**Order Management System**

Create a java project for Order management system. Below are some high-level features in this app.

1. Add Order
2. View Order
3. View By Order Id
4. Sort Order
5. Delete Order by Id
6. Mark as Delivered.
7. Generate Report.
8. Exit

Create an Order Class with following properties.

* + Private Order Id: String
  + Private Order Description: String
  + Private Delivery Address: String
  + Private Order Date: LocalDateTime
  + Private Amount: Double
  + Private Delivery Datetime: LocalDateTime

Create and interface ‘OrderManagement’ for all the methods.

Create a text file called ‘OrderManagement.txt’ for storing the Order Data in file system.

On Start of application show the following menu.

Menu:

\*\*\*\*\*\*\*\*\*\*\*\*Order Management System\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Add Order
2. View Order List
3. View By Order Id
4. Sort Order
5. Delete Order by Id
6. Mark as Delivered.
7. Generate Report.
8. Exit

Choose Option:

Read the ‘OrderManagement.txt’ file to load the Order Details into a list of Order Object

If no order details available into text file, then create an empty list of Order Object.

All the operations will be performed on the list of the orders and if user selects “Exit” option then update the OrderManagement.txt file with current order list and then exit. Next time when the program starts, it should load all the orders from the text file before user can do any operation.

Add Order:

User can add the order by entering following values.

* + Order Id
  + Order Description
  + Delivery Address
  + Order Date: Order date will be current local datetime.
  + Amount

After adding the order show the message “Order Added Successfully”

Then show “Do you want to enter more order details(Y/N)

If Y: Enter another order.

If N: Show the menu.

Validation:

Order Id must be unique if user enters a duplicate id, then show message

‘Duplicate Order Id. Please enter unique order id’

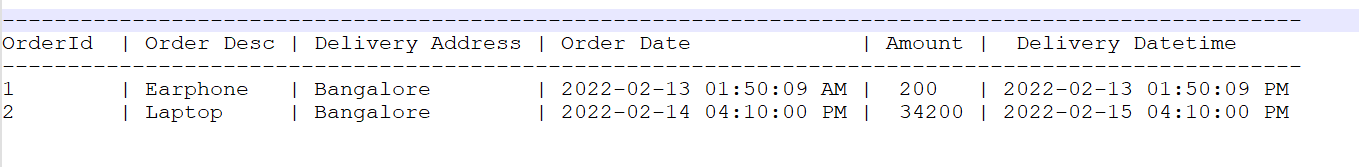
Ask user to add new order id.

If user inputs different data for selected property than do proper error handling.

Note: For View Order List and View by Order Id use the method overloading.

View Order List:

User can generate the list of all the order details. Show all the Order details in console.



After that show the menu options.

View By Order Id:

User will input order id and app will show complete order detail.

Text

Description automatically generated with medium confidence

After that show the menu options.

Sort Order:

Show a Sub-menu and ask for Sort property input.

\*\*\*\*\*\*\*\* Choose Sort Order Property\*\*\*\*\*\*\*\*\*

* 1. OrderId
  2. Order Desc
  3. DeliveryAddress
  4. Order Date
  5. Amount
  6. Delivery Datetime
  7. Exit

Sort List based on selected option and show the result on the screen.

Show Message “Successfully Sorted by <Property>

Show the main menu.

If select Exit Option Show the main menu.

Mark as Delivered:

User will input order id and app will update the Delivery Date with current local datetime value.

And show the message successfully delivered.

Then show “Do you want to mark another Order as Delivered(Y/N)

If Y: Enter another order id.

If N: Show the menu.

Validation: If Order is already delivered then show message ‘Order is already delivered on <LocalDatetime>’

Delete Order by Id:

User will input order id and the app will delete the order from the list.

And show the message ‘Order deleted Successfully’.

Then show “Do you want to delete another order(Y/N)

If Y: Enter another order id.

If N: Show the menu.

Validation:

If order id is not available, then do proper error handling and show message “Order Id is not available.

Generate Report:

Generate a new text file with all the order details with proper formatting for delivered Order.

Text file name: Order\_Report\_<currenttimestamp>

Show the menu option.

Use Multithreading for generate report. And after generating show the message ‘Report Generated Successfully’

Exit:

Write all the Order details into the text file.

And exit from application.