Title Slide

University of Ghana - Department of Computer Science

Course: DCIT308 - Data Structures and Algorithms II

Project Title: Vehicle Tracking & Maintenance System

Group Members: [Enter Names]

Presentation Date: July 17, 2025

Vehicle Tracking & Maintenance System - Slide Deck

Problem Statement

Adom Logistics manually tracks vehicles, drivers, and deliveries.

This leads to service delays, route mix-ups, and fuel inefficiency.

Goal: Build an offline console-based system using only core data structures.

Vehicle Tracking & Maintenance System - Slide Deck

System Overview

Modules Implemented:

- Vehicle Database
- Driver Assignment
- Delivery Tracking
- Maintenance Scheduler
- Fuel Reports & Sorting
- File Storage (Offline Support)

Vehicle Database

Structure Used: HashMap

Why: Fast lookup by registration number

Operations: Add, Remove, Search, Sort by mileage/type

Time Complexity: O(1) average-case lookup

Driver Assignment

Structure Used: Queue

Why: FIFO - Assign earliest available driver

Operations: Enqueue, Dequeue, Rotate

Time Complexity: O(1) enqueue/dequeue

Delivery Tracking

Structure Used: LinkedList

Why: Dynamic insertion, status updates, rerouting

Operations: Add, Update, Search deliveries

Time Complexity: O(n) traversal

Maintenance Scheduler

Structure Used: Min-Heap

Why: Prioritize vehicles needing urgent service

Operations: Insert, Remove Min

Time Complexity: O(log n)

Fuel Efficiency & Sorting

Algorithms Used:

- Insertion Sort (Mileage)

- Merge Sort (Driver Name)

- Quick Sort (Fuel Usage)

Complexity: O(n^2) to O(n log n)

Filters: By vehicle type, performance

Search Operations

Binary Search used for reg number lookup in sorted data

Time Complexity: O(log n)

Prerequisite: Data must be sorted

File Storage

All records saved in .txt files for offline use

Files:

- vehicles.txt
- drivers.txt
- deliveries.txt
- maintenance.txt

I/O: BufferedReader and PrintWriter

Summary Table

Module	Structure / Algorithm	Time Complexity
Vehicles	HashMap	O(1)
Drivers	Queue	O(1)
Deliveries	LinkedList	O(n)
Maintenance	Min-Heap	O(log n)
Sorting	Insertion/Merge/Quick	O(n^2)/O(nlogn)
Search	Binary Search	O(log n)

Conclusion

The system uses efficient, appropriate data structures for real logistics needs.

It demonstrates strong practical understanding of DSA concepts.

Project meets all requirements: offline, fast, organized, and modular.