

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Jul 27 12:52:01 2024

@author: alishahbazaman
"""

fleet = []
shipments = []
delivered = []

def main_menu():
    while True:
        print("\nMain Menu:")
        print("1. Fleet Management")
        print("2. Shipment Management")
        print("3. Delivery Management")
        print("4. Quit Application")
        choice = input("Select an option: ")

        if choice == '1':
            fleet_management_menu()
        elif choice == '2':
            shipment_management_menu()
        elif choice == '3':
            delivery_management()
        elif choice == '4':
            print("Goodbye!")
            break
        else:
            print("Invalid option. Please try again.")

def fleet_management_menu():
    while True:
        print("\nFleet Management Menu:")
        print("1. Add a vehicle")
        print("2. Update vehicle information")
        print("3. Remove a vehicle")
        print("4. View all vehicles")
        print("5. Quit fleet management")
        choice = input("Select an option: ")

        if choice == '1':
            add_vehicle()
        elif choice == '2':
            update_vehicle()
        elif choice == '3':
            remove_vehicle()
        elif choice == '4':
            view_all_vehicles()
        elif choice == '5':
            break
        else:
            print("Invalid option. Please try again.")

def add_vehicle():

```

```

vehicle_id = input("Enter Vehicle ID: ")
if not is_unique_vehicle_id(vehicle_id):
    print("Error: Vehicle ID must be unique.")
    return

vehicle_type = input("Enter Vehicle Type: ")
vehicle_capacity = input("Enter Vehicle Capacity: ")
if not vehicle_capacity.isdigit() or int(vehicle_capacity) <= 0:
    print("Error: Vehicle Capacity must be a positive integer.")
    return

fleet.append((vehicle_id, vehicle_type, int(vehicle_capacity)))
print("Vehicle added successfully.")

def update_vehicle():
    vehicle_id = input("Enter Vehicle ID to update: ")
    vehicle = find_vehicle_by_id(vehicle_id)
    if vehicle is None:
        print("Error: Vehicle ID does not exist.")
        return

    vehicle_type = input("Enter new Vehicle Type: ")
    vehicle_capacity = input("Enter new Vehicle Capacity: ")
    if not vehicle_capacity.isdigit() or int(vehicle_capacity) <= 0:
        print("Error: Vehicle Capacity must be a positive integer.")
        return

    fleet.remove(vehicle)
    fleet.append((vehicle_id, vehicle_type, int(vehicle_capacity)))
    print("Vehicle information updated successfully.")

def remove_vehicle():
    vehicle_id = input("Enter Vehicle ID to remove: ")
    vehicle = find_vehicle_by_id(vehicle_id)
    if vehicle is None:
        print("Error: Vehicle ID does not exist.")
        return

    confirmation = input("Are you sure you want to remove this vehicle? (yes/no): ")
    if confirmation.lower() == 'yes':
        fleet.remove(vehicle)
        print("Vehicle removed successfully.")
    else:
        print("Vehicle removal cancelled.")

def view_all_vehicles():
    if not fleet:
        print("No vehicles in the fleet.")
        return

    print("\nFleet Vehicles:")
    print("-" * 45)
    print("ID".ljust(15, " ") + " | " +
          "Type".ljust(15, " ") + " | " +
          "Capacity")

```

```

print("-" * 45)
for vehicle in fleet:
    print(vehicle[0].ljust(15, " ") + " | " +
          vehicle[1].ljust(15, " ") + " | " +
          str(vehicle[2]).ljust(20, " "))
print("-" * 45)

# Helper function to check if vehicle ID is unique
def is_unique_vehicle_id(vehicle_id):
    return all(vehicle_id != vehicle[0] for vehicle in fleet)

# Helper function to find vehicle by ID
def find_vehicle_by_id(vehicle_id):
    for vehicle in fleet:
        if vehicle[0] == vehicle_id:
            return vehicle
    return

def shipment_management_menu():
    while True:
        print("\nShipment Management Menu:")
        print("1. Create a new shipment")
        print("2. Track a shipment")
        print("3. View all shipments")
        print("4. Quit shipment management")
        # Displays all options
        choice = input("Select an option: ")
        # Input option

        if choice == '1':
            create_shipment()
        elif choice == '2':
            track_shipment()
        elif choice == '3':
            view_all_shipments()
        elif choice == '4':
            break
        # Input choices redirecting to respective defined functions
        else:
            print("Invalid option. Please try again.")
            # If input choice is invalid

def create_shipment():
    shipment_id = input("Enter Shipment ID: ")
    if not is_unique_shipment_id(shipment_id):
        print("Error: Shipment ID must be unique.")
        return
    # Error and return if ID already exist
    origin_location = input("Enter the origin location of the shipment: ")
    destination_location = input("Enter the destination location of the shipment: ")

    weight = input("Enter Vehicle Weight: ")
    if not weight.isdigit() or float(weight) <= 0:
        print("Error: Vehicle weight must be a positive numeric value.")

```

```

        return
    # Error and return if input is not a positive numerical

    print(view_all_vehicles())
    vehicle_id = input("Choose Vehicle ID: ")
    if not find_vehicle_by_id(vehicle_id):
        print("Error: Vehicle ID not in the fleet.")
        return
    # Error and return if the vehicle ID doesn't exist

    delivery_status = "In Transit"
    # Shipment changed to In Transit if all inputs succeed

    shipments.append((shipment_id, origin_location, destination_location,
float(weight),
                        vehicle_id, delivery_status))
    # All values added to the Shipments list
    print("Shipment added successfully.")

def track_shipment():
    shipment_id = input("Enter Shipment ID: ")
    if not find_shipment_by_id(shipment_id):
        print("Error: Shipment ID not found.")
        # Error if the shipment ID does not exist
    else:
        print("Shipment Status:", find_shipment_by_id(shipment_id)[-1])
        # Shipment status which is the last value of the sublist printed

def view_all_shipments():
    if not shipments:
        print("No shipments in the system.")
        return
    # Error message if there are not shipments in the system

    print("\nAll Shipments:")
    print("-" * 110)
    print("Shipment ID".ljust(15, " ") + " | " +
          "Origin Location".ljust(15, " ") + " | " +
          "Destination Location".ljust(15, " ") + " | " +
          "Weight".ljust(15, " ") + " | " +
          "Vehicle ID".ljust(15, " ") + " | " +
          "Delivery Status")
    print("-" * 110)
    # Headings printed in tabular form

    for shipment in shipments:
        print(shipment[0].ljust(15, " ") + " | " +
              shipment[1].ljust(15, " ") + " | " +
              shipment[2].ljust(20, " ") + " | " +
              str(shipment[3]).ljust(15, " ") + " | " +
              shipment[4].ljust(15, " ") + " | " +
              shipment[5].ljust(15, " "))
        print("-" * 110)
    # Values printed in tabular form

    exit_menu = input(print("To exit the menu, type 'exit': "))

```

```

if exit_menu.lower() == 'exit':
    return
# Exit menu takes back to menu if type "exit" in lower case

def is_unique_shipment_id(shipment_id):
    return all(shipment_id != shipment[0] for shipment in shipments)
    # Unique shipment function used in creating shipment function to ensure the ID
    added
    # doesn't exist

def find_shipment_by_id(shipment_id):
    for shipment in shipments:
        if shipment[0] == shipment_id:
            return shipment
    return
    # Finding shipment by ID function used in the track shipment function

def delivery_management():
    while True:
        print("\nDelivery Management Menu:")
        print("1. Mark Shipment Delivery")
        print("2. View delivery status for a shipment")
        print("3. Quit delivery management")
        choice = input("Select an option: ")

        if choice == '1':
            mark_shipment_delivery()
        elif choice == '2':
            view_delivery_status()
        elif choice == '3':
            break
        else:
            print("Invalid option. Please try again.")

def mark_shipment_delivery():
    shipment_id = input(" Enter Shipment ID: ")
    #show shipment details
    if not find_shipment_by_id(shipment_id):
        print("Error: Shipment ID not found.")
        # Error if the shipment ID does not exist
    else:
        for shipment in shipments:
            if shipment[0] == shipment_id:
                print(shipment)

    #do you want to change the shipment to leave? yes or no, yes will marked as
    delivered
    confirmation = input("Do you want to change the shipment to
    deliver? (yes/no): ")
    if confirmation.lower() == 'yes':
        delivered.append(shipment_id)
        shipments.remove(find_shipment_by_id(shipment_id))
        print("Shipment has been delivered.")
    else:
        print("Shipment still in transit.")

```

```
def view_delivery_status():
    for shipment in shipments:

        print("The shipment ID: " + shipment[0] + " is in transit.")
    for deliver_id in delivered:
        print("The shipment ID: " + deliver_id + " is delivered.")

#need to show view all shipment which have delivered and in transit


if __name__ == "__main__":
    main_menu()
```