UNIVERSITY OF TECHNOLOGY, JAMAICA

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

ADVANCED PROGRAMMING (CIT3009)

GROUP PROJECT (25%)

Due Date: November 17, 2025 Interview: Week 13

Project: SmartShip Package Management System

Overview: SmartShip Ltd. is a courier company that ships packages for customers locally and internationally. The company wants to modernize its operations with a system that allows customers to create shipment requests, track packages, and receive invoices. Staff members should be able to manage vehicles and delivery schedules, ensuring no vehicle is overbooked or double-assigned. Managers require reports on shipments, revenue, and delivery performance.

Functional Requirements

1. User & Customer Management – dantae

- Roles: Customer, Clerk, Driver, Manager. → Jasses
- Customers can create accounts, log in, and request shipments.
- Clerks process shipment requests, assign packages to delivery routes, and handle payments.
- Drivers can view their assigned deliveries and update package status (e.g., *In Transit, Delivered*).
- Managers can manage user accounts and oversee all operations.

2. Shipment Module Raheem

- Customers can create a **shipment order**: specify sender info, recipient info, package weight, dimensions, type (standard, express, fragile), and destination (Address + Zone 1, 2, 3 or 4).
- System calculates shipping cost based on weight, distance, and type.

- Shipment statuses: *Pending, Assigned, In Transit, Delivered, Cancelled*.
- Tracking numbers generated automatically.
- Customers can track packages online via their account.
- Distance is determined by zone. There are four zones (zone 1 to 4), zone 4 being the furthest distance
    - Distance can be a randomly generated number based on the zone

3. Vehicle & Scheduling Module dav

- The company has a fleet of vehicles with capacity limits of both weight and quantity.
- Clerks or managers assign packages to vehicles and routes.
- System must prevent assigning more packages than a vehicle's available capacity.
- Vehicle schedules should not overlap (a vehicle can't be in two places at once).
- Drivers log in to see their route and packages for the day.

4. Billing & Payment Module AI

- Invoices generated for every shipment.
- Customers can pay via cash or card.
- Status: *Paid / Partially Paid / Unpaid*.
- Discounts or surcharges (e.g., express delivery, fragile handling).
- Receipts are displayed or PDF exported.

5. Reporting Module Dav

- Managers can generate reports:
    - Daily/weekly/monthly shipments.
    - Delivery performance (on-time vs delayed).
    - Revenue reports.
    - Vehicle utilization (how full each vehicle runs).
- Reports exportable to **PDF**.

TOMAI

Entity & Design Requirements (due Week 2)

- **ER Diagram** with relationships:
    - Customer ↔ Shipment
    - Shipment ↔ Vehicle (many-to-many resolved by Assignment)
- **Class Diagram** mapping business objects (e.g., User, Customer, Shipment, Vehicle, Invoice, Payment).
- Ensure **3NF normalization**.

Software Architecture -dantae

- **Java Client/Server application** with database hosted on server.
- **GUI**:
    - Customer portal: create shipments, track packages, view/pay invoices.
    - Clerk portal: assign shipments to vehicles, update status, manage payments.
    - Driver portal: view assigned shipments, update delivery status.
    - Manager portal: reporting, fleet management, user management.
- **Concurrency**: handle multiple shipment requests simultaneously without overbooking vehicles.

Extra Credit

1. **Threaded Server & Multi-user handling (5%)**
   Server handles concurrent shipment bookings and assignments safely.
2. **Formatted PDF Reports (5%)**
   Professionally formatted reports with tables, totals, and charts (if possible).

Assessment Breakdown

Criteria	Marks
UML Class and Database ER Diagrams	5%
Comments, Programming Conventions, Documentation	5%
Graphical User Interface	10%
Client/Server Networking Model	15%
Database Connectivity (mandatory) / File Use (optional)	15%
Logging, Exception Handling, Validation	10%
Classes, Interfaces, Inheritance, Polymorphism	15%
Core Functionality (requirements above)	25%
Total	**100%**

Project Guidelines

- Team project: **4–5 persons** per group
- Programming conventions:
    - **Classes**: Capitalized (e.g., Person)
    - **Functions**: camelCase (e.g., getName())
    - **Variables**: meaningful names; lowercase with camelCase (e.g., registrationStatus)

- Avoid single-letter variable names (e.g., x)
- Proper indentation and comments required:
    - Each function documented
    - Important lines briefly explained
- Continuous assessment:
    - Project is worth **20% of final grade**
    - **5%** assessed weekly for milestone progress
    - Missing milestones results in loss of weekly grade

Late Submission

As per the university's policy, late submission of projects will attract a penalty as follows:

Time of Submission after Deadline	Penalty (%)
One Day (or any part thereof)	10
Two Days (or any period > 1 day, but <= 48 hrs.)	20
Three Days (or any period > 2 days, but <= 72 hrs.)	50
Beyond Three Days	100