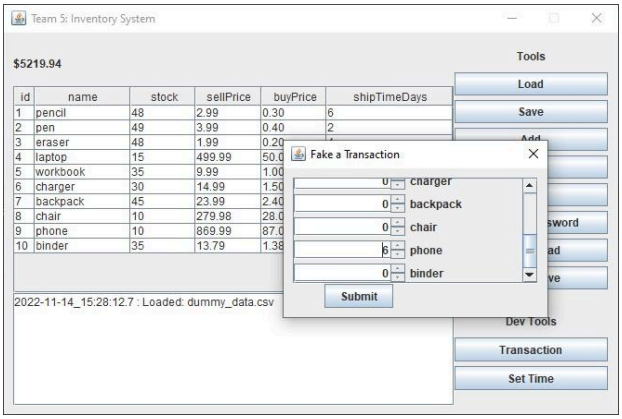


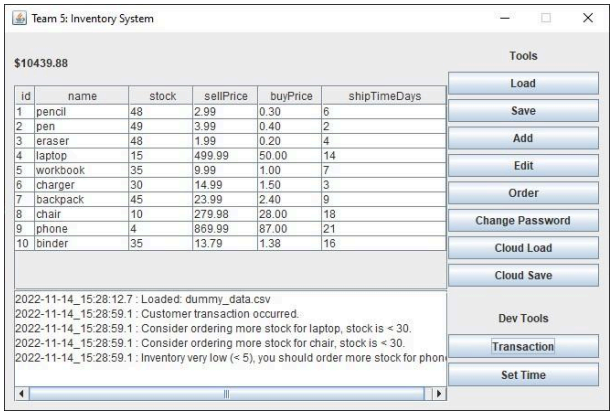


Example 3: Inventory Analysis

These 3 screenshots show the process of clicking ‘Transaction’, simulating a customer buying 6 phones, and the resulting notifications to buy more of specific products.

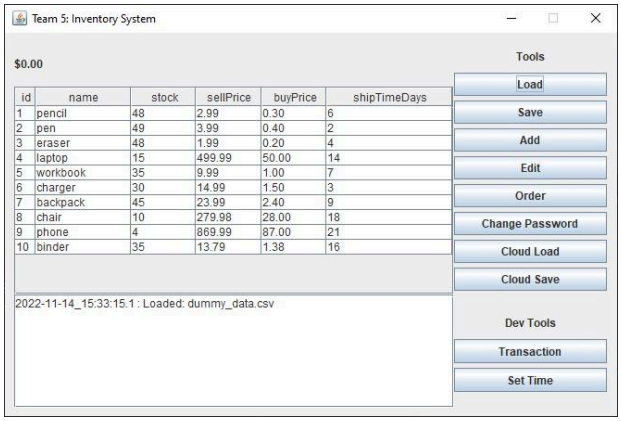


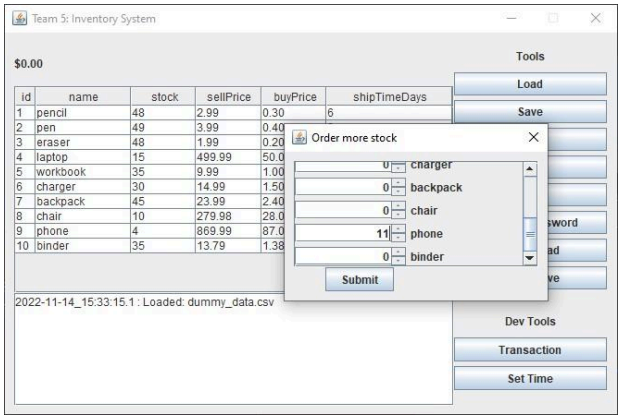


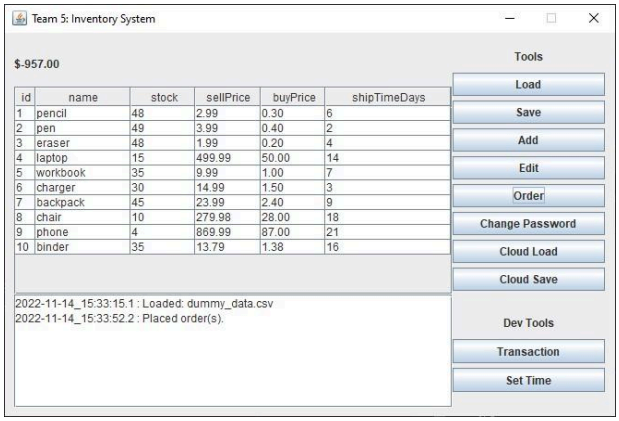


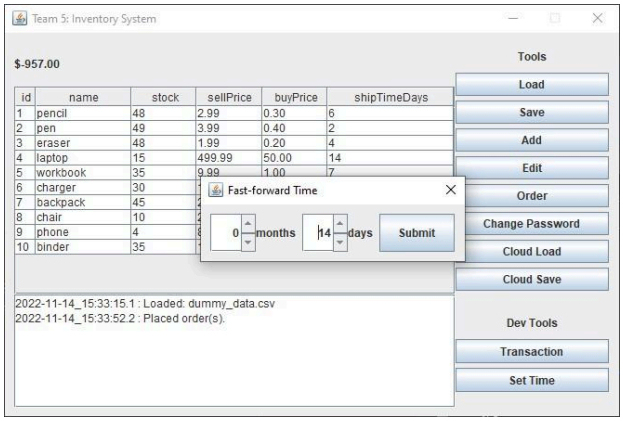
Example 4: Savestate and encryption

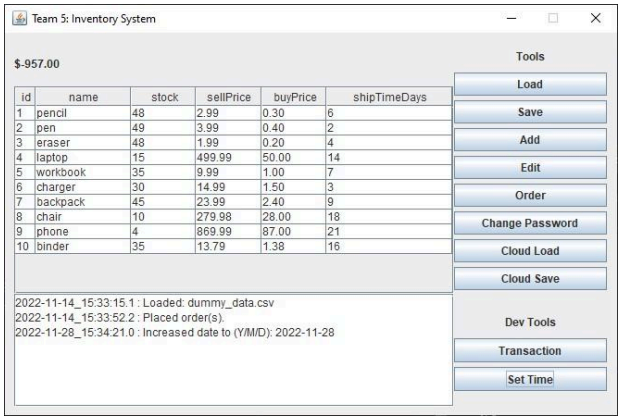
These 8 screenshots show the process of placing an order for 11 phones, simulating the passing of 2 weeks of time, closing and re-opening the program, and simulating the passage of another 2 weeks. Notice how the revenue, date, and orders placed were saved between runs of the program.

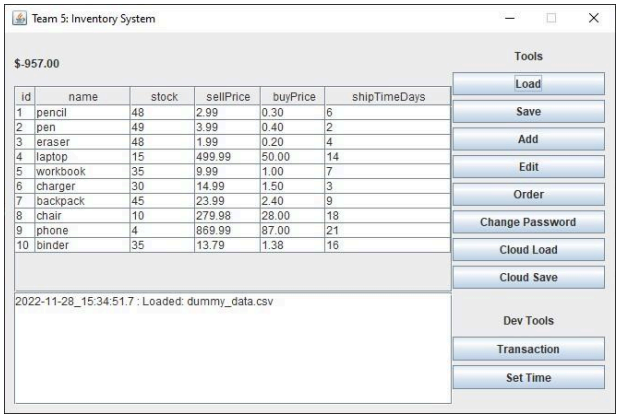


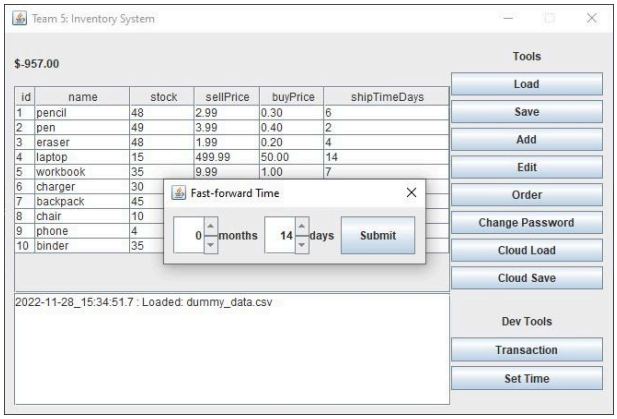


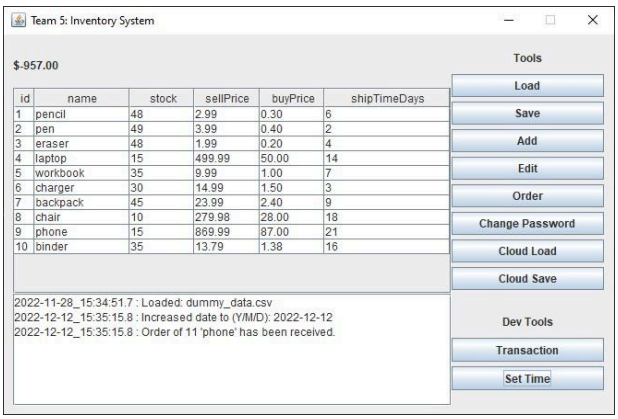





Team 5: Inventory System
$-957.00
Tools
Load
Save
Add
Edit
Order
Change Password
Cloud Load
Cloud Save
Dev Tools
Transaction
Set Time
id name stock sellPrice buyPrice shipTimeDays
1 pencil 48 2.99 0.30 6
2 pen 49 3.99 0.40 2
3 eraser 48 1.99 0.20 4
4 laptop 15 499.99 50.00 14
5 workbook 35 9.99 1.00 7
6 charger 30 14.99 1.50 3
7 backpack 45 23.99 2.40 9
8 chair 10 279.98 28.00 18
9 phone 4 869.99 87.00 21
10 binder 35 13.79 1.38 16
2022-11-14_15:33:15.1 : Loaded: dummy_data.csv
2022-11-14_15:33:52.2 : Placed order(s).
2022-11-28_15:34:21.0 : Increased date to (Y/M/D): 2022-11-28


Team 5: Inventory System
$-957.00
Tools
Load
Save
Add
Edit
Order
Change Password
Cloud Load
Cloud Save
Dev Tools
Transaction
Set Time
id name stock sellPrice buyPrice shipTimeDays
1 pencil 48 2.99 0.30 6
2 pen 49 3.99 0.40 2
3 eraser 48 1.99 0.20 4
4 laptop 15 499.99 50.00 14
5 workbook 35 9.99 1.00 7
6 charger 30 14.99 1.50 3
7 backpack 45 23.99 2.40 9
8 chair 10 279.98 28.00 18
9 phone 4 869.99 87.00 21
10 binder 35 13.79 1.38 16
2022-11-28_15:34:51.7 : Loaded: dummy_data.csv


Team 5: Inventory System
$-957.00
Tools
Load
Save
Add
Edit
Order
Change Password
Cloud Load
Cloud Save
Dev Tools
Transaction
Set Time
Fast-forward Time
0 months
14 days
Submit
id name stock sellPrice buyPrice shipTimeDays
1 pencil 48 2.99 0.30 6
2 pen 49 3.99 0.40 2
3 eraser 48 1.99 0.20 4
4 laptop 15 499.99 50.00 14
5 workbook 35 9.99 1.00 7
6 charger 30
7 backpack 45
8 chair 10
9 phone 4
10 binder 35
2022-11-28_15:34:51.7 : Loaded: dummy_data.csv


Team 5: Inventory System
$-957.00
Tools
Load
Save
Add
Edit
Order
Change Password
Cloud Load
Cloud Save
Dev Tools
Transaction
Set Time
id name stock sellPrice buyPrice shipTimeDays
1 pencil 48 2.99 0.30 6
2 pen 49 3.99 0.40 2
3 eraser 48 1.99 0.20 4
4 laptop 15 499.99 50.00 14
5 workbook 35 9.99 1.00 7
6 charger 30 14.99 1.50 3
7 backpack 45 23.99 2.40 9
8 chair 10 279.98 28.00 18
9 phone 15 869.99 87.00 21
10 binder 35 13.79 1.38 16
2022-11-28_15:34:51.7 : Loaded: dummy_data.csv
2022-12-12_15:35:15.8 : Increased date to (Y/M/D): 2022-12-12
2022-12-12_15:35:15.8 : Order of 11 'phone' has been received.