<u>Junior intern assignment – Dan Ribak</u>

Following are my answers:

<u>Analysis of Delay Factors</u>: Briefly describe the common and potential causes of shipment delays for a logistics company of this scale. Focus on both operational factors and external factors.

These are some of the common and potential causes of shipment delays for a logistics company operating at scale:

Operational (Internal) Factors:

1. **Warehouse congestion and slow processing:** High warehouse load, staffing shortages, or inefficiencies in warehouse management systems (WMS) can slow down shipment handling.
2. **Equipment or vehicle failures:** Mechanical issues with trucks, forklifts, or conveyor systems may cause unplanned downtime.
3. **Communication and coordination issues:** Lack of synchronization between transport, warehouse, and customer service systems leads to fragmented data and late responses.
4. **Capacity overload:** Seasonal demand peaks (e.g., holidays, major sales) that exceed operational capacity can create fulfillment bottlenecks.

External Factors:

1. **Adverse weather conditions**: Storms, snow, heavy rain, or extreme heat disrupt road, air, and sea transportation.
2. **Port or border congestion:** High traffic at ports or customs checkpoints, Customs and regulatory delays, extended inspections, missing permits, or trade regulations slow cross-border shipments.
3. **Political or social instability**: Strikes, protests, wars, security issues, or geopolitical conflicts disrupting transport routes.

Early Delay Identification Logic: Present a simple flowchart or a clear textual description outlining your proposed logic for early delay identification.

To proactively detect shipments at risk of delay, the system applies three complementary logic rules, each addressing a different type of risk: operational progress, inactivity, and external disruption.

Logic 1: Stage Progress Rule

Rule:

Checks whether a shipment remains in an early operational stage relative to its expected delivery date.

Explanation:

This rule identifies potential operational or status lag. When a shipment has not progressed to the final distribution stages despite the delivery date approaching, it may indicate a risk of delay

Required Data:

- Current shipment stage/status
- Expected delivery date (ETA)
- Last stage update timestamp (optional)

Logic 2: Inactivity

Rule:

Checks whether a shipment has not reported any status or location update within a defined time threshold, while its status is still not marked as Delivered.
Optionally, each delivery stage may have its own inactivity threshold.

Explanation:

This rule detects potential inactivity or stalled movement.
When no update is recorded for an extended period, it likely indicates that the shipment is stuck at some point along the supply chain.
Such inactivity patterns serve as a strong early indicator of possible delivery delays.

Required Data:

- Current shipment stage/status
- Last location or checkpoint timestamp
- Threshold table

Logic 3: External Data Rule

Rule:

Checks whether external conditions, such as severe weather, port congestion, or customs delays, are reported along the shipment's route **and** the estimated time of arrival (ETA) falls within a defined threshold (e.g., the next five days).
If both conditions are met, the shipment is marked as "At risk."

Explanation:

This rule integrates external contextual data to enhance early-warning accuracy.
By correlating the shipment's route and ETA with external data sources (e.g., weather APIs, port congestion indexes, or customs reports), the system can proactively flag shipments likely to be delayed due to factors beyond operational control.

Required Data:

- Shipment route (origin, destination, checkpoints)
- ETA
- External data sources (weather, port congestion, customs reports)

Machine Learning: Data fields required

For a future ML-based delay prediction model, the system should incorporate both existing operational data and additional contextual inputs to enable accurate, explainable predictions.

In addition, the model must have access to the final delivery outcome, (i.e. isDelayed = true/false) whether each shipment was ultimately delayed or delivered on time, to serve as the training label for supervised learning.

The model's objective is to learn correlations between shipment attributes, operational behavior, and actual delivery outcomes.

Required fields:

Core operational and time related data:

- Current shipment stage and status
- Expected delivery date (ETA)
- Last checkpoint timestamp
- Time spent in each stage
- Actual vs. planned delivery duration

External contextual data:

- Weather conditions along the route

- Port congestion and customs delays

Indirect and contextual metadata:

- Shipment type and service level (standard, express, temperature-controlled, etc.)
- Origin and destination countries or regions
- Route complexity (distance, number of borders crossed, transport mode)
- Carrier ID and reliability metrics

Target variable:

- Delivery outcome → isDelayed = true / false

Alert Triggering Approach

In the initial stage, I would choose a rule-based alerting system.
It allows me to implement clear, explainable logic that can be quickly tested and trusted by the operations team, especially while the available data is still limited.

Later on, as more shipment data accumulates, I would integrate a machine-learning component alongside the rule-based system, not to replace it, but to enhance it. This combined approach would keep the alerts transparent and reliable while gradually improving accuracy through data-driven learning.

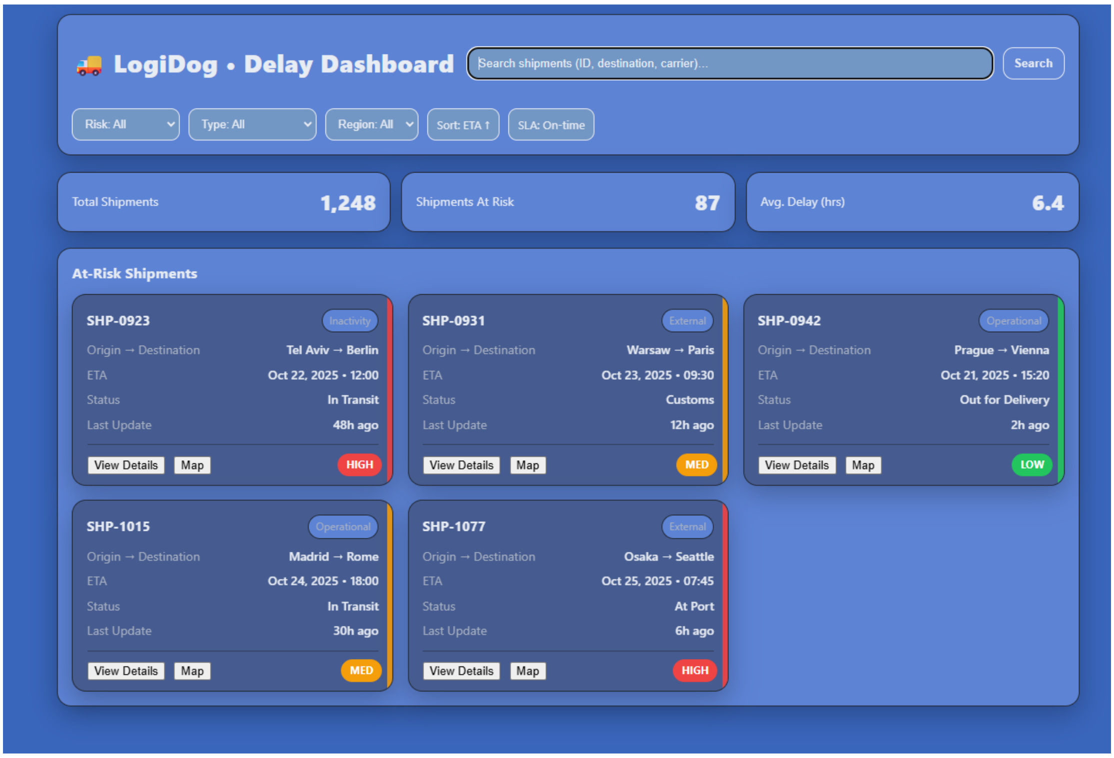
Approach	Advantages	Disadvantages
Rule-Based System	1. Simple and transparent: easy to understand, validate, and adjust. 2. Fast implementation: can be deployed immediately	1. Static logic: requires manual updates when conditions or business rules change. 2. Limited adaptability: cannot automatically learn from new data patterns.
Machine-Learning System	1. Adaptive: continuously improves as more data becomes available. 2. Pattern recognition: detects complex relationships beyond human-defined rules.	1. Requires large, high-quality data: performance depends on sufficient historical data. 2. Less explainable: difficult for users to fully understand how alerts are generated. 3. Higher maintenance – needs monitoring, retraining, and model tuning over time.

User Interface (UX/UI) Design for a Primary Alerts Screen

Visual Design:

A visual mock-up of the proposed dashboard is attached below for reference.
The design illustrates the primary layout, shipment cards, filters, and risk indicators for at-risk shipments.

The complete functional HTML version of this dashboard prototype is included in the submitted repository (dashboard.html) as part of this assignment.



Real-time Data Update:

For real-time updates, I would use a hybrid communication model combining REST APIs and WebSockets.
The dashboard initially loads data via a standard API request to retrieve the current list of at risk shipments.
After loading, it opens a persistent WebSocket connection to receive push notifications for any shipment status or risk-level changes.
This approach ensures both fast initial loading and instant synchronization without unnecessary polling, making it ideal for a logistics alert system.

Data Structure for Display:

For displaying shipment data on the dashboard, I would use a **denormalized data structure** optimized for fast reads.

Each shipment record includes all relevant fields, such as ETA, current stage, risk type, so the dashboard can render it directly without multiple joins.

To improve performance, the **backend periodically aggregates** data from multiple collections (e.g., shipments, events, and external sources) into a pre-computed view collection.

Example:


```
{
  "shipmentId": "SHP-0923",
  "origin": "Tel Aviv",
  "destination": "Berlin",
  "ETA": "2025-10-22T12:00:00Z",
  "currentStage": "In Transit",
  "riskType": "Inactivity",
  "riskLevel": "High",
  "lastUpdate": 1760863350000
}
```