Report for OOP Project: Shopping System

Problem Description

The objective of this project is to create an online shopping system using Object-Oriented Programming (OOP) principles in Python. The system allows users to create accounts, log in, browse products, add items to a shopping cart, and finalize purchases. Additionally, it maintains a history of user activities and purchases.

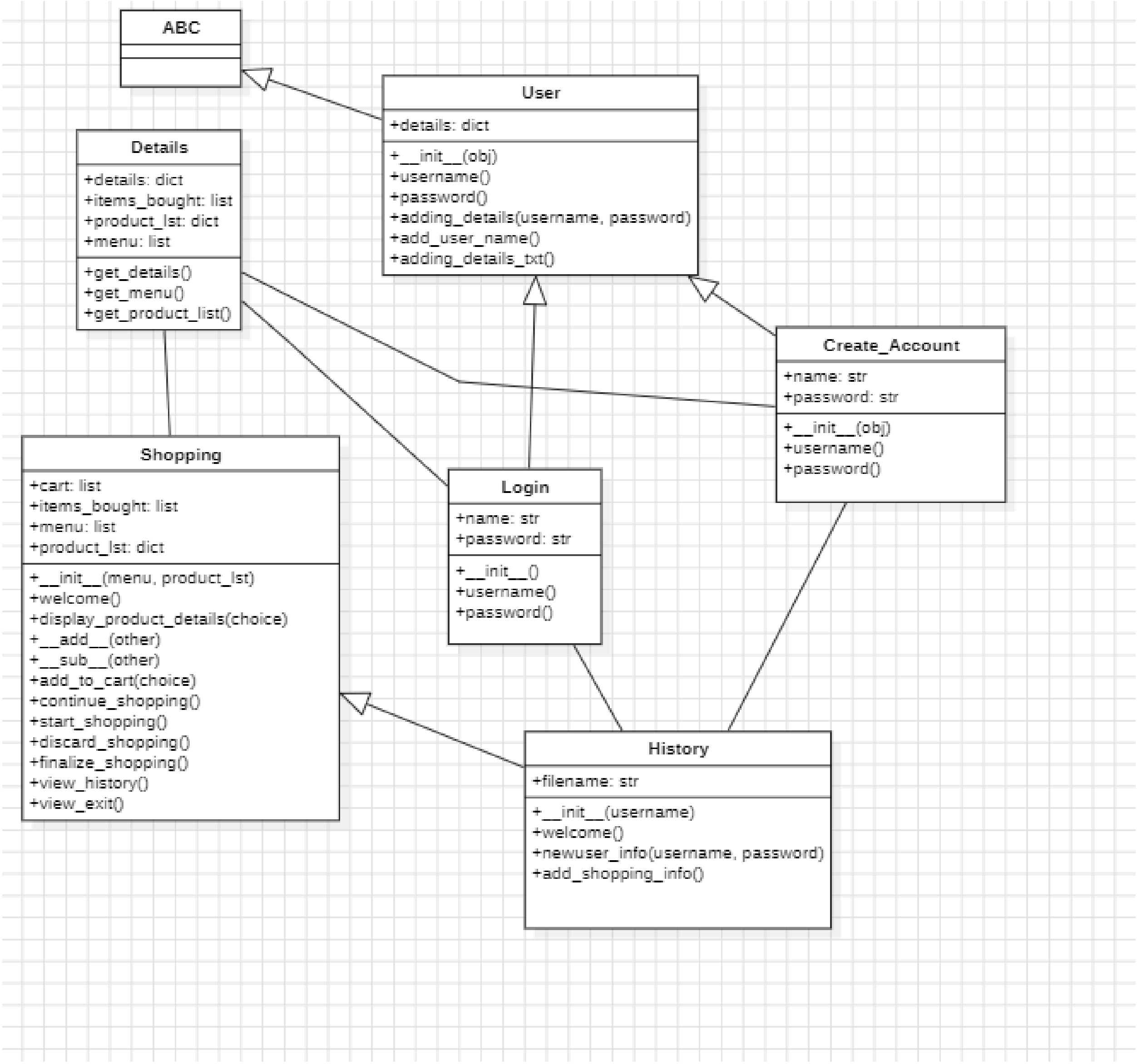
Distinguishing Features of Your Project

- **Account Management**: Users can create accounts with unique usernames and alphanumeric passwords. The system checks for existing usernames to ensure uniqueness.
- **Shopping Cart**: Users can browse a list of products, view product details, add items to their cart, and remove items if needed.
- **Purchase History**: The system records the shopping history for each user, including the date and time of purchases.
- **Operator Overloading**: The Shopping class uses operator overloading for adding and removing items from the cart.
- **File Management**: User details and history are stored in text files for persistence between sessions.

Flow of Your Project Including Class Diagram

1. **Details Class**: Manages product details and loads user details from a file.
2. **User Class (Abstract Base Class)**: Defines the structure for user-related operations.
    - **Create_Account Class**: Handles user account creation.
    - **Login Class**: Manages user login.
3. **Shopping Class**: Manages the shopping cart, product browsing, and finalizing purchases.
4. **History Class**: Handles recording and displaying the user's shopping history.

Class Diagram



Most Challenging Part While Working on the Project

The most challenging part was implementing the operator overloading in the Shopping class to manage adding and removing items from the shopping cart. Ensuring proper synchronization between the cart and the actual product list was crucial.

New Things Learned in Python While Working on the Project

- **Abstract Base Classes (ABC)**: Learned how to use ABCs to define a common interface for the User class.
- **Operator Overloading**: Implemented operator overloading to enhance the functionality of adding and removing items in the Shopping class.
- **File Handling**: Improved skills in file handling for storing and retrieving user details and shopping history.

Future Expansions

- **Product Inventory Management**: Implementing inventory tracking to manage stock levels.
- **User Interface**: Developing a graphical user interface (GUI) for better user experience.
- **Payment Integration**: Adding secure payment gateways for transaction processing.

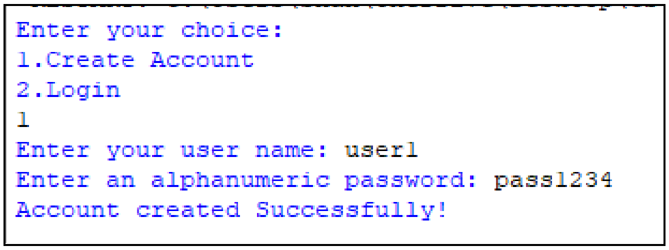
List of References

- Python Documentation: https://docs.python.org/3/
- Real Python Tutorials: https://realpython.com/
- ChatGpt

Test Case Runs

1. **Test Case 1: Creating an Account**
   - **Input**: Username: user1, Password: pass1234
   - **Expected Output**: "Account created Successfully!"

Screenshot:

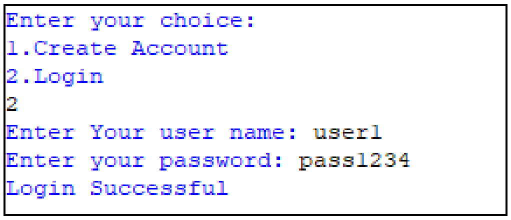


```
Enter your choice:
1.Create Account
2.Login
1
Enter your user name: user1
Enter an alphanumeric password: pass1234
Account created Successfully!
```

2. **Test Case 2: Logging In**

   - **Input**: Username: user1, Password: pass1234
   - **Expected Output**: "Login Successful"

Screenshot:

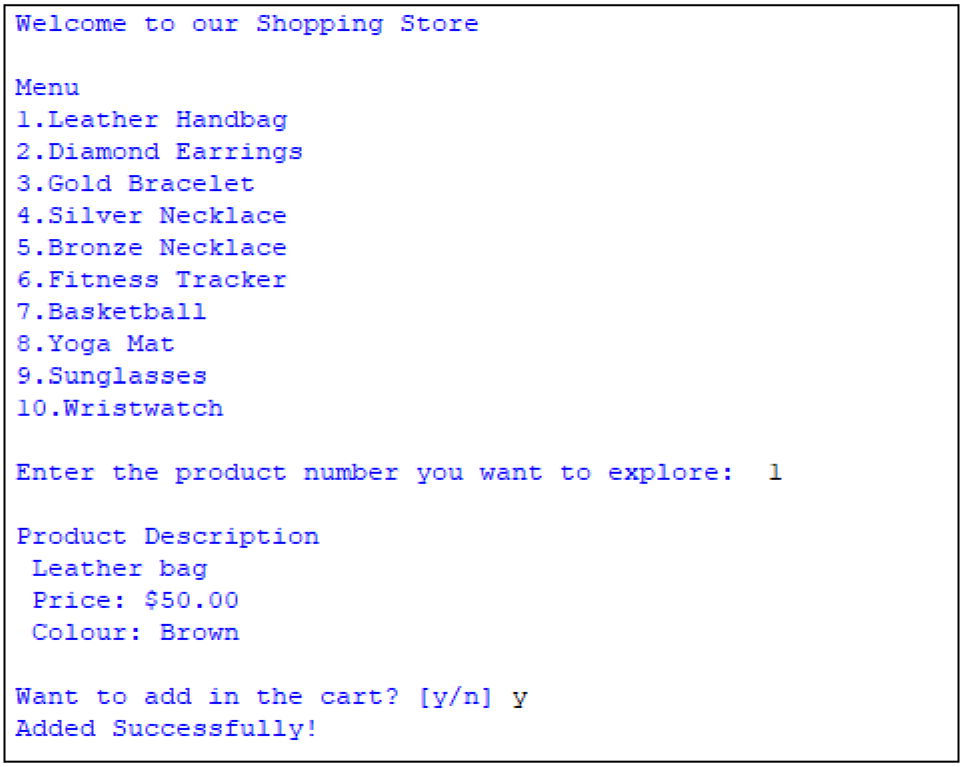


```
Enter your choice:
1.Create Account
2.Login
2
Enter Your user name: user1
Enter your password: pass1234
Login Successful
```

3. **Test Case 3: Adding Items to Cart**

- **Input**: Product Number: 1 (Leather Handbag)
- **Expected Output**: "Added Successfully!"

Screenshot:



```
Welcome to our Shopping Store

Menu
1.Leather Handbag
2.Diamond Earrings
3.Gold Bracelet
4.Silver Necklace
5.Bronze Necklace
6.Fitness Tracker
7.Basketball
8.Yoga Mat
9.Sunglasses
10.Wristwatch

Enter the product number you want to explore:  1

Product Description
 Leather bag
 Price: $50.00
 Colour: Brown

Want to add in the cart? [y/n] y
Added Successfully!
```