## **🗓️ Phase 2: Intelligence Analysis, Scoring, and Alerting**

**Objective:** Add logic and features that analyze intel, track reporter and target metrics, detect threats, and produce actionable reports.

#### **✅ 0. Build a Dal class in C#**

### **Encapsulate all direct interactions with the database inside a Dal class or service**

### **This class should include methods such as:**

### **GetPersonByName()**

### **GetPersonBySecretCode()**

### **InsertNewPerson()**

### **InsertIntelReport()**

### **UpdateReportCount()**

### **UpdateMentionCount()**

### **GetReporterStats()**

### **GetTargetStats()**

### **CreateAlert()**

### **GetAlerts()**

#### **✅ Benefits and Guidelines:**

### **Keeps SQL logic separate from business logic**

### **Makes testing and refactoring easier**

### **Use parameterized queries to prevent SQL injection**

### **Store the connection string securely (not hardcoded)**

### 

### **This DAL implementation should replace any raw SQL calls previously written in the main program from Phase 1.**

### 

### 

### **📊 Reporter Analysis: Scoring & Promotion**

#### **✅ 1. Evaluate Reporter Performance**

* For each person marked as reporter or both:  
  + Count total number of reports (num\_reports)
  + Calculate the **average length** of all their intel texts

#### **✅ 2. Promote to potential\_agent**

* If both of the following are true:  
  + The reporter has submitted **10 or more reports**
  + The average text length is **≥ 100 characters**
* Then:  
  + Change their type field to potential\_agent
  + Log this promotion

Reporter eligibility should be evaluated **every time** a new intel is added.

### **🎯 Target Analysis: Mentions & Risk Detection**

#### **✅ 3. Identify Dangerous Targets**

* For each person marked as target or both:  
  + Count total mentions (num\_mentions)
* If a target has been mentioned in **20 or more reports**:  
  + Mark them as **dangerous**
  + Log this status (you can optionally add a flag in code or DB)

This check should also run after every new intel entry.

### **⏱️ Burst-Based Alert System**

#### **✅ 4. Analyze Temporal Patterns**

* Check for targets who are reported **3 or more times within a 15-minute window**
* If such a burst is found:  
  + Create an **alert**
  + Record the:  
    - target\_id
    - Time window (start and end)
    - Reason (e.g., "Rapid reports detected")

#### **✅ 5. Store and Show Alerts**

* Optionally create an Alerts table with:  
  + id, target\_id, created\_at, reason
* Display current alerts in a summary view

### **📋 Reporting Interface (Basic Output View)**

#### **✅ 6. Create an Analysis Menu**

The C# application should include a console (or UI) menu to:

* List all **potential agents**:  
  + Name, report count, average length
* List all **dangerous targets**:  
  + Name, mention count
* List all **active alerts**:  
  + Target, timestamp, description

You may allow sorting by counts or filtering by thresholds.

### **🧪 Logging, and Stability**

#### **✅ 7. Logs**

* Continue logging:  
  + Promotions to agent
  + Dangerous targets flagged
  + Alerts generated

## **✅ Phase 2 Deliverables**

* Reporter scoring and automatic promotion
* Target risk detection and burst alerting
* Alert logging and optional alert table
* Reporting interface (console or UI)
* Additional logs and test coverage
* Updated database (if Alerts table added)

## **⭐ Extras (Optional / Bonus Features)**

These features are not required for the core 2-day build but are great stretch goals.

### **🔐 Encryption**

* Encrypt text column when saving intel
* Decrypt only when needed for analysis
* Use symmetric encryption (e.g., AES or XOR for demonstration)

### **💰 Payment System**

* Add a Payments table:  
  + Reporter, amount, reason, timestamp
* Define payment logic:  
  + Pay per word
  + Bonuses for high-priority targets or burst alerts

### **🖥️ UI Improvements**

* Replace console interface with WinForms/WPF:  
  + Dropdowns for selecting users
  + Text areas for reports
  + Tables for alerts and stats

Would you like a **combined printable brief**, a **lesson delivery plan**, or a **clickable project board** version of both phases?