**Labs 06 & 07:** **Draw Class Diagram (System S1)**

**Lab Learning Outcomes (LLO)**

By completion of the lab the students should be able to

1. To identify the key classes, their attributes with data types, methods with arguments/parameters and relationship between the classes involved in the Traffic Violation Monitoring System.
2. Create a comprehensive class diagram to represent the system's implementation.

**Tools Required**

For example, [StarUML](https://staruml.io/), [Lucidchart](https://www.lucidchart.com/pages/), [penpot](https://penpot.app/) and [Figma](https://www.figma.com/) etc

**Task to Do**

In this lab, you will work on developing a class diagram that outlines the interactions between different classes in our proposed solution. This lab will provide you with experience in visualizing and analyzing system requirements using class diagram modeling. You may start by considering the followings directions:

1. **Understanding the Problem and Solution**
   1. Provide a brief overview of the Traffic Violation Monitoring System problem and its suggested solution. (Please refer to Lab 1)
   2. Emphasize the importance of designing a class diagram to represent the system's structure and relationships.
2. **Identifying Classes and Attributes:**
   1. Identify all possible classes that would be essential for the system's implementation. Don’t worry in the beginning, just keep writing, you can discard later, if something doesn't appear to be a class.
   2. Have them brainstorm and list attributes for each class, considering the data that needs to be captured and processed.
   3. Students are encouraged to think about the types of vehicles, traffic data, violations, and users involved.
3. **Defining Methods and Parameters**
   1. Define methods for each class that represent the actions or operations related to that class.
   2. Identify the parameters required for each method, such as input data or conditions.
   3. Emphasize the importance of accurately defining methods to reflect the system's functionality.
4. **Establishing Relationships**
   1. Provide examples of possible relationships between classes in the Traffic Violation Monitoring System.
   2. Determine the appropriate relationships and dependencies between the identified classes.
5. **Creating the Class Diagram**
   1. Pick up a suitable diagramming tool
   2. Create a class diagram that incorporates the identified classes, attributes, methods, and relationships.
   3. Don’t forget to label associations, multiplicity, and indicate aggregation or composition where relevant.

For your convenience, you may also consider the following classes, attributes, methods and relationship and draw a class diagram using one of the above tools. Please note that these classes, attributes, methods and relationships may not be sufficient, and you might come across new ones.

**Classes, Attributes and Methods:**

1. **TrafficCamera**
   * Attributes: cameraID, location (latitude, longitude), orientation
   * Methods:
     1. captureImageVideo(location: Location, timestamp: Timestamp)
     2. calibrateCamera()
2. **TrafficData**
   * Attributes: vehicleNumberPlate, speed, timestamp, location
   * Methods:
     1. processData(data: RawData)
     2. retrieveData(vehicleNumberPlate: NumberPlate)
3. **Violation**
   * Attributes: violationType, severity, evidence (image or video), timestamp
   * Methods:
     1. calculateSeverity(speed: Speed)
     2. generateNotification() **// parameters are missing, lets write them, if needed.**
4. **Driver**
   * Attributes: driverID, name, licenseNumber
   * Methods:
     1. registerDriver(name: Name, licenseNumber: LicenseNumber)
     2. updateProfile(driverID: ID)
     3. disputePenalty() **// parameters are missing, lets write them, if needed.**
5. **Notification**
   * Attributes: notificationID, message, timestamp
   * Methods:
     1. sendNotification(driverID: ID, message: Message)
     2. viewNotifications(driverID: ID)

**Relationships:**

1. **TrafficCamera** has an association with **TrafficData**
   1. Relationship: Aggregation (one-to-many)
   2. Explanation: A TrafficCamera captures multiple instances of TrafficData.
2. **TrafficData** is associated with **Violation**
   1. Relationship: Aggregation (one-to-one)
   2. Explanation: TrafficData contains information about a specific violation.
3. **Driver** is associated with **Violation**
   1. Relationship: Aggregation (one-to-many)
   2. Explanation: A driver can be associated with multiple traffic data records and violations.
4. **Notification** is associated with **Driver**
   1. Relationship: Aggregation (one-to-many)
   2. Explanation: A driver can receive multiple notifications.

**Questions to Answer:**

1. Please think about if we can use ‘TrafficPolice’, ‘VehicleSpeed’, ‘Penalty’ and ‘Pictures’ as a Class or not or anything else? Please justify your answer with proper reasoning and identify attributes and methods, if you think as Yes.
2. Do you think these classes, attributes, methods and relationships are sufficient to implement the detection of the following violations? If not, identify the all the classes with their attributes, methods and relationships to implement the detection of the following violations.
   1. Over speeding
   2. Crossing red lights
   3. Changing lanes abruptly or without indicators
   4. Not wearing seat belts
   5. Driving erratically or zigzagging
   6. Stopping beyond the stop line at red lights
3. What changes will be required in terms of defining classes, attributes, methods and relationships, if the violations are detected at intersections of roads or straight roads?
4. Makre sure to identify the input, processing details and output of all the methods?
5. Once you are done answering all the questions above, lets draw the class diagram.

Remember that it is a simplified representation, and you can adapt and modify the class diagram to suit your specific teaching & learning objectives and level of detail. You can adjust and expand upon these classes, attributes, methods, and relationships to align with the complexity and scope of the Traffic Violation Monitoring System in your context.

**Examples**

You may also get help from the following samples produced by the students. These samples may not be complete and accurate, so **you are required to complete them and identify any mistakes/errors or space for improvements or changes required.**

**Sample 1**

