GUI Inventory Management System

with JSON and SQL file processing and PDF exports

Developing report

Group 0000000 Gyrfalcon
Yang Xikun 23101291d,
Jin Qinhao 23107919d,
Jin Yixiao 23101231d,
Ren Yixiao 23102145d

Nov 27, 2023

Contents

Introduction

Our project focuses on how we can facilitate storage managers to manage their products and transactions and keep track of the latest information about each product.

The project would first store comprehensive information for each product including the prices, quantities, and supplier details. A unique ID code for each product is required so users can retrieve the product and its information. Users can make transactions on existing products as well.

The program supports the deletion, addition, and modification of product information by retrieving the ID. All the data is stored in a JSON file or a SQL database where you can view all the information about each product as well as the transaction history.

Users can also export their data to a PDF report.

Objectives

1. Design the structure of the inventory file and implement the function of product managing, including retrieving existing product information, adding and updating product information to the inventory system, and deleting products from the inventory system.

a. Design the structure of the inventory file for product details, including the product name, price, quantity, supplier details, and the numeric code used as a unique identifier for each product.

b. Batch operation: 1) load inventory files for processing, 2) display the existing product information on the output device; and 3) dump inventory data from memory to inventory files.

c. Individual record operations: within the inventory file, locate a specific record by product's unique identifier, and 1) read the record and/or show it, 2) delete the record from the inventory file, 3) modify the record and save it to inventory file according to user input (the information can be modified including product price changes, restocking quantities, and supplier details).

2. Design the structure of the sales transaction and implement the function of Sales tracking.

a. Design the structure for sales transaction information including the product sold quantity, customer information, and sales date.

b. Receive the user input of the sales transaction, save it to the sale transaction file, and then update the inventory quantity to the inventory file accordingly.

c. Read records from the sale transaction file and display the sales transactions.

3. Integrate a database (MySQL) to store and retrieve data.

a. Create a MySQL database for data storage, including: 1) database table design according to the structures of inventory and sale transaction file, scripts of table creating, 2) setup of MySQL database, 3) configuration of the database including user name and password, database name and so on, 4) creating of instance of database and tables.

b. Write SQL files for addition, deletion, change, and query of database operating, to store and retrieve inventory and sale transactions data.

c. Write a system configuration file to set up the usage mode of the database: we can choose to use a database or file for data storage.

4. Statement of Inventory and Sales:

a. Inventory:

Generate a comprehensive report of stored items and their detailed information, including, 1)Inventory list: provide a complete inventory list, including product name, specification, model, quantity, price, and other information. 2） Inventory status analysis: classify the statistics of the inventory items, and analyze the inventory, purchase cycle, and sales situation of all kinds of items, to understand the inventory status. 3） inventory warning prompt: according to the inventory status analysis results, set the inventory warning line, when the inventory is lower than a certain quantity, automatically remind the user to purchase in time or adjust the sales strategy. 4)Inventory optimization suggestions: According to the inventory status and sales situation, provide inventory optimization suggestions, including reasonable allocation of inventory, inventory cost reduction, etc.

b. Sales

Generate a comprehensive sales report including sales targets and profit and loss statements, including, 1)Sales data statistics: statistical sales data of each period, including sales volume, sales volume, customer unit price, and other indicators. 2) Sales target achievement situation: According to the sales data and sales target, analyze the target achievement situation, including the sales volume achievement rate, sales volume achievement rate, and other indicators.3)Sales analysis report: analyze the sales data, including sales trends, hot-selling products, customer groups, etc., to understand the sales status and market demand. 4)Income statement: based on the sales data and cost data, generate an income statement, including revenue, cost, gross profit, net profit, and other indicators, to understand the profitability of the sales business.

5. Implement a graphic user interface (GUI) to enhance the visual appeal of the program.

a. Choose PyQt as the GUI library and learn how to use it to create a user interface.

b. Design and implement a user-friendly interface that includes elements such as buttons, text boxes, etc., so that users can interact with the program.

c. Provides options on the interface to perform inventory and sale transactions management operations, such as adding, updating, deleting products, and viewing records of inventory data and sale transactions.

To meet the objectives above, our group will first design the overall framework and classes of the project, such as products, transactions, and inventory management systems. After implementing the basic functions of the above classes, we will then implement the GUI and database functions, and finally refine the code after testing.

Project Design

Our project is to create an inventory management system, processing products, and transactions and relating them to multiple file types. To achieve this, we divided our project into 4 modules.

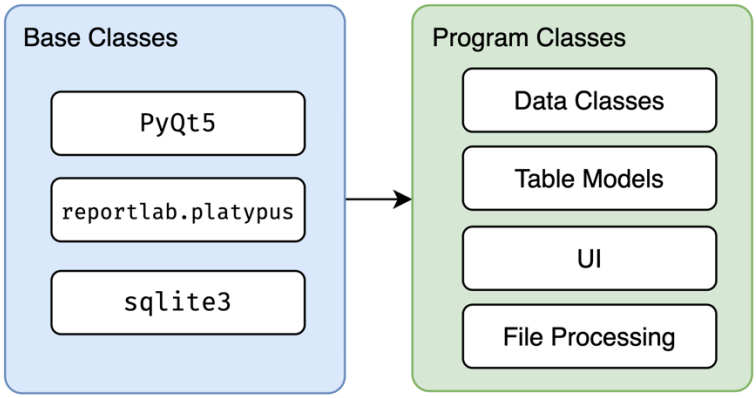
The first module is **Data Classes + File Processing**, which processes the inventory data, especially users' input (including creating products, editing products, deleting products, and making transactions). This module is also responsible for encoding/decoding the inventory data to/from JSON or SQL files, as well as exporting it to PDF files.

The second module is **UI**, which creates the main UI layouts, including windows, UI components, and message boxes (such as error alerts, and success alerts).

The third module is **Table Models**, which relates the inventory data to the UI layouts. The products and transactions data are organized in classes, therefore we need a block of code to make the table (a UI component, showing the data in a table) understand the data in the classes.

More detailed about the GUI, the main window consists of a sidebar, including buttons: *Add a new product*, *Edit a product*, *Delete a product*, *Make a new transaction*, etc. By clicking these buttons and following the instructions, the product details window, transaction details window, product ID searching window, and file processing window will be shown correspondingly.

We use PyQt to create UI layouts, reportlab.platypus to generate PDF files, and SQLite3 to process SQL databases. Hence these libraries include the base classes of our project. The program classes are responsible for the project, and many of them are inherited from the base classes. The hierarchy diagram is roughly shown below:



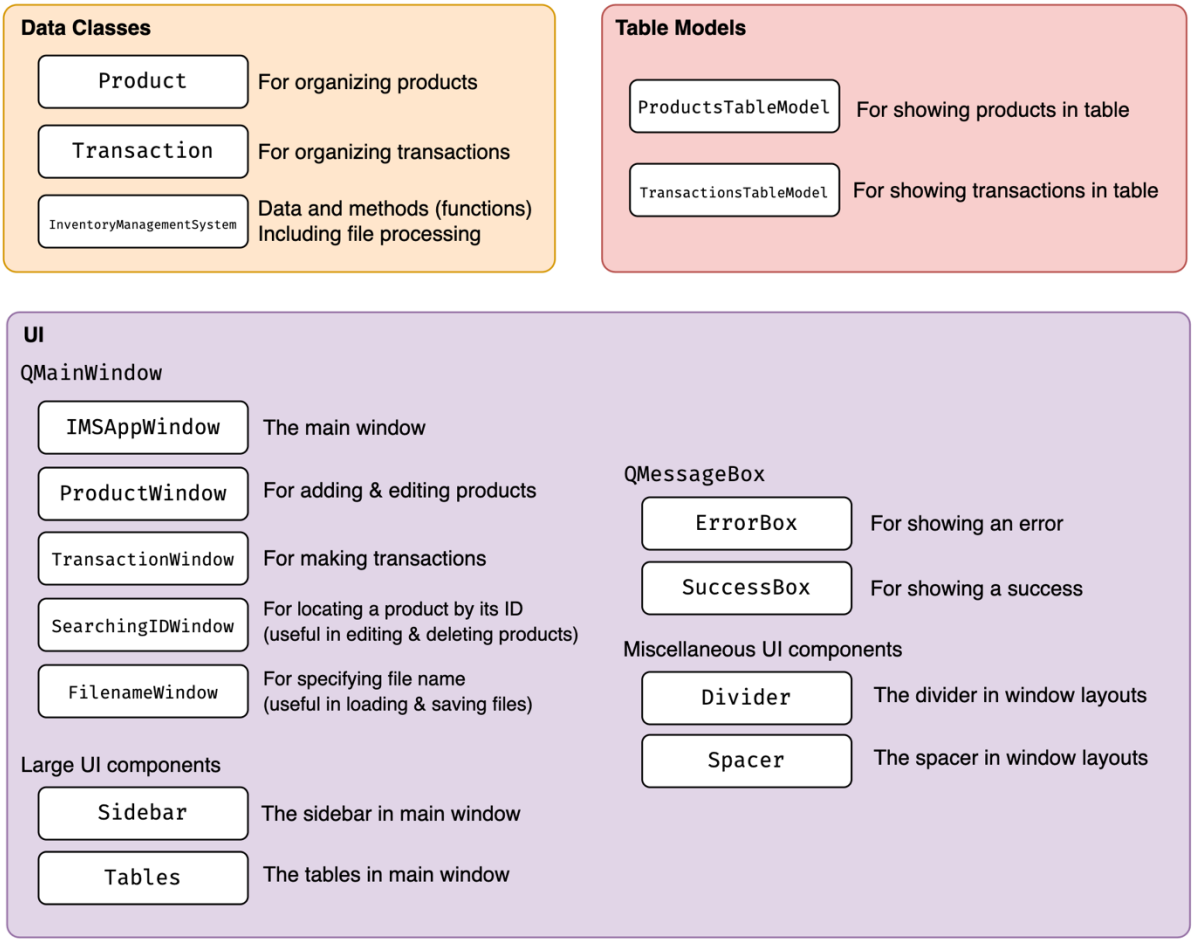
Implementation Details and Libraries Used **(our challenges are underlined)**

1. Object-oriented. We used classes to organize both UI layouts and inventory data. We built the following classes:

a. `Product` for organizing products, `Transaction` for organizing transactions, `InventoryManagementSystem` for processing inventory data;

b. `Divider`, `Spacer` for corresponding UI components, `ErrorBox`, `SuccessBox` for error/success alerts, `ProductTableModel`, `TransactionTableModel` for showing tables; `Tables`, `Sidebar` for the main layouts, `IMSAppWindow` for the main window.

They and their categories are shown below:

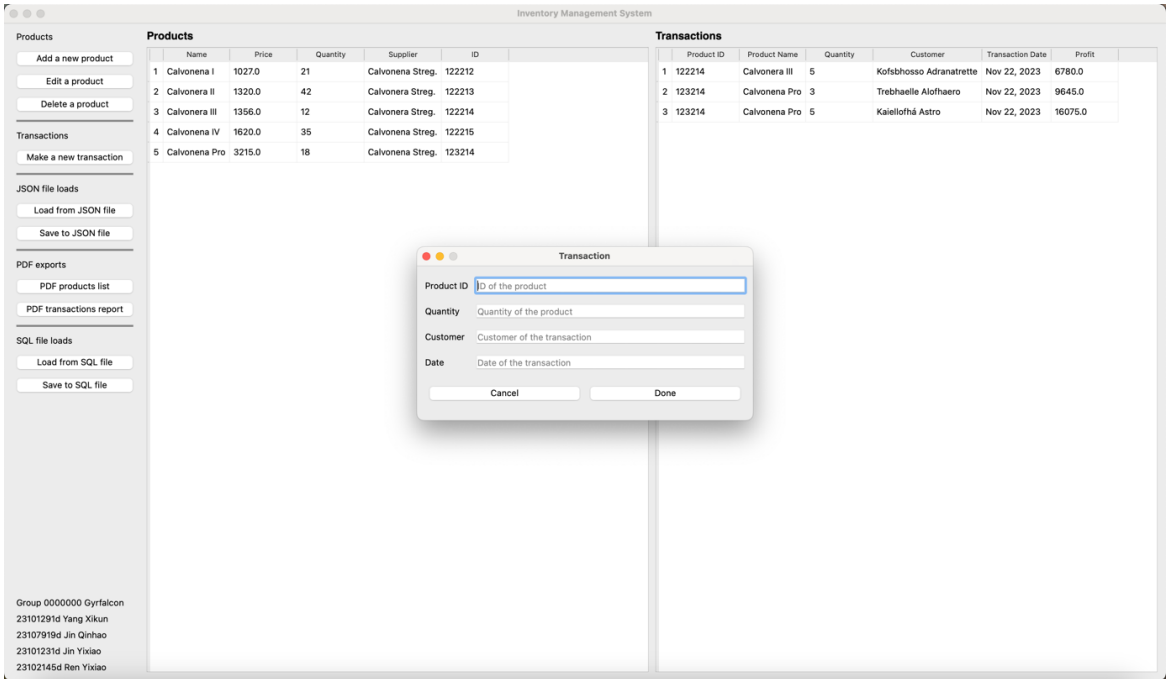


2.1. Signal-slot system. A `QtCore.pyqtSignal()` is declared for sending signals. When the inventory data changes, we use `emit()` to send out the signal. A receiver in the `Tables` class receives the signal and then repaints the tables.

This part is challenging. We spent hours coming up with this idea and checked PyQt documentation to implement it. Refreshing GUI components are always challenging, in every language.

2.2. Event filter. We used `keyPressEvent` to sensor the key-pressing events. When the *Enter* key is being pressed, some features may be activated.

3. Graphic user interface. We used `PyQt5` to build our GUI views. Windows consists of layouts, widgets, and components. We adjusted the layouts for a long time, trying to make them friendly to all our customers.



4. Polymorphism. The `FilenameWindow` class is capable of processing both JSON files, PDF files, and SQL files. The `FilenameWindow` constructor requires a parameter `mode`, which is allowed to be `load_json, save_json, load_sql, save_sql, pdf_export_products_list` and `pdf_export_transactions_report`.

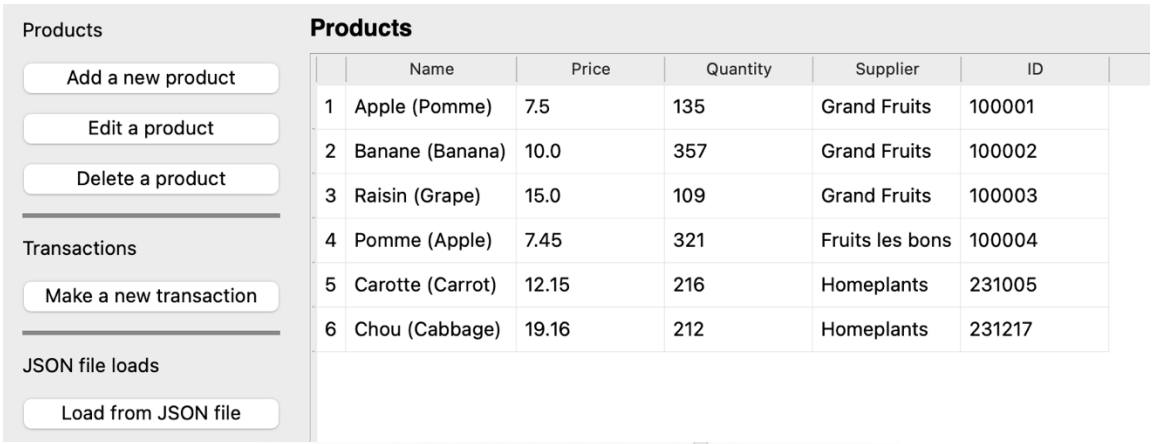
5. Database support. Expressions such as **`CREATE TABLE IF NOT EXISTS`** `products…` are for SQL file processing. Please take a view of our source codes for detailed information. We checked the documentation of SQLite and implemented the SQL file-processing feature for a long time. SQLite SQL file is binary, and we debugged this mainly through our Python code.

We used `PyQt5`, `reportlab.platypus`, and `SQLite3` to implement our project.

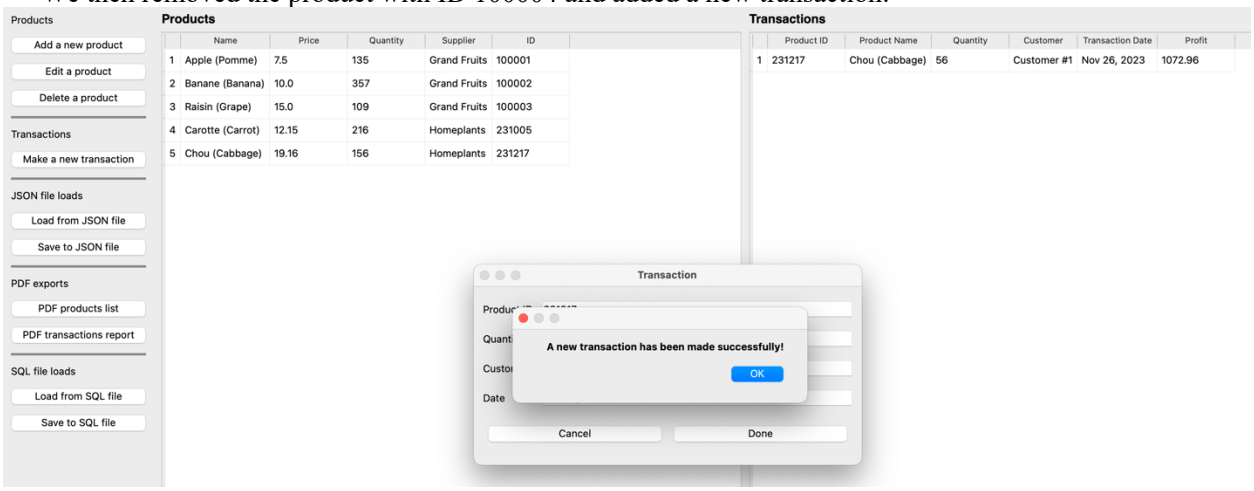
Testing and Results

We added the following products to the system and then saved them to `main_test_json.json`.

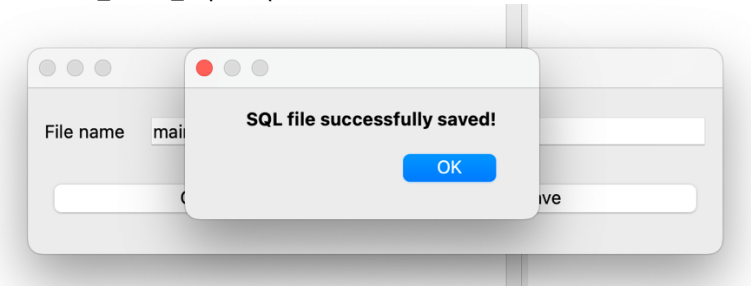
Name	Price	Quantity	Supplier	ID
Pomme (Apple)	7.5	135	Grand Fruits	100001
Banane (Banana)	10	357	Grand Fruits	100002
Raisin (Grape)	15	109	Grand Fruits	100003
Pomme (Apple)	7.45	321	Fruits les bons	100004
Carotte (Carrot)	12.15	216	Homeplants	231005
Chou (Cabbage)	19.16	212	Homeplants	231217



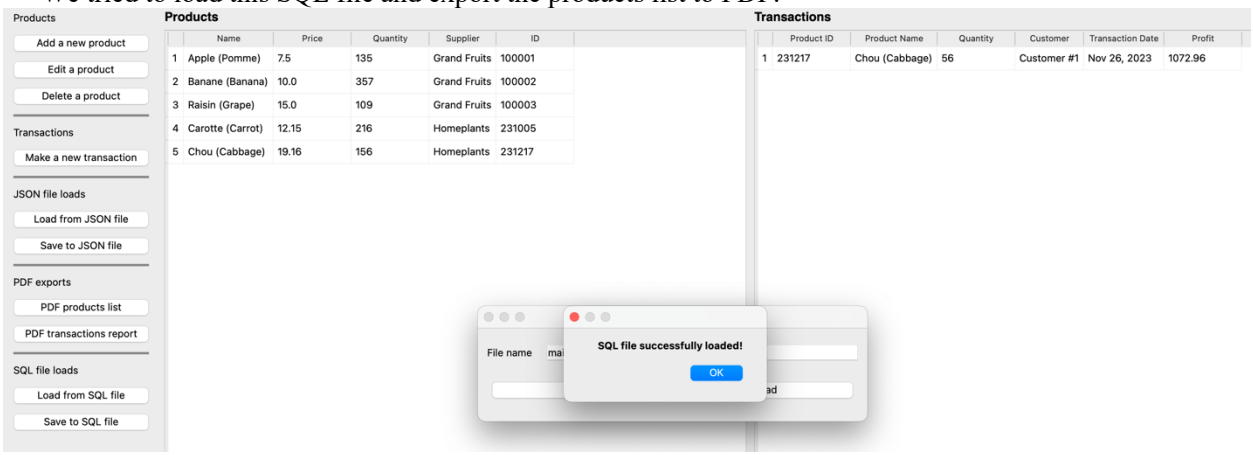
We then removed the product with ID 100004 and added a new transaction:



We then saved the data to `main_test_sql.sql`:



We tried to load this SQL file and export the products list to PDF:



Products List

Name	Price	Quantity	Supplier	ID
Apple (Pomme)	7.5	135	Grand Fruits	100001
Banane (Banana)	10.0	357	Grand Fruits	100002
Raisin (Grape)	15.0	109	Grand Fruits	100003
Carotte (Carrot)	12.15	216	Homeplants	231005
Chou (Cabbage)	19.16	156	Homeplants	231217

Then we finished the test. The program performed well with no errors.

Conclusion

Our project achieved all the set objectives. The project implemented the functions of collecting and updating information on products and transactions by storing the data in files (JSON or SQL). It also accomplished GUI to visually display the details of products and transactions information in tables. Users are also able to export their data to PDF reports.

We have learned practical Python programming knowledge from this project. For instance, we are now familiar with various concepts including class inheritance and lambda expressions. In addition, we learned how to apply the GUI library(PyQt) and database (SQLite) in our project. In this project, we improved our ability of computational thinking by designing the general framework and implementing basic functions first, and then implementing additional functions.

Our project can be improved further and might be converted into a real product.

Here are some suggestions for improvements or additional features:

1. Mobile adaptation: Consider developing a mobile version so that users can manage inventory anytime, anywhere on their phone or tablet.

2. Report customization: Allows users to customize the fields and format of the report to better meet their personalized needs.

3. Supplier management: Add a supplier management module so that users can manage supplier information, including contact information, supply products, etc.

4. Inventory management warning: set the inventory warning line, when the inventory is lower than a certain quantity, and automatically send a reminder notice, to replenish the inventory in time.

5. Sales forecast: By analyzing historical sales data, provide a sales forecast function to help users predict future sales trends.

These improvements and additional features can improve the ease of use, efficiency, and security of the system to better meet the needs of users.

For this inventory management system project, if it is possible to convert it into a real product, here are some future ideas:

1. Cloud deployment: Consider deploying the system in the cloud so that users can access and use the system anytime and anywhere over the Internet, without having to install any client software.

2. More custom features: Add more custom features to meet users' personalized needs. For example, allow users to customize the fields and format of the inventory report or the appearance and layout of the system.

3. Integrate other systems: Consider integration with other systems, such as financial systems, order management systems, etc., to improve the accuracy and operability of the data.

4. Data analysis and optimization: Through data analysis, understand users' usage habits and needs, and further optimize the function and performance of the system. For example, by analyzing sales data, understanding the situation of hot products, to timely adjusting the inventory strategy.

These ideas can help transform the inventory management system project into a real product and improve product competitiveness and market share.

Reference

1. Riverbank Computing. (2023, October 30). PyQt. In Wikipedia. Retrieved November 27, 2023, from https://en.wikipedia.org/wiki/PyQt

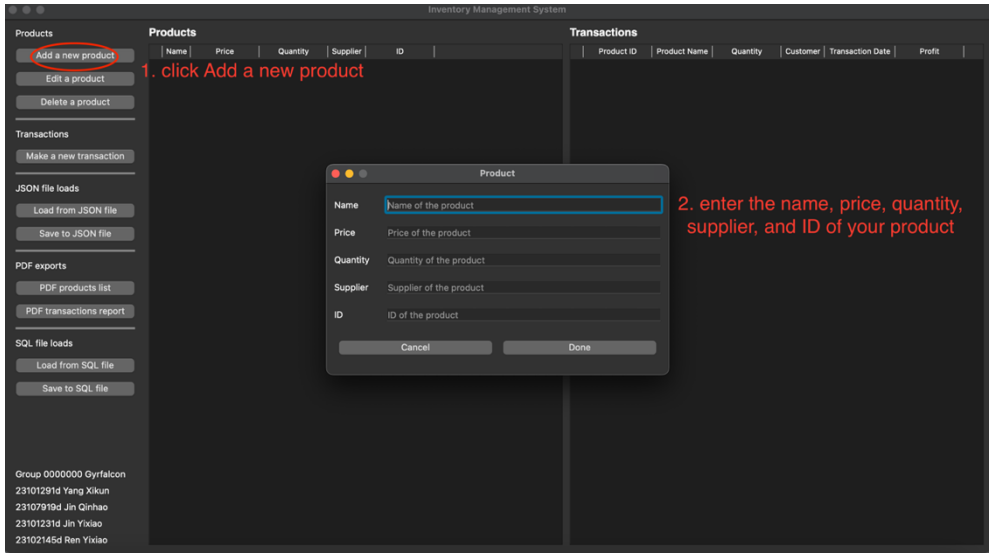
2. Wikipedia contributors. (2023, November 27). Graphical user interface. In Wikipedia. Retrieved November 27, 2023, from https://en.wikipedia.org/wiki/Graphical_user_interface

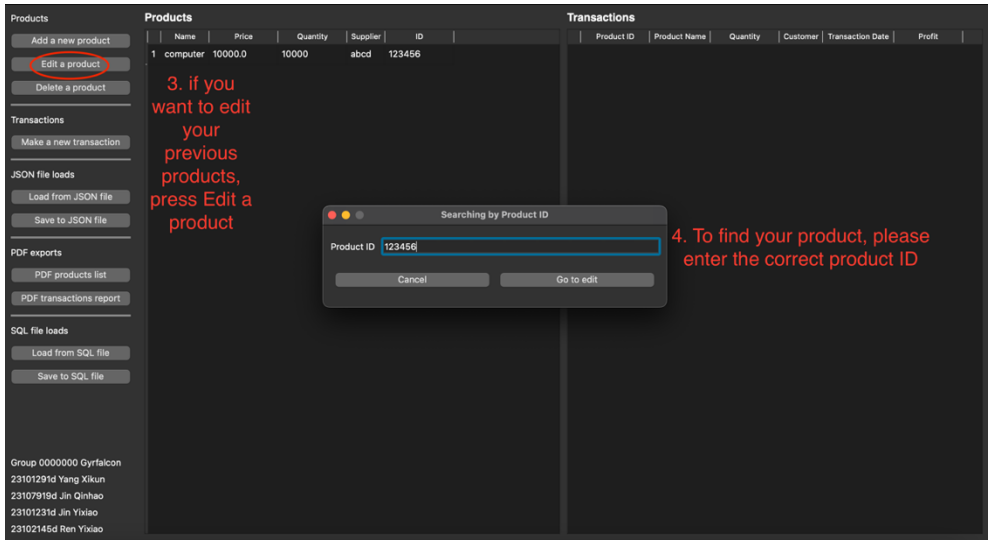
3. Wikipedia contributors. (2021, September 3). Graphical user interface. In Wikipedia. Retrieved September 3, 2021, from https://en.wikipedia.org/wiki/Graphical_user_interface

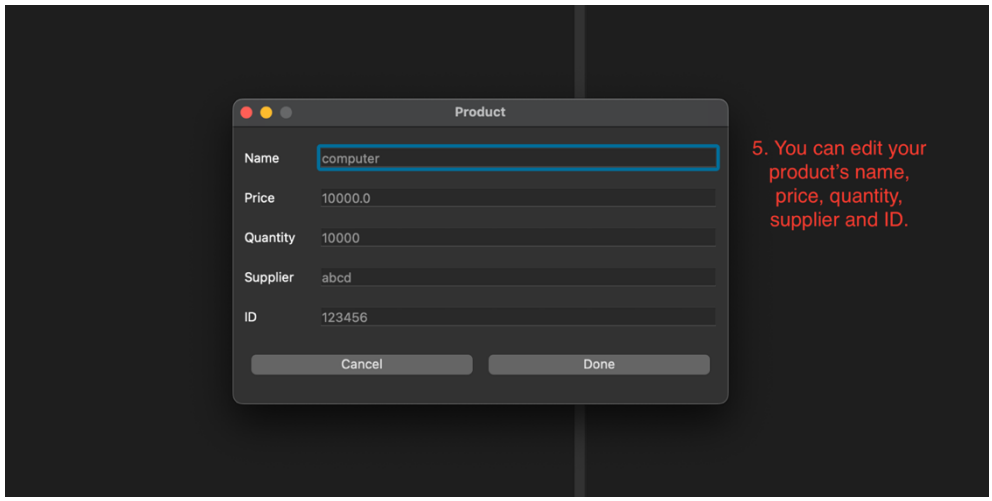
4. Wikipedia contributors. (2021, September 29). SQL. In Wikipedia. Retrieved November 27, 2023, from https://en.wikipedia.org/wiki/SQL.

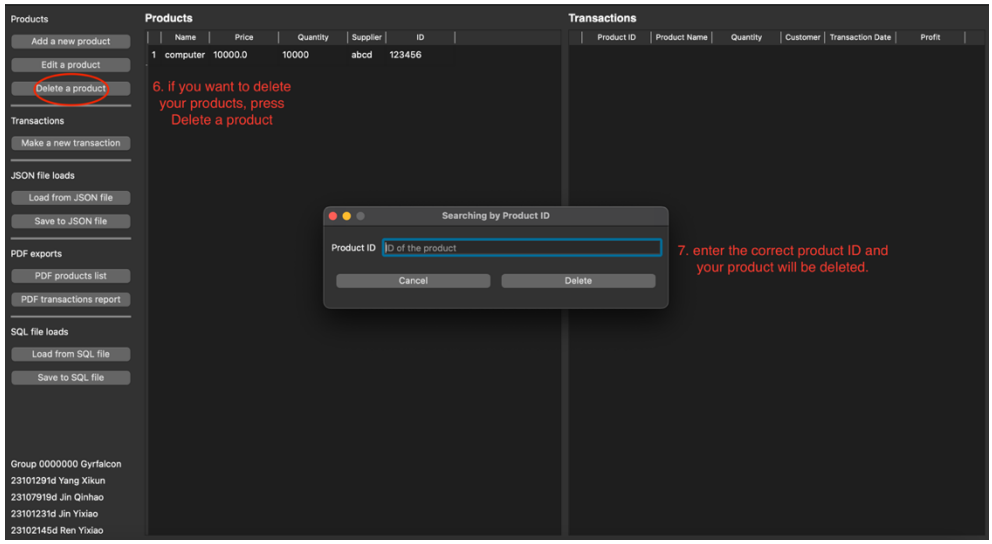
Appendixes

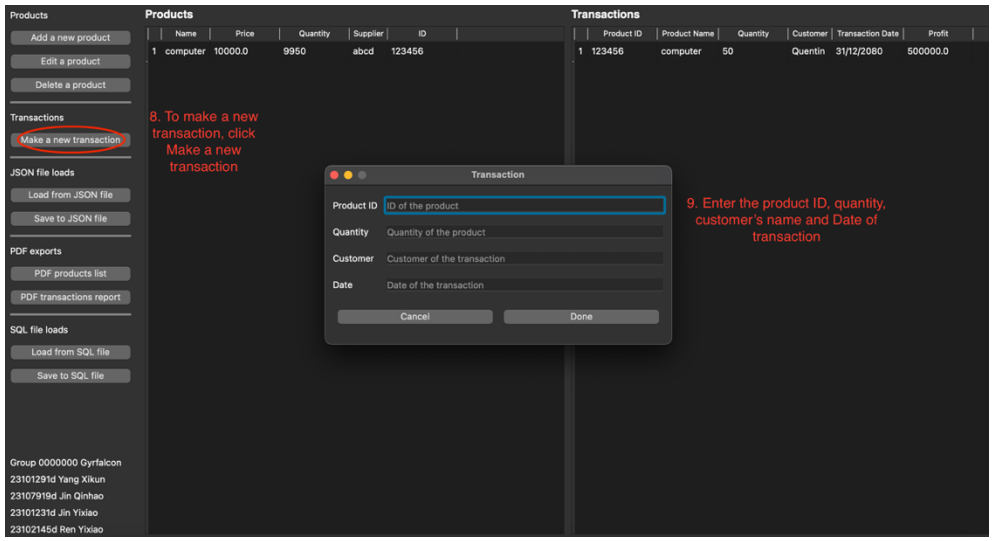
A. User Manual

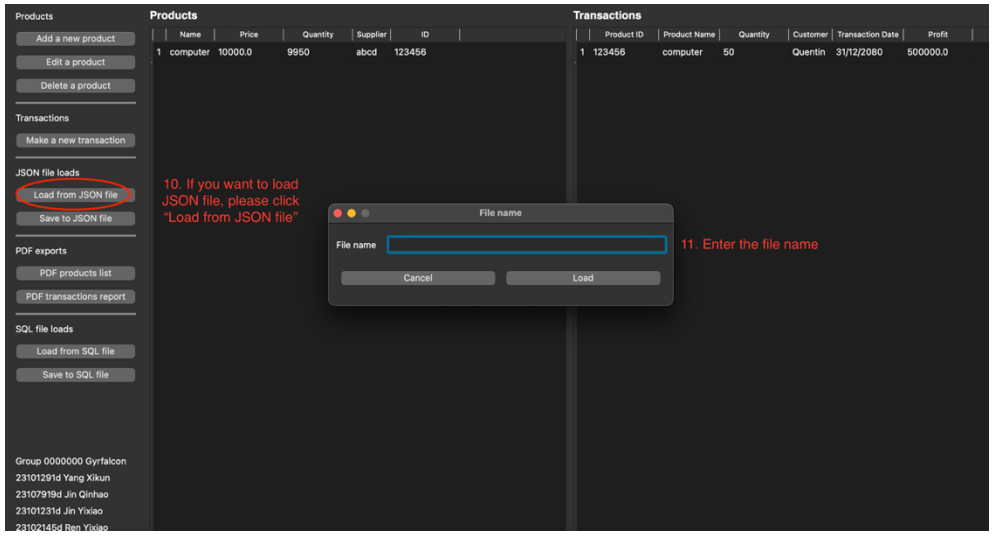


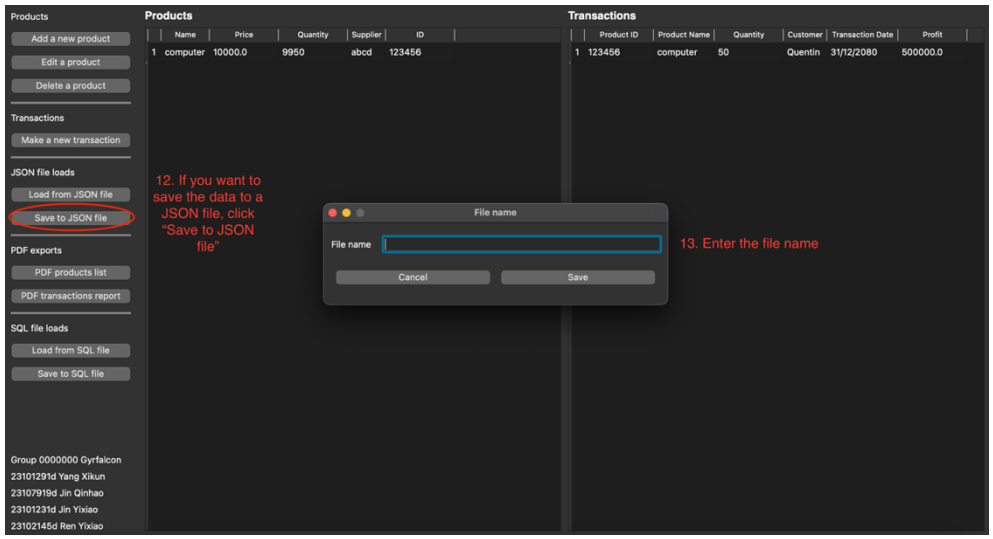


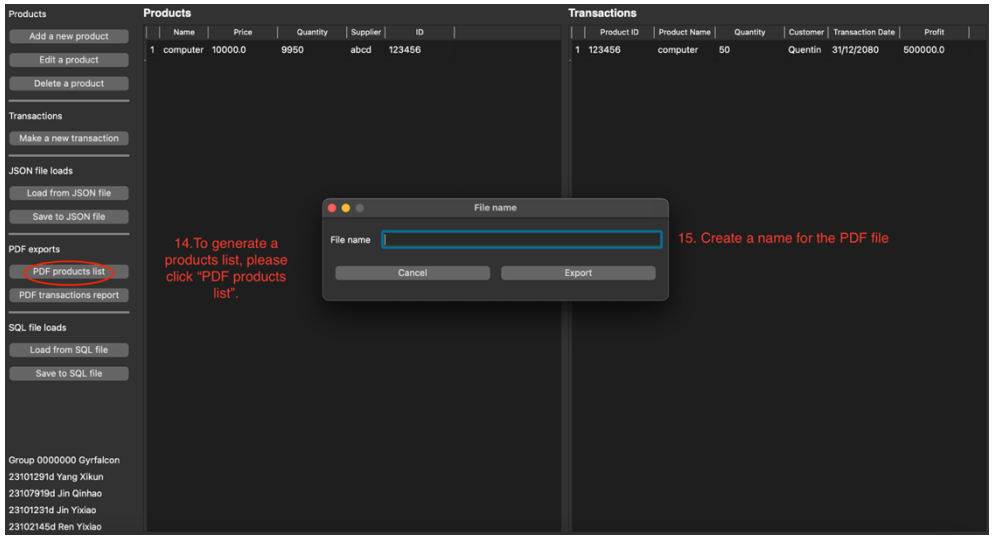


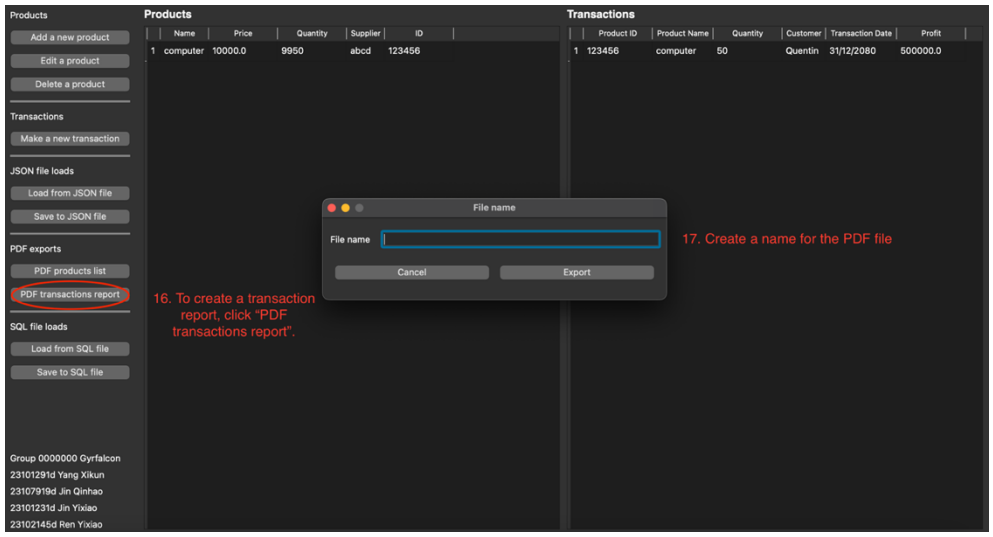

Products
Add a new product
Edit a product
Delete a product
Transactions
Make a new transaction
JSON file loads
Load from JSON file
Save to JSON file
PDF exports
PDF products list
PDF transactions report
SQL file loads
Load from SQL file
Save to SQL file
Group 0000000 Gyrfalcon
23101291d Yang Xikun
23107919d Jin Qinhao
23101231d Jin Yixiao
23102145d Ren Yixiao
Products
Name Price Quantity Supplier ID
1 computer 10000.0 10000 abcd 123456
Transactions
Product ID Product Name Quantity Customer Transaction Date Profit
6. if you want to delete your products, press Delete a product
Searching by Product ID
Product ID ID of the product
Cancel Delete
7. enter the correct product ID and your product will be deleted.

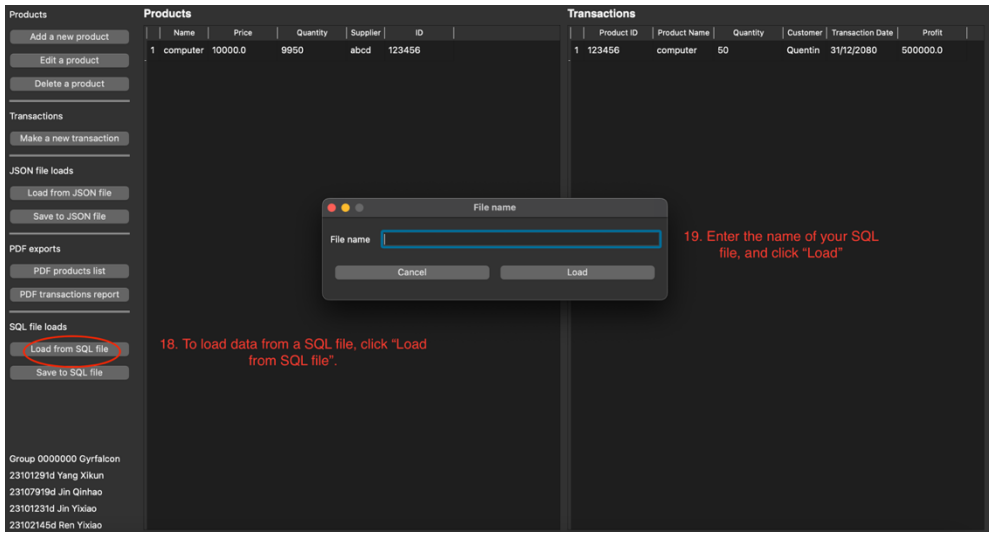

Products
Add a new product
Edit a product
Delete a product
Transactions
Make a new transaction
JSON file loads
Load from JSON file
Save to JSON file
PDF exports
PDF products list
PDF transactions report
SQL file loads
Load from SQL file
Save to SQL file
Group 0000000 Gyrfalcon
23101291d Yang Xikun
23107919d Jin Qinhao
23101231d Jin Yixiao
23102145d Ren Yixiao
Products
Name Price Quantity Supplier ID
1 computer 10000.0 9950 abcd 123456
Transactions
Product ID Product Name Quantity Customer Transaction Date Profit
1 123456 computer 50 Quentin 31/12/2080 500000.0
8. To make a new transaction, click Make a new transaction
Transaction
Product ID ID of the product
Quantity Quantity of the product
Customer Customer of the transaction
Date Date of the transaction
Cancel Done
9. Enter the product ID, quantity, customer's name and Date of transaction

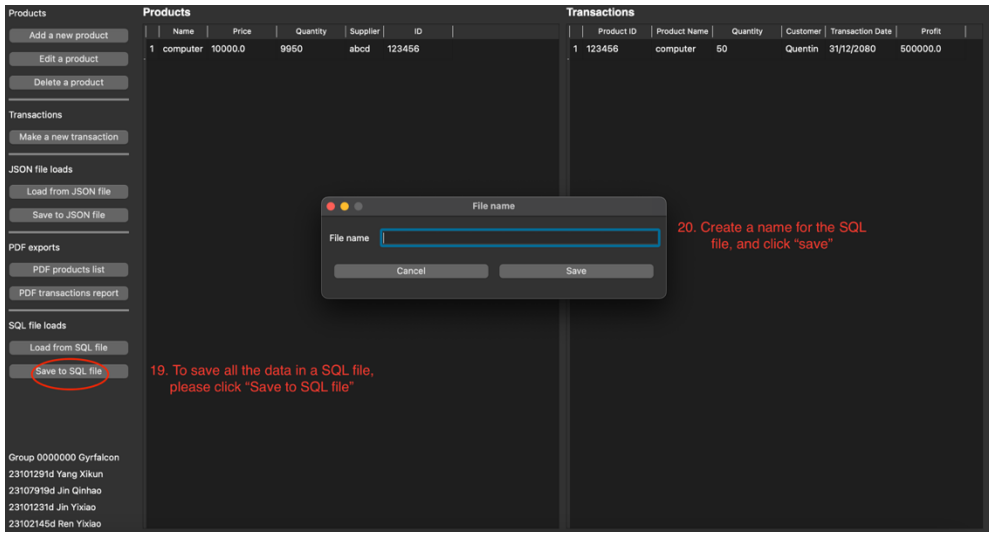

Products
Add a new product
Edit a product
Delete a product
Transactions
Make a new transaction
JSON file loads
Load from JSON file
Save to JSON file
PDF exports
PDF products list
PDF transactions report
SQL file loads
Load from SQL file
Save to SQL file
Group 0000000 Gyrfalcon
23101291d Yang Xikun
23107919d Jin Qinhao
23101231d Jin Yixiao
23102145d Ren Yixiao
Products
Name Price Quantity Supplier ID
1 computer 10000.0 9950 abcd 123456
Transactions
Product ID Product Name Quantity Customer Transaction Date Profit
1 123456 computer 50 Quentin 31/12/2080 500000.0
10. If you want to load JSON file, please click "Load from JSON file"
File name
File name
Cancel Load
11. Enter the file name


Products
Add a new product
Edit a product
Delete a product
Transactions
Make a new transaction
JSON file loads
Load from JSON file
Save to JSON file
PDF exports
PDF products list
PDF transactions report
SQL file loads
Load from SQL file
Save to SQL file
Group 0000000 Gyrfalcon
23101291d Yang Xikun
23107919d Jin Qinhao
23101231d Jin Yixiao
23102145d Ren Yixiao
Products
Name Price Quantity Supplier ID
1 computer 10000.0 9950 abcd 123456
Transactions
Product ID Product Name Quantity Customer Transaction Date Profit
1 123456 computer 50 Quentin 31/12/2080 500000.0
12. If you want to save the data to a JSON file, click "Save to JSON file"
File name
File name
Cancel
Save
13. Enter the file name


Products
Add a new product
Edit a product
Delete a product
Transactions
Make a new transaction
JSON file loads
Load from JSON file
Save to JSON file
PDF exports
PDF products list
PDF transactions report
SQL file loads
Load from SQL file
Save to SQL file
Group 0000000 Gyrfalcon
23101291d Yang Xikun
23107919d Jin Qinhao
23101231d Jin Yixiao
23102145d Ren Yixiao
Products
Name Price Quantity Supplier ID
1 computer 10000.0 9950 abcd 123456
Transactions
Product ID Product Name Quantity Customer Transaction Date Profit
1 123456 computer 50 Quentin 31/12/2080 500000.0
14.To generate a products list, please click "PDF products list".
File name
File name
Cancel
Export
15. Create a name for the PDF file


Products
Add a new product
Edit a product
Delete a product
Transactions
Make a new transaction
JSON file loads
Load from JSON file
Save to JSON file
PDF exports
PDF products list
PDF transactions report
SQL file loads
Load from SQL file
Save to SQL file
Group 0000000 Gyrfalcon
23101291d Yang Xikun
23107919d Jin Qinhao
23101231d Jin Yixiao
23102145d Ren Yixiao
Products
Name Price Quantity Supplier ID
1 computer 10000.0 9950 abcd 123456
Transactions
Product ID Product Name Quantity Customer Transaction Date Profit
1 123456 computer 50 Quentin 31/12/2080 500000.0
16. To create a transaction report, click "PDF transactions report".
File name
File name
Cancel
Export
17. Create a name for the PDF file


Products
Add a new product
Edit a product
Delete a product
Transactions
Make a new transaction
JSON file loads
Load from JSON file
Save to JSON file
PDF exports
PDF products list
PDF transactions report
SQL file loads
Load from SQL file
Save to SQL file
Group 0000000 Gyrfalcon
23101291d Yang Xikun
23107919d Jin Qinhao
23101231d Jin Yixiao
23102145d Ren Yixiao
Products
Name Price Quantity Supplier ID
1 computer 10000.0 9950 abcd 123456
Transactions
Product ID Product Name Quantity Customer Transaction Date Profit
1 123456 computer 50 Quentin 31/12/2080 500000.0
File name
File name
Cancel
Load
18. To load data from a SQL file, click "Load from SQL file".
19. Enter the name of your SQL file, and click "Load"


Products
Add a new product
Edit a product
Delete a product
Transactions
Make a new transaction
JSON file loads
Load from JSON file
Save to JSON file
PDF exports
PDF products list
PDF transactions report
SQL file loads
Load from SQL file
Save to SQL file
Group 0000000 Gyrfalcon
23101291d Yang Xikun
23107919d Jin Qinhao
23101231d Jin Yixiao
23102145d Ren Yixiao
Products
Name Price Quantity Supplier ID
1 computer 10000.0 9950 abcd 123456
Transactions
Product ID Product Name Quantity Customer Transaction Date Profit
1 123456 computer 50 Quentin 31/12/2080 500000.0
File name
File name
Cancel
Save
19. To save all the data in a SQL file, please click "Save to SQL file"
20. Create a name for the SQL file, and click "save"

B. Code Listings

https://github.com/FrankYang0610/InventoryManagementSystem

C. Testing Evidences

The testing evidence is attached under:

https://github.com/FrankYang0610/InventoryManagementSystem/tree/main/TestingEvidences