Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Fall, Year:2024), B.Sc. in CSE (Day)

LAB REPORT NO #06
Course Title: Integrated Design Project 1
Course Code: CSE 324
Section:213 D7

Lab Experiment Name: Advanced Vehicle Tracking System:

UML Use Case Design.

Student Detail

Name		ID
1.	Md. Rajuan Hossen	221002100
2.	Hasebul Hasan	221002104

Lab Date : **18/07/2024**
Submission Date : **19/07/2024**
Course Teacher's Name : **Md. Romzan Alom**

[For Teachers use only: Don't Write Anything inside this box]

Introduction

A UML Use Case Diagram is a visual representation of the interactions between users (actors) and the system. It identifies the functional requirements and user interactions with the system. For the AVTS (Advanced AI-driven Vehicle Tracking System), the use case diagram illustrates how users such as customers and administrators interact with the system to perform tasks like tracking vehicles, calculating fares, and managing services.

Components of a UML Use Case Diagram

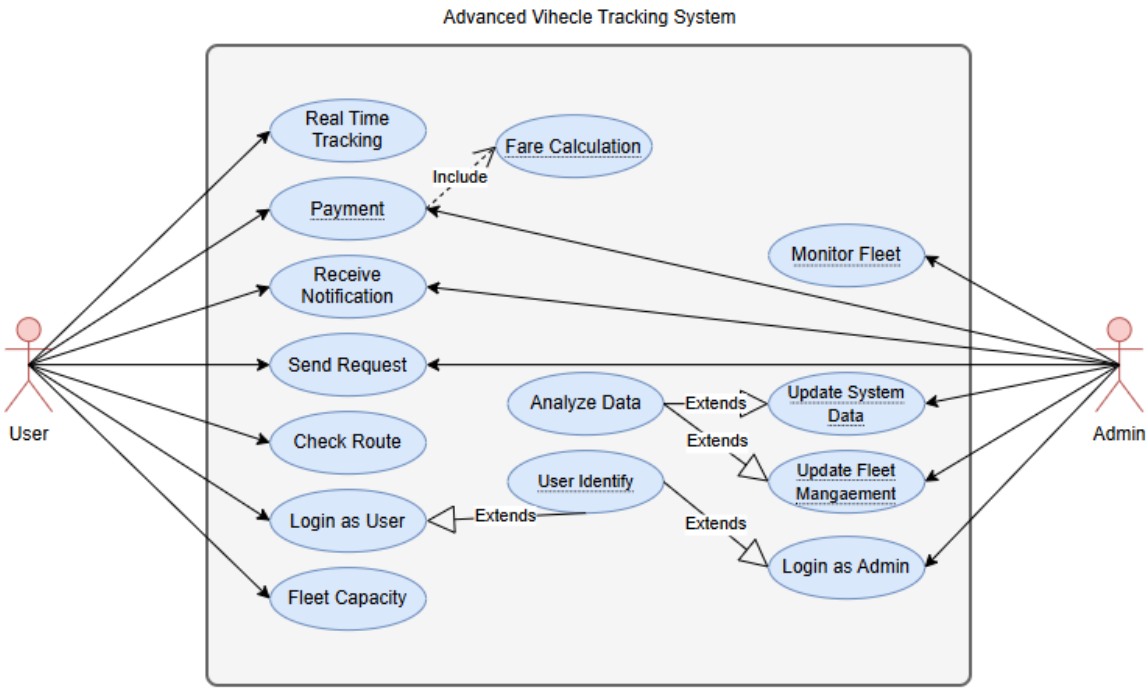
1. **Actors**: Represent users or external systems interacting with the system.
2. **Use Cases**: Specific functionalities or tasks performed by the system for the actor.
3. **Relationships**: Connections between actors and use cases (e.g., associations, generalizations, includes, extends).
4. **System Boundary**: Defines the scope of the system being modeled.

UML Use Case Diagram

The diagram includes the following actors and use cases:

- **Actors**: Customers, administrators, IoT devices, external APIs.
- **Use Cases**:
    - Real-time vehicle tracking
    - Monitoring bus capacity
    - Sending distance notifications
    - Requesting a pickup
    - Viewing available services
    - Notifying vehicle damage
    - Selecting a service
    - Calculating fares

Figure 1: UML Use Case Diagram



Discussion

Creating the UML Use Case Diagram involved identifying all the actors and their respective interactions with the AVTS system. One of the challenges was ensuring the inclusion of all essential use cases while avoiding redundancy. Additionally, defining relationships like "include" and "extend" required precise understanding of use case dependencies, which initially led to confusion in modeling optional and mandatory actions.