**Lab 05:** **Draw Use Case Diagram (System S1)**

**Lab Learning Outcomes (LLO)**

By completion of the lab the students should be able to

1. Identifying the actors and possible use-cases for the system S1 as identified in Lab 1.
2. Develop a use-case diagram and show 'extend' or 'include' relationships between the use-cases.

**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 use case diagram that outlines the interactions between different actors and system functionalities in our proposed solution. This lab will provide you with hands-on experience in visualizing and analyzing system requirements using use case modeling. You may start by considering the followings directions:

1. **Understanding the Problem:** Begin by reviewing the provided problem description and suggested solution. Familiarize yourself with the major traffic violations and the desired features of the system that aims to tackle this issue.
2. **Identifying Actors and Use Cases:** Identify the key actors who will interact with the system, such as Traffic Police Officers and Vehicle Owners. Then, identify the primary use cases that represent various functionalities of the traffic violation system.
3. **Creating the Use Case Diagram:** Using a diagramming tool of your choice (such as draw.io or Lucidchart), create a use case diagram that visually represents the relationships between actors and use cases. Include the identified actors, use cases, and their associations.
4. **Defining Use Case Details:** For each use case, provide a brief description outlining its purpose and functionality. Also, consider the 'extend' and 'include' relationships between use cases as specified in the lab guidelines.
5. **Enhancing Use Case Descriptions:** Elaborate on the 'extend' relationship by describing when and how a use case extends another use case. Similarly, detail the 'include' relationship, explaining how one use case is included in another use case.

For your convenience, you may consider the following actors, use cases and relationships and convert them into use case diagram using one of the above tools. Please note that these actors, use cases and relationships may not be sufficient and you might come across new ones. Success scenarios and extensions are written for few use cases, you are required to write for the remaining use cases.

1. **Actors:**
   * Traffic Police Officer
   * Vehicle Owner
2. **Use Cases:**
   * **Capture Traffic Data**
     + Description: Captures traffic data using cameras or sensors.
     + Actor: System
     + Main Success Scenarios:
       - Cameras capture images/videos of vehicles at an intersection.
       - Data includes vehicle number plates, speed, and direction.
     + Extensions:
       - Camera Malfunction: If cameras fail to capture data, system alerts administrators and logs the issue.
       - Incomplete Data: If captured data is incomplete, system logs partial information and continues normal operation.
   * **Analyze Traffic Data**
     + Description: Analyzes captured data and extracts relevant information.
     + Actor: System
   * **Detect Violation**
     + Description: Identifies traffic violations by comparing data with rules.
     + Actor: System
     + Main Success Scenarios:
       - System analyzes captured data and identifies a red-light violation.
       - Violation severity level is determined based on speed and type.
     + Extensions:
       - No Violation Detected: If no violation is identified, system continues with normal data processing.
       - Detection Error: If violation detection algorithm fails, system logs the issue and alerts administrators.
   * **Classify Violation**
     + Description: Assigns severity level to detected violations.
     + Actor: System
   * **Verify Violation**
     + Description: Allows traffic police officers to review and confirm violations.
     + Actor: Traffic Police Officer
     + Includes: Generate Evidence
     + Main Success Scenarios:
       - Traffic Police Officer reviews images/videos of a red-light violation. (*NOTE THE JOB IS DONE BY THE POLICA OFFICER, IT IS NOT AUTOMATED. IDEALLY, WE NEED TO AUTOMATE THE PROCESS, WHAT CHANGES WILL BE REQUIRED*?)
       - Officer confirms violation and assigns a penalty.
     + Extensions:
       - Violation Not Confirmed: If evidence is unclear, officer requests additional information and temporarily suspends the violation for further review.
       - Verification Error: If verification process fails, officer logs the issue and forwards it to system administrators for resolution.
   * **Generate Evidence**
     + Description: Generates evidence (images/videos) for verified violations.
     + Actor: System
   * **Communicate with Vehicle Owner**
     + Description: Notifies vehicle owners about violations and penalties.
     + Actor: System
   * **View Violation Details**
     + Description: Allows vehicle owners to view details of violations.
     + Actor: Vehicle Owner
   * **Dispute Violation**
     + Description: Enables vehicle owners to dispute violations they disagree with.
     + Actor: Vehicle Owner
   * **Pay Penalty Online**
     + Description: Allows vehicle owners to pay penalties online.
     + Actor: Vehicle Owner
3. **Relationships:**
   * **Extend Relationships:**
     + Extend from **Verify Violation** to **Generate Evidence**
       - Description: When a violation is verified, evidence is generated for supporting verification.
   * **Include Relationships:**
     + Include **Verify Violation** in **Communicate with Vehicle Owner**
       - Description: The verification process is included when communicating with vehicle owners about violations.

**Notes:**

* The **Traffic Police Officer** and **Vehicle Owner** are the primary actors interacting with the system.
* Use cases represent specific functionalities of the system or actions performed by actors.
* The 'extend' relationship indicates that one use case (extended use case) can be added based on certain conditions of another use case (base use case).
* The 'include' relationship indicates that one use case (included use case) is a part of another use case (including use case).

Remember that this is a simplified representation, and you can adapt and modify the use case diagram to suit your specific teaching & learning objectives and level of detail. The relationships and interactions between use cases help students understand the flow and logic of the system's functionality.

**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**

A diagram of a diagram

Description automatically generated

**Sample 2**

A diagram of a company

Description automatically generated