Design Document:
Warehouse Management System

Table of Contents

Introduction to project

The customer is a Warehouse company that needs a system for managing their operations. They would like to have a a monolithic system that has lots of different functionalities packaged in a single program. They wish for functionality for, for example:

- managing their product stock
- creating product orders
- tracking transactions
- managing payments
- storing receipts
- retrieving all sorts of business-related statistics
- paying salaries
- communicating internally through a chat channel
- etc.

Design decisions

Having such a huge monolithic system that manages all these business operations would introduce a vulnerability to their business. Eventual software failure or cybersecurity threats should be considered. A problem arising in one part of the system could put the whole system in an unreliable state, which could impede the business from operating while we try to find out where the problem is and how to fix it. (This could take a long time, as the system is huge.)

Imagine for example the hypothetical situation where a bug in the systems's chat causes the whole system to be unstable. In the worst case scenario, we might need to close the whole system for maintenance, during which the company would be unable to do anything as everything is in this one system. Or an even worse hypothetical scenario could be that an employee clicks on a suspicious link and downloads some ransomware or gives attackers access to their credentials and as a result the whole system is exposed to the attackers.

Therefore, to avoid this vulnerability, this project suggests that we delegate the various functionalities to the separate applications described in the diagram "Warehouse management system". In this way, smaller applications can focus on their specific areas of responsibility.

This project focuses on the functionality that is related to managing the warehouse stock. That is, removing sold items from the inventory, returning items to the inventory, and managing orders. Hence, the application that has been developed in this sprint is an Inventory System rather than a Warehouse Management System. See the use case diagram for an overview of the functionality provided by this app.

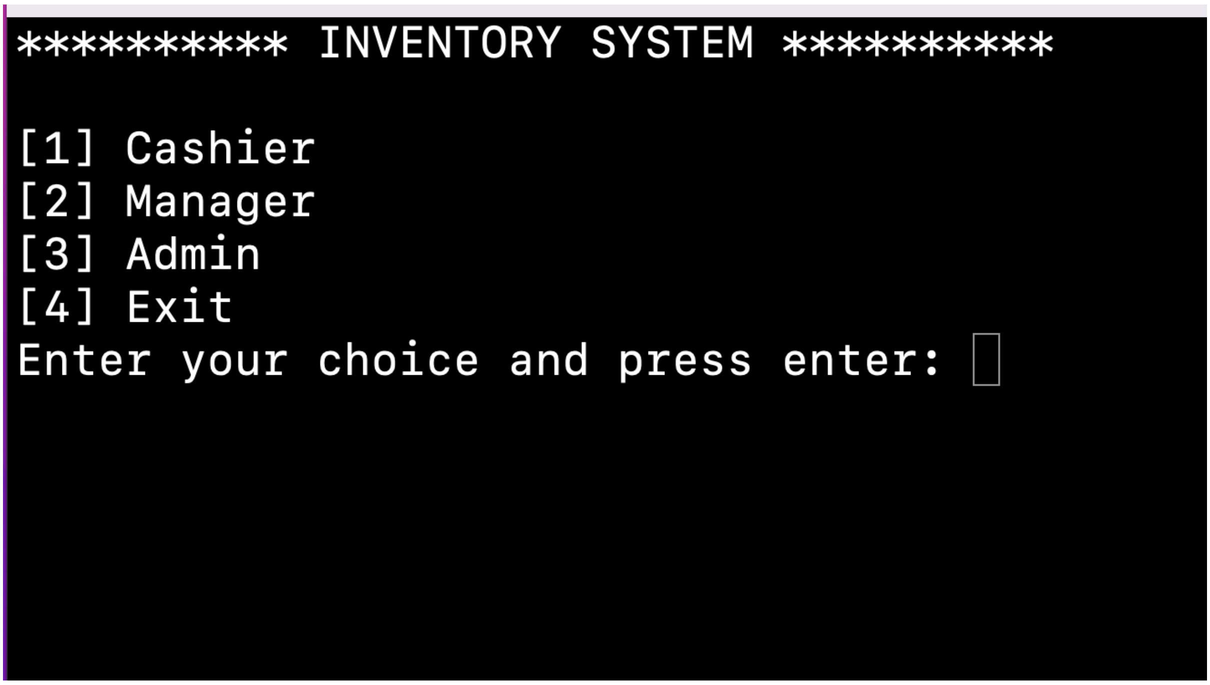
Multiple Simultaneous Users

The Inventory System will be used by many cashiers at the same time. The goal in this sprint has been to make the program update it's inventory data as soon as a user makes any changes to any type of data (order, product or transaction). The program implements the Observer - Subject design pattern, in which FileHandlers watch the pools that contain inventory data items and save the current state of the pools to file as soon as they observe any changes. And other users can then retrieve the up to date data from file.

An implementation of a FileWatcher with WatchService has also been in process in this sprint. A FileWatcher desing guarantees that all users get the latest changes from the files as soon as some changes are made to the watched files. This implementation is however not stable enough to be delivered yet. The version that is delivered in this sprint updates the inventory data pools from file whenever a user begins an operation.

Application Introduction

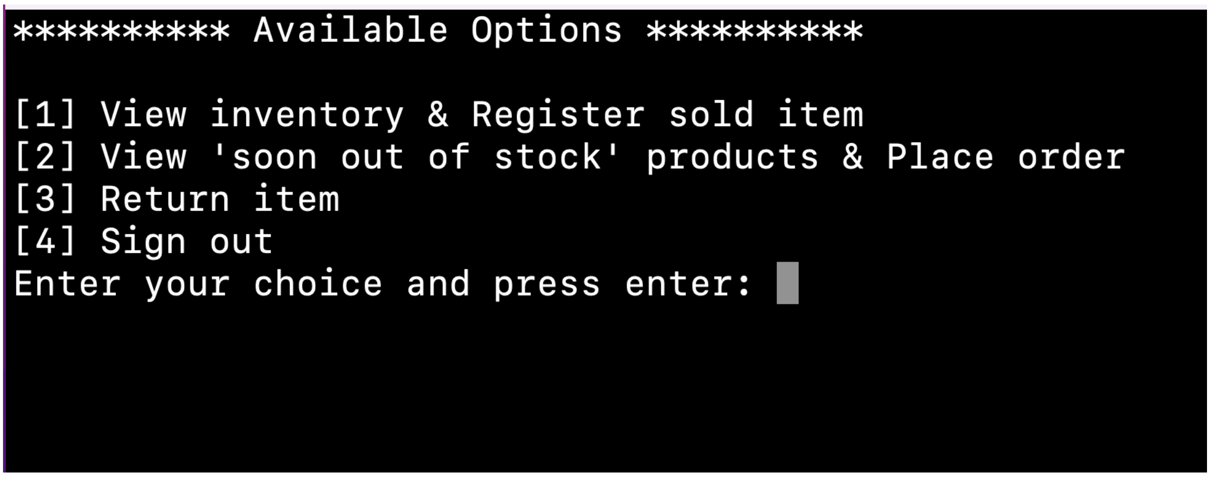
Main Menu



```
*********** INVENTORY SYSTEM ***********

[1] Cashier
[2] Manager
[3] Admin
[4] Exit
Enter your choice and press enter: 
```

In the main menu the user can choose to sign in as a cashier, manager, or admin.

Cashier Menu

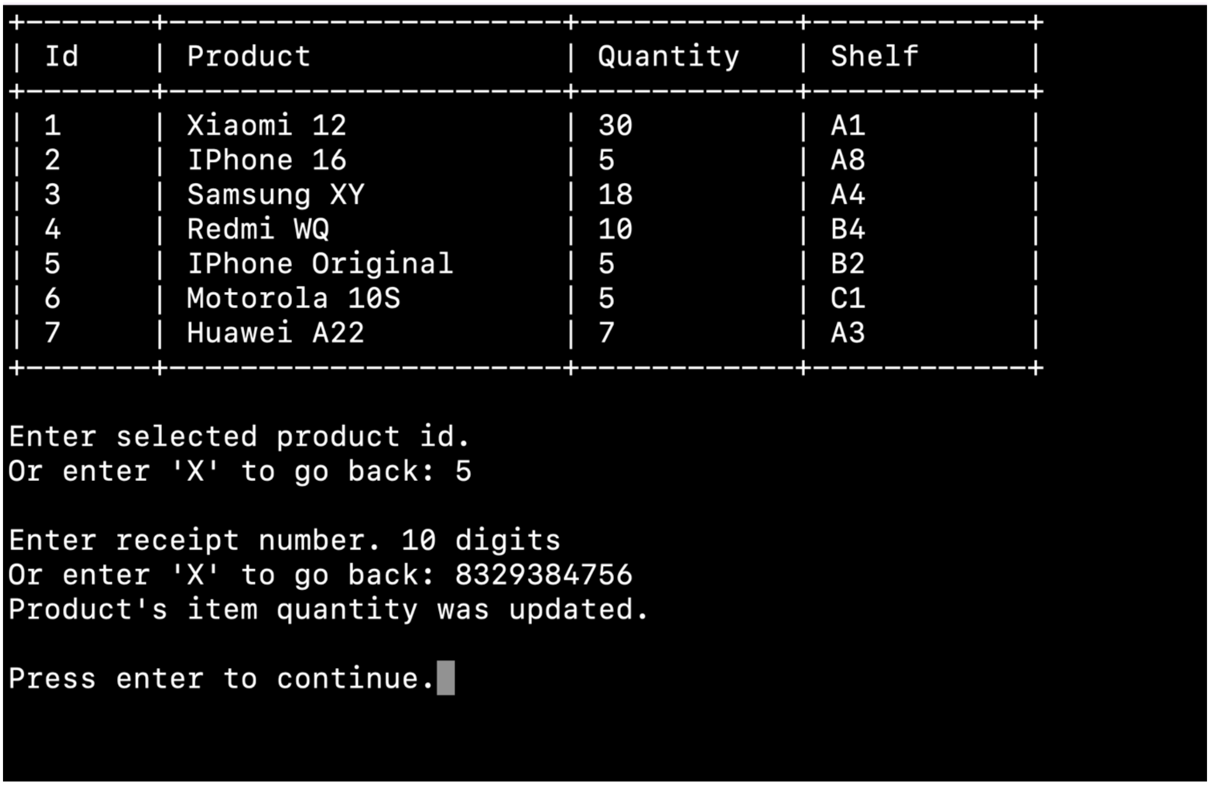


```
********** Available Options **********

[1] View inventory & Register sold item
[2] View 'soon out of stock' products & Place order
[3] Return item
[4] Sign out
Enter your choice and press enter: 
```

The cashier menu provides the options shown in the screen shot above.

View inventory & Register sold item

If option 1 is chosen, the cashier can see the list of products in the inventory, choose one by entering its id and then update its item quantity by entering a receipt number. See screenshot below.

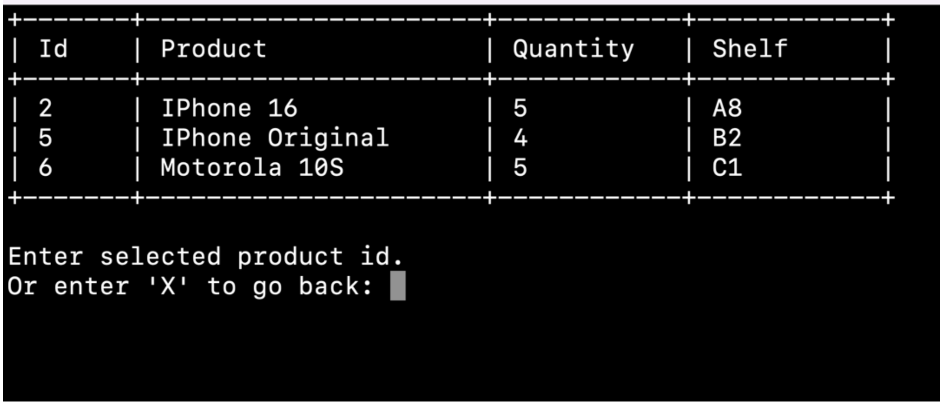


```
+-------+----------------------+------------+------------+
| Id    | Product              | Quantity   | Shelf      |
+-------+----------------------+------------+------------+
| 1     | Xiaomi 12            | 30         | A1         |
| 2     | IPhone 16            | 5          | A8         |
| 3     | Samsung XY           | 18         | A4         |
| 4     | Redmi WQ             | 10         | B4         |
| 5     | IPhone Original      | 5          | B2         |
| 6     | Motorola 10S         | 5          | C1         |
| 7     | Huawei A22           | 7          | A3         |
+-------+----------------------+------------+------------+

Enter selected product id.
Or enter 'X' to go back: 5

Enter receipt number. 10 digits
Or enter 'X' to go back: 8329384756
Product's item quantity was updated.

Press enter to continue.
```

The inventory doesn’t display product prices because according to this project design the prices will be registered in the payment system and it would be cumbersome to have to update prices in two different places.

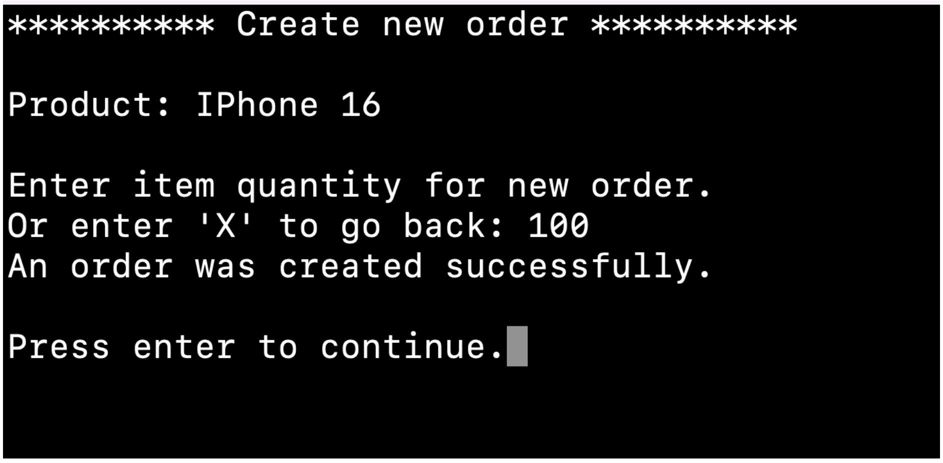
View ‘Soon out of stock’ products & Place order

If the cashier chooses option 2 the application shows the table seen below. Once again, the cashier enters a product id. See screenshot below.



```
+-------+----------------------+------------+------------+
| Id    | Product              | Quantity   | Shelf      |
+-------+----------------------+------------+------------+
| 2     | IPhone 16            | 5          | A8         |
| 5     | IPhone Original      | 4          | B2         |
| 6     | Motorola 10S         | 5          | C1         |
+-------+----------------------+------------+------------+

Enter selected product id.
Or enter 'X' to go back:
```

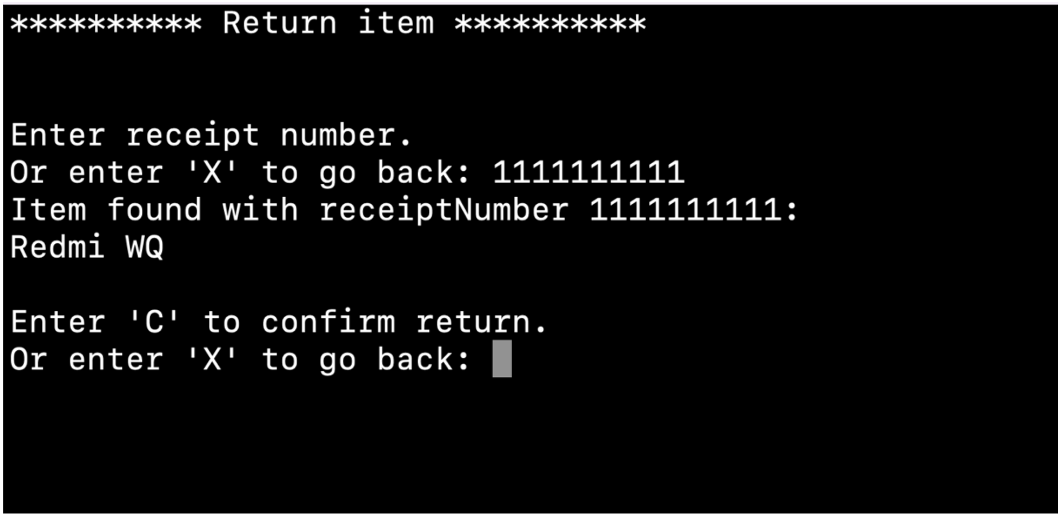
In the next screen the cashier enters the quantity to be ordered.



```
********** Create new order **********

Product: IPhone 16

Enter item quantity for new order.
Or enter 'X' to go back: 100
An order was created successfully.

Press enter to continue.
```

Return item

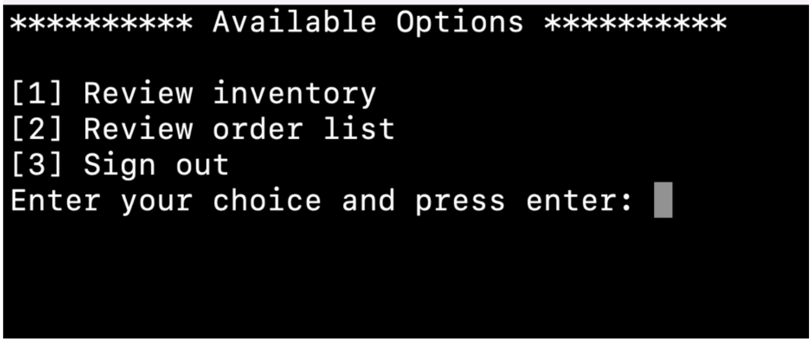
If the cashier chooses to return an item, she or he needs to enter a valid receipt number. The program checks that the receipt number is found in the registered transactions. Following screen is presented:



```
********** Return item **********


Enter receipt number.
Or enter 'X' to go back: 1111111111
Item found with receiptNumber 1111111111:
Redmi WQ

Enter 'C' to confirm return.
Or enter 'X' to go back:
```

Manager Menu

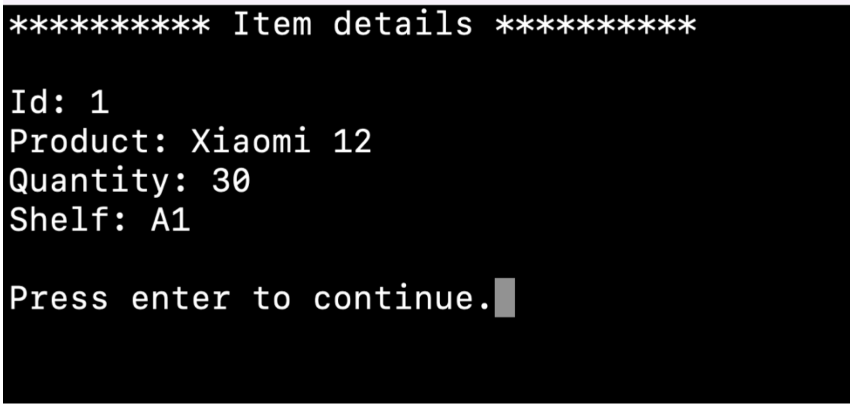
The manager has the options seen in the screenshot below.



```
********** Available Options **********

[1] Review inventory
[2] Review order list
[3] Sign out
Enter your choice and press enter:
```

Review inventory

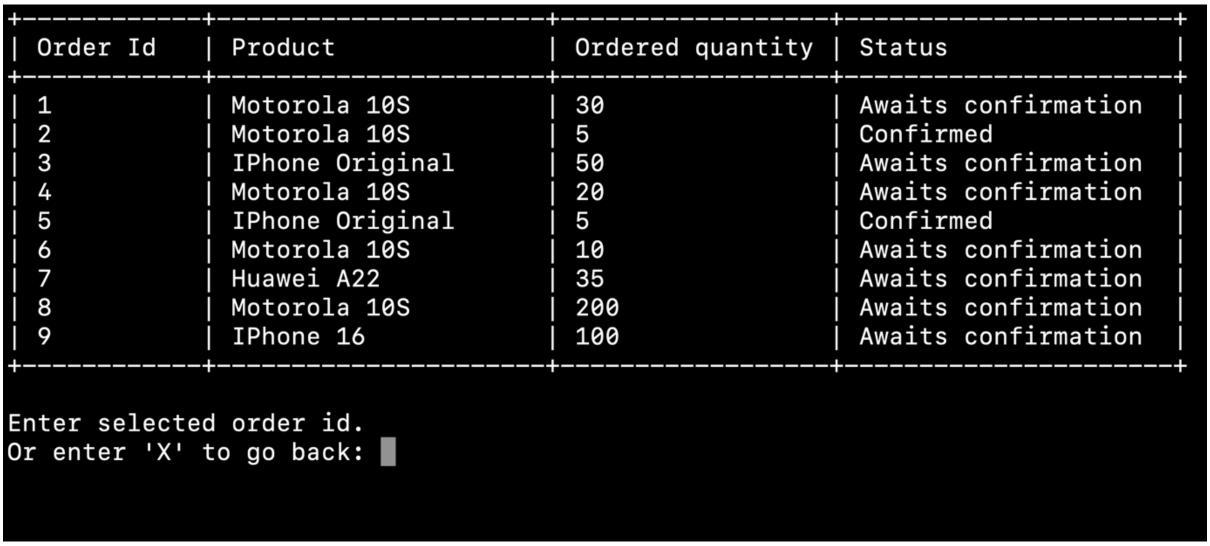
If the manager chooses to review the inventory, she or he sees the same table with products as the cashier, but the following screen is different, because the manager doesn’t register sold items. The manager can instead inspect the product details. In a next version, this screen could also display information orders or sales for the product.



```
********** Item details **********

Id: 1
Product: Xiaomi 12
Quantity: 30
Shelf: A1

Press enter to continue.
```

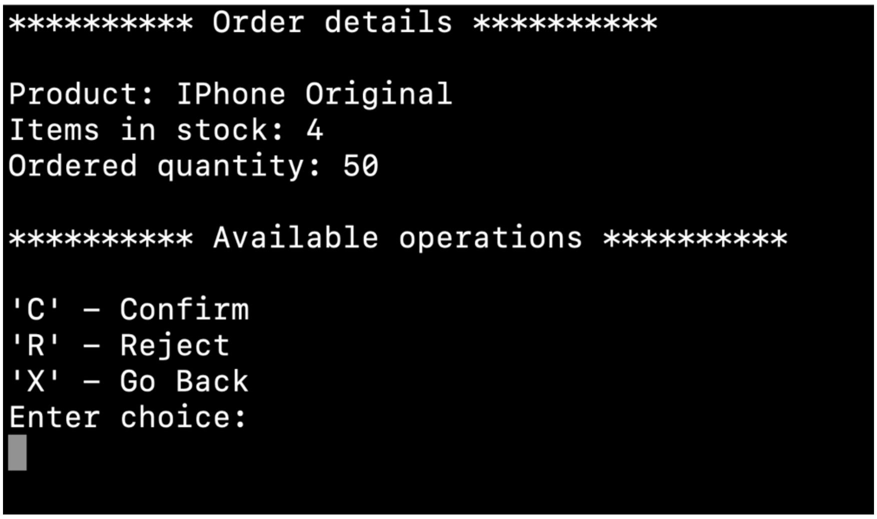
Review order list

The manager can see a list of all orders that cashiers have created.



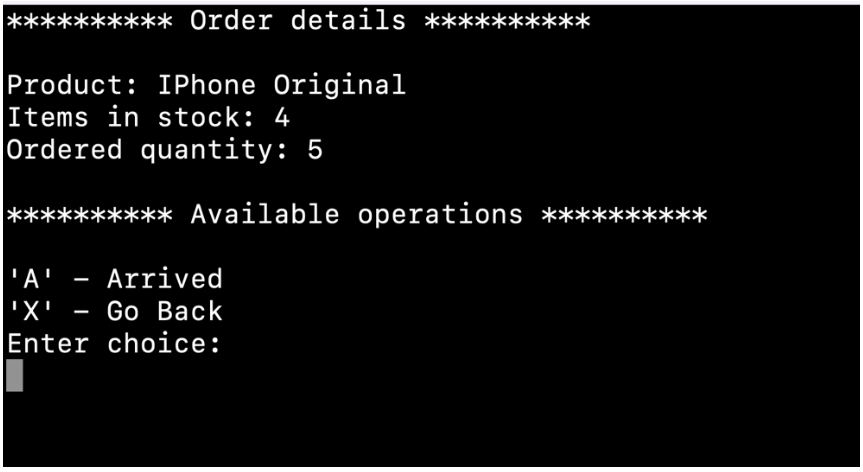
```
+------------+----------------------+------------------+----------------------+
| Order Id   | Product              | Ordered quantity | Status               |
+------------+----------------------+------------------+----------------------+
| 1          | Motorola 10S         | 30               | Awaits confirmation  |
| 2          | Motorola 10S         | 5                | Confirmed            |
| 3          | IPhone Original      | 50               | Awaits confirmation  |
| 4          | Motorola 10S         | 20               | Awaits confirmation  |
| 5          | IPhone Original      | 5                | Confirmed            |
| 6          | Motorola 10S         | 10               | Awaits confirmation  |
| 7          | Huawei A22           | 35               | Awaits confirmation  |
| 8          | Motorola 10S         | 200              | Awaits confirmation  |
| 9          | IPhone 16            | 100              | Awaits confirmation  |
+------------+----------------------+------------------+----------------------+

Enter selected order id.
Or enter 'X' to go back:
```

If the manager chooses an order that await confirmation, he or she can see the screen shown below and choose to either confirm or reject the order.



```
********** Order details **********

Product: IPhone Original
Items in stock: 4
Ordered quantity: 50

********** Available operations **********

'C' - Confirm
'R' - Reject
'X' - Go Back
Enter choice:
```

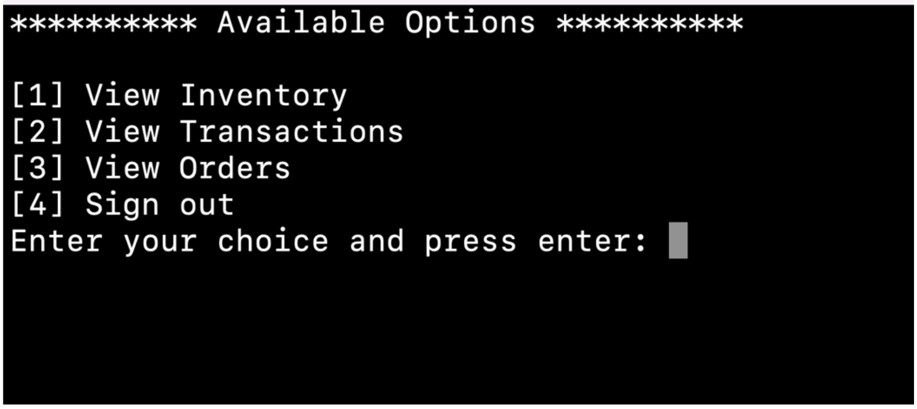
If the manager chooses an order that is already confirmed, she or he can mark it as arrived. Then the items in the order are added to the products in the inventory.



```
********** Order details **********

Product: IPhone Original
Items in stock: 4
Ordered quantity: 5

********** Available operations **********

'A' - Arrived
'X' - Go Back
Enter choice:
```

Admin Menu

The admin can see the following menu options.

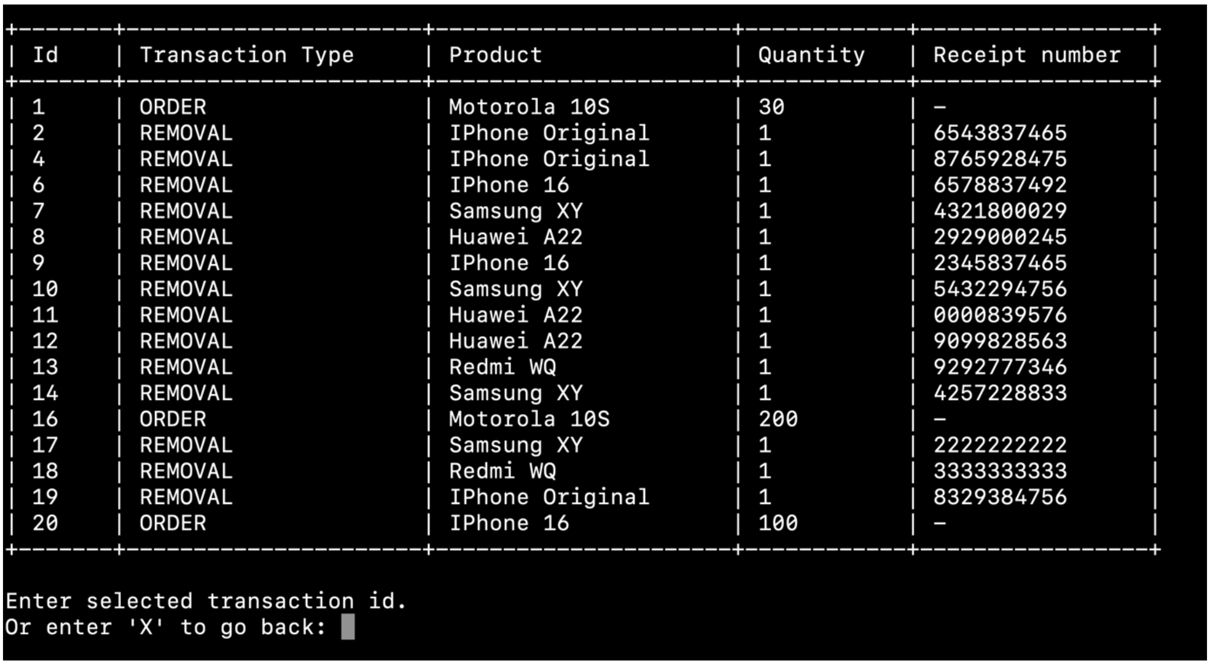


```
********** Available Options **********

[1] View Inventory
[2] View Transactions
[3] View Orders
[4] Sign out
Enter your choice and press enter:
```

View Inventory

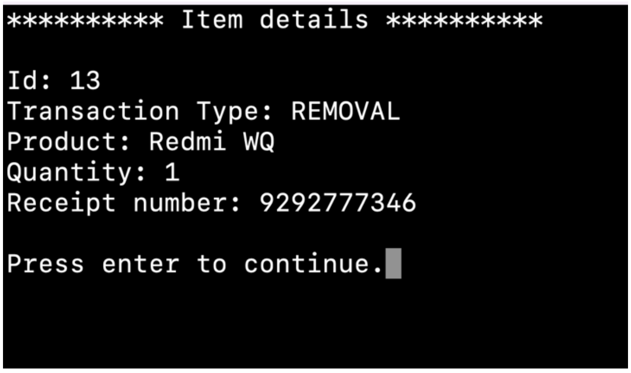
The admin can inspect the inventory in the same manner as the manager.

View Transactions



```
+--------+----------------------+----------------------+------------+-----------------+
| Id     | Transaction Type     | Product              | Quantity   | Receipt number  |
+--------+----------------------+----------------------+------------+-----------------+
| 1      | ORDER                | Motorola 10S         | 30         | -               |
| 2      | REMOVAL              | IPhone Original      | 1          | 6543837465      |
| 4      | REMOVAL              | IPhone Original      | 1          | 8765928475      |
| 6      | REMOVAL              | IPhone 16            | 1          | 6578837492      |
| 7      | REMOVAL              | Samsung XY           | 1          | 4321800029      |
| 8      | REMOVAL              | Huawei A22           | 1          | 2929000245      |
| 9      | REMOVAL              | IPhone 16            | 1          | 2345837465      |
| 10     | REMOVAL              | Samsung XY           | 1          | 5432294756      |
| 11     | REMOVAL              | Huawei A22           | 1          | 0000839576      |
| 12     | REMOVAL              | Huawei A22           | 1          | 9099828563      |
| 13     | REMOVAL              | Redmi WQ             | 1          | 9292777346      |
| 14     | REMOVAL              | Samsung XY           | 1          | 4257228833      |
| 16     | ORDER                | Motorola 10S         | 200        | -               |
| 17     | REMOVAL              | Samsung XY           | 1          | 2222222222      |
| 18     | REMOVAL              | Redmi WQ             | 1          | 3333333333      |
| 19     | REMOVAL              | IPhone Original      | 1          | 8329384756      |
| 20     | ORDER                | IPhone 16            | 100        | -               |
+--------+----------------------+----------------------+------------+-----------------+

Enter selected transaction id.
Or enter 'X' to go back:
```

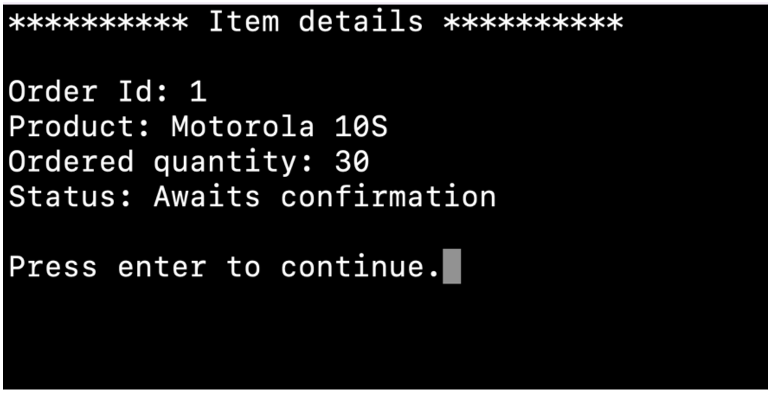
The admin can see all transactions made in the system, select one and inspect its details.



```
********** Item details **********

Id: 13
Transaction Type: REMOVAL
Product: Redmi WQ
Quantity: 1
Receipt number: 9292777346

Press enter to continue.
```

View Orders

The admin can also see the list of all the orders in the system, but she or he can’t manage them. The admin can instead inspect the order details.



```
********** Item details **********

Order Id: 1
Product: Motorola 10S
Ordered quantity: 30
Status: Awaits confirmation

Press enter to continue.
```

Additional Documentation

The project’s requirements, product backlog, use case diagrams and class diagrams can be found in the Project Management -directory of this project’s GitHub repository. These documents can further clarify the technical implementation of this app.