**Week 1: Problem Analysis and System Design**

**1. Read the Project Description:**

* **DeliveryAgent Class**: Represents a delivery agent, including attributes like name, contact number, vehicle type, and availability.
* **DeliveryAgentManager Class**: Manages the collection of agents, providing CRUD operations.
* **CSVUtils Class**: Handles reading from and writing to CSV files for storing agent information.
* **Main Class**: Manages input/output and acts as the entry point of the application.

**Key Features and Requirements**

**Delivery Agent Management:**

* Register, update, and delete delivery agents.
* Categorize agents by type of goods delivered (documents, medical supplies, food, etc.).
* Manage delivery vehicles (bikes, motorcycles, cars, vans, trucks).
* CSV-based data storage for persistent data.

**2. Problem Analysis**

**Main Components (Classes)**

1. **DeliveryAgent**
   * **Attributes: name, contactNumber, vehicleType, availability.**
   * **Methods: constructor, getters/setters, toCSV(), fromCSV().**
2. **DeliveryAgentManager**
   * **Attributes: List<DeliveryAgent> agents.**
   * **Methods: addAgent(), updateAgent(), deleteAgent(), saveToCSV(), loadFromCSV(), getAgents().**
3. **CSVUtils**
   * **Methods: writeToCSV(), readFromCSV().**
4. **Main**
   * **Methods: main(), inputAgent().**

**Relationships**

* **DeliveryAgentManager manages multiple DeliveryAgent instances.**
* **CSVUtils interacts with DeliveryAgentManager for data persistence.**

**3. Class Responsibilities**

1. **DeliveryAgent**
   * **Responsibilities: Hold agent details, convert to/from CSV format.**
   * **Collaborations: Utilized by DeliveryAgentManager for CRUD operations.**
2. **DeliveryAgentManager**
   * **Responsibilities: Manage the list of delivery agents, handle CRUD operations, and data persistence.**
   * **Collaborations: Uses CSVUtils for reading/writing data.**
3. **CSVUtils**
   * **Responsibilities: Perform file operations for saving and loading agent data.**
   * **Collaborations: Called by DeliveryAgentManager.**
4. **Main**
   * **Responsibilities: Serve as the application entry point, handle user interactions.**
   * **Collaborations: Interacts with DeliveryAgentManager for managing agents.**

**4. UML Class Diagram:**

**classDiagram**

**class** DeliveryAgent **{**

**-** String name

**-** String contactNumber

**-** String vehicleType

**-** boolean availability

**+** DeliveryAgent**(**String, String, String, boolean**)**

**+** String getName**()**

**+** void setName**(**String**)**

**+** String getContactNumber**()**

**+** void setContactNumber**(**String**)**

**+** String getVehicleType**()**

**+** void setVehicleType**(**String**)**

**+** boolean isAvailable**()**

**+** void setAvailability**(**boolean**)**

**+** String toCSV**()**

**+** static DeliveryAgent fromCSV**(**String**)**

**}**

**class** DeliveryAgentManager **{**

**-** List<DeliveryAgent> agents

**+** DeliveryAgentManager**()**

**+** void addAgent**(**DeliveryAgent**)**

**+** void updateAgent**(**int, DeliveryAgent**)**

**+** void deleteAgent**(**int**)**

**+** void saveToCSV**(**String**)**

**+** void loadFromCSV**(**String**)**

**+** List<DeliveryAgent> getAgents**()**

**}**

**class** CSVUtils **{**

**+** static void writeToCSV**(**String, List<DeliveryAgent>**)**

**+** static List<DeliveryAgent> readFromCSV**(**String**)**

**}**

**class** Main **{**

**+** static void main**(**String[]**)**

**+** static DeliveryAgent inputAgent**(**Scanner**)**

**}**

    DeliveryAgentManager "1" **-->** "0..\*" DeliveryAgent **:** manages

    Main "1" **-->** "1" DeliveryAgentManager **:** uses

    Main "1" **-->** "1" CSVUtils **:** interacts with

    DeliveryAgentManager "1" **-->** "1" CSVUtils **:** uses

    CSVUtils "1" **-->** "0..\*" DeliveryAgent **:** processes

A screenshot of a computer

Description automatically generated