SHSG Inventory Manager

Fiona Berger, Adriano Cambria, Lionel Crepin

Introduction

The goal of this project is to develop both an accessible and user-friendly platform that displays the SHSG's current inventory. Furthermore, an emphasis on user customization has been set. Some possible functions include sorting the inventory in a variety of methods and modifying multiple attributes of each object in the inventory.

The most important aspect of an inventory platform besides the correctness is the readability of the data. This goes both for the system administrator as well as external persons who are only getting a quick overview of the inventory.

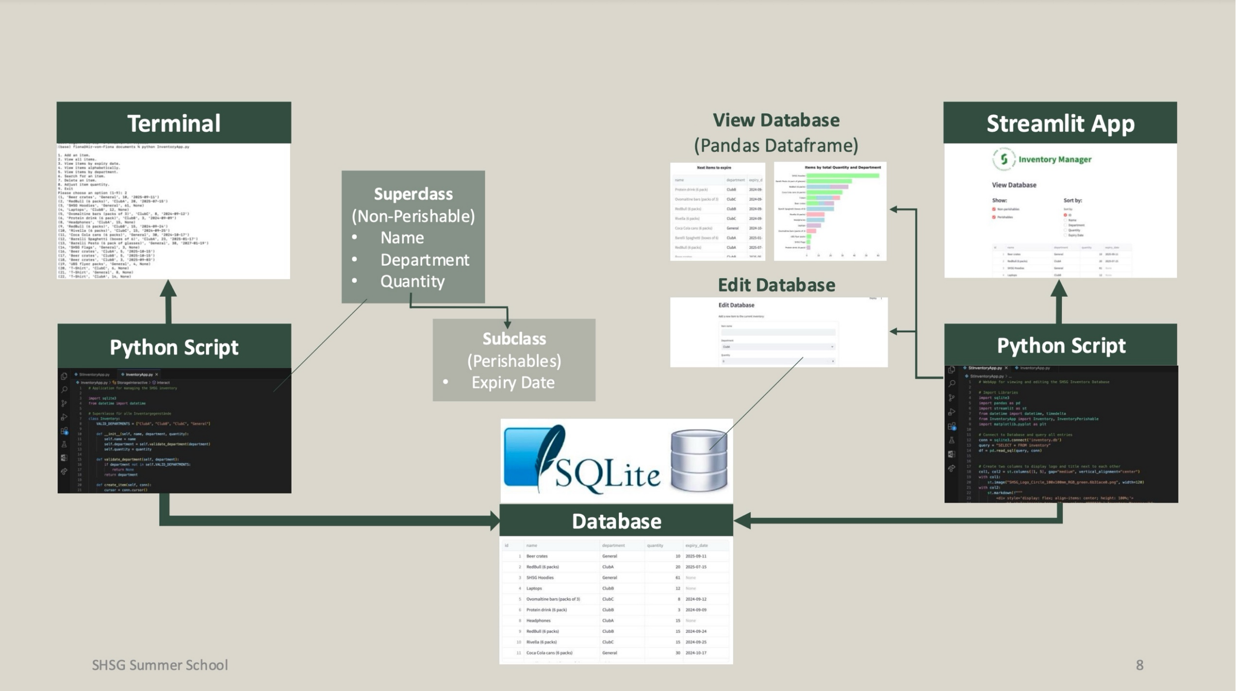
Thus, the main goal of the inventory platform is to display the correct data, provide a readable inventory, and have the option for high customization that can be adjusted to the user's preferences.

Approach

First, we thought about the essential functions the code should have. Most of these directly correspond to the formal requirements mentioned in the document provided by the professors. From there on we discussed how we wanted to incorporate each of the requirements. Specifically, the filter and search functions elicited an interesting discussion. At the core of the discussion lay the idea of having multiple options to filter the inventory. The discussion itself was held in an atmosphere dwelled in curiosity, each of us provided further ideas on which options shall be implemented. This process of discussing ideas and improving already existing ideas was found in every aspect of the platform. These discussions were sometimes quite intensive especially when it came to what function do we want to add but also calming when it came on how to visualize the data in the best possible way.

Our approach to the code is distinguished by our willingness to make compromises and our collaborative nature, as well as our objective of developing a user-friendly inventory platform.

Technical Description



Terminal

```
(base) fiona@Air-von-Fiona documents % python InventoryApp.py

1. Add an item.
2. View all items.
3. View items by expiry date.
4. View items alphabetically.
5. View items by department.
6. Search for an item.
7. Delete an item.
8. Adjust item quantity.
9. Exit
Please choose an option (1-9): 2
(1, 'Beer crates', 'General', 10, '2025-09-11')
(2, 'RedBull (6 packs)', 'ClubA', 20, '2025-07-15')
(3, 'SHSG Hoodies', 'General', 61, None)
(4, 'Laptops', 'ClubB', 12, None)
(5, 'Ovomaltine bars (packs of 3)', 'ClubC', 8, '2024-09-12')
(6, 'Protein drink (6 pack)', 'ClubB', 3, '2024-09-09')
(8, 'Headphones', 'ClubA', 15, None)
(9, 'RedBull (6 packs)', 'ClubB', 15, '2024-09-24')
(10, 'Rivella (6 packs)', 'ClubC', 15, '2024-09-25')
(11, 'Coca Cola cans (6 packs)', 'General', 30, '2024-10-17')
(12, 'Barelli Spaghetti (boxes of 6)', 'ClubA', 23, '2025-01-17')
(13, 'Barelli Pesto (6 pack of glasses)', 'General', 38, '2027-01-19')
(14, 'SHSG Flags', 'General', 3, None)
(16, 'Beer crates', 'ClubA', 5, '2025-10-15')
(17, 'Beer crates', 'ClubB', 5, '2025-10-15')
(18, 'Beer crates', 'ClubB', 3, '2025-09-03')
(19, 'UBS flyer packs', 'General', 4, None)
(20, 'T-Shirt', 'ClubC', 6, None)
(21, 'T-Shirt', 'General', 8, None)
(22, 'T-Shirt', 'ClubA', 14, None)
```

Python Script

```python
# Application for managing the SHSG inventory

import sqlite3
from datetime import datetime

# Superklasse für alle Inventargegenstände
class Inventory:
    VALID_DEPARTMENTS = ["ClubA", "ClubB", "ClubC", "General"]

    def __init__(self, name, department, quantity):
        self.name = name
        self.department = self.validate_department(department)
        self.quantity = quantity

    def validate_department(self, department):
        if department not in self.VALID_DEPARTMENTS:
            return None
        return department

    def create_item(self, conn):
        cursor = conn.cursor()
```

Superclass
(Non-Perishable)
- Name
- Department
- Quantity

Subclass
(Perishables)
- Expiry Date

Database

id	name	department	quantity	expiry_date
1	Beer crates	General	10	2025-09-11
2	RedBull (6 packs)	ClubA	20	2025-07-15
3	SHSG Hoodies	General	61	None
4	Laptops	ClubB	12	None
5	Ovomaltine bars (packs of 3)	ClubC	8	2024-09-12
6	Protein drink (6 pack)	ClubB	3	2024-09-09
8	Headphones	ClubA	15	None
9	RedBull (6 packs)	ClubB	15	2024-09-24
10	Rivella (6 packs)	ClubC	15	2024-09-25
11	Coca Cola cans (6 packs)	General	30	2024-10-17

View Database
(Pandas Dataframe)

Next items to expire

name	department	expiry_d
Protein drink (6 pack)	ClubB	2024-09-
Ovomaltine bars (packs of 3)	ClubC	2024-09-
RedBull (6 packs)	ClubB	2024-09-
Rivella (6 packs)	ClubC	2024-09-
Coca Cola cans (6 packs)	General	2024-10-
Barelli Spaghetti (boxes of 6)	ClubA	2025-01-
RedBull (6 packs)	ClubA	2025-07-

Edit Database

Edit Database

Add a new item to the current inventory:

Item name

Department: ClubA

Quantity: 0

Streamlit App

View Database

Show:
☑ Non-perishables
☑ Perishables

Sort by:
● ID
○ Name
○ Department
○ Quantity
○ Expiry Date

Python Script

```python
# WebApp for viewing and editing the SHSG Inventorx Database

# Import Libraries
import sqlite3
import pandas as pd
import streamlit as st
from datetime import datetime, timedelta
from InventoryApp import Inventory, InventoryPerishable
import matplotlib.pyplot as plt

# Connect to Database and query all entries
conn = sqlite3.connect('inventory.db')
query = "SELECT * FROM inventory"
df = pd.read_sql(query, conn)


# Create two columns to display logo and title next to each other
col1, col2 = st.columns([1, 5], gap="medium", vertical_alignment="center")
with col1:
    st.image("SHSG_Logo_Circle_100x100mm_RGB_green.6b31ace0.png", width=120)
with col2:
    st.markdown(f"""
        <div style='display: flex; align-items: center; height: 100%;'>
```

Figure 1: Graphical representation of our programme

The SHSG Inventory Management Application can help departments in managing their stock effectively. It allows adding, deleting, and modifying the product based on name, department, quantity, and expiry date if it is a perishable item.

Streamlit allowed us to build an intuitive UI. Users can see an inventory database that is capable of filtering items and sorting entries based on ID, name, department, amount, or expiration date. Finally, by exploiting the flexibility in managing large inventories, users can easily search and inspect items. Reports regarding departments will calculate the number of perishable items expiring soon. Colour-coded overviews add to clarity, and warnings for products with expiration dates within 30 days serve to prioritize stock rotation.

Interactive user interfaces provide the facility to create, read, update, and delete inventory objects. Add new items, edit the quantity, or remove obsolete inventory. A confirmation dialog box ensures that crucial actions like deletion of an item and editing of the quantity are performed only when the user confirms the action to avoid unintentional changes. Inventory data is managed with SQLite for efficient retrieval and storing.

Pie and bar charts provide important pictorial information, which enables the user to have a quick overview over the inventory. A dashboard displaying the segmentation of inventory across departments, the ratio between the perishable and non-perishable items is displayed. Departmental product shares are represented as pie charts and item counts as bar graphs. Color-coded data visualizations provide immediate information on stock levels and possible shortages, therefore assisting in making inventory decisions.

Integrating Streamlit with SQLite and Pandas provides ease in data manipulation and visualization, hence, allow easy inventory management by non-technical users. The system is designed in such a manner that it makes it very convenient to perform day-to-day operations and long-run inventory planning by flawlessly shifting between viewing, updating, and modification in the inventory data.

Conclusion

The main purpose of displaying the code in a structured manner was achieved. Furthermore, our objective of enabling the platform to be customized to the users' preferences has been completed. Nonetheless, not everything that we had planned was realized. The biggest cut we had to make was the direct editing button on the table displayed on the Streamlit website. The editing button intended to directly change the properties of the respective object in the inventory, sadly each time we tried to implement the editing button the database got corrupted, which meant that we had to scrap the idea, mainly due to the time constraints.

Lastly, if Streamlit does not function for reasons which are unknown, then the whole Streamlit side of the code cannot be used due to it being dependent on Streamlit. Therefore, an adjustment of the code which would enable the creation of a GUI which is not as heavily reliant on Streamlit or other functions would be beneficial as it would make the code more stable.

Group Reflection

The group project caught our interest, as we are all drawn to coding and software engineering. The framework and setting of the project seemed realistic and motivated us to make a functioning app that had the potential to help the SHSG. The determination to polish and refine our code into an accessible, capable, and user-friendly platform motivated us to go beyond the knowledge provided by this course. Thereby, creating multiple in-depth conversations on how to solve certain errors or further improve the existing code through new ideas or changes. Complex problems and codes could be solved with the help of ChatGPT and the coaches. Information provided during the first week on how to prompt effectively, facilitated the work with ChatGPT. Overall, the group project left an outstanding impression on our group and ignited the curiosity for further coding projects.