****

**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 |
|  | 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]**

|  |
| --- |
| **Lab Report Status**  **Marks: …………………………………                              Signature:.....................**  **Comments:..............................................                              Date:..............................** |

**Introduction**

The Advanced AI-driven Vehicle Tracking System (AVTS) is a next-generation platform designed to improve fleet management using AI, IoT, and machine learning. The system offers real-time vehicle tracking, predictive maintenance, route optimization, driver behavior monitoring, and customer notifications.

To ensure a clear understanding of system functionality, a Unified Modeling Language (UML**)** use case diagram was developed. UML diagrams help visualize the interactions between different actors (e.g., Fleet Manager, Drivers, Passengers) and system use cases. This report outlines the design process for the UML use case diagram and discusses potential challenges encountered during its development.

**Discussion**

Developing the UML use case diagram presented challenges, particularly in distinguishing actors like the Fleet Manager and System Administrator and defining their unique interactions with the system. Representing complex use cases such as Predictive Maintenance and Route Optimization required careful consideration to avoid redundancy. Incorporating IoT sensors added complexity due to their autonomous operations and multiple interactions. Scalability was difficult to represent within the static nature of UML diagrams, and tool limitations sometimes restricted clarity. Despite these challenges, iterative revisions helped address these issues, resulting in a functional and coherent diagram.