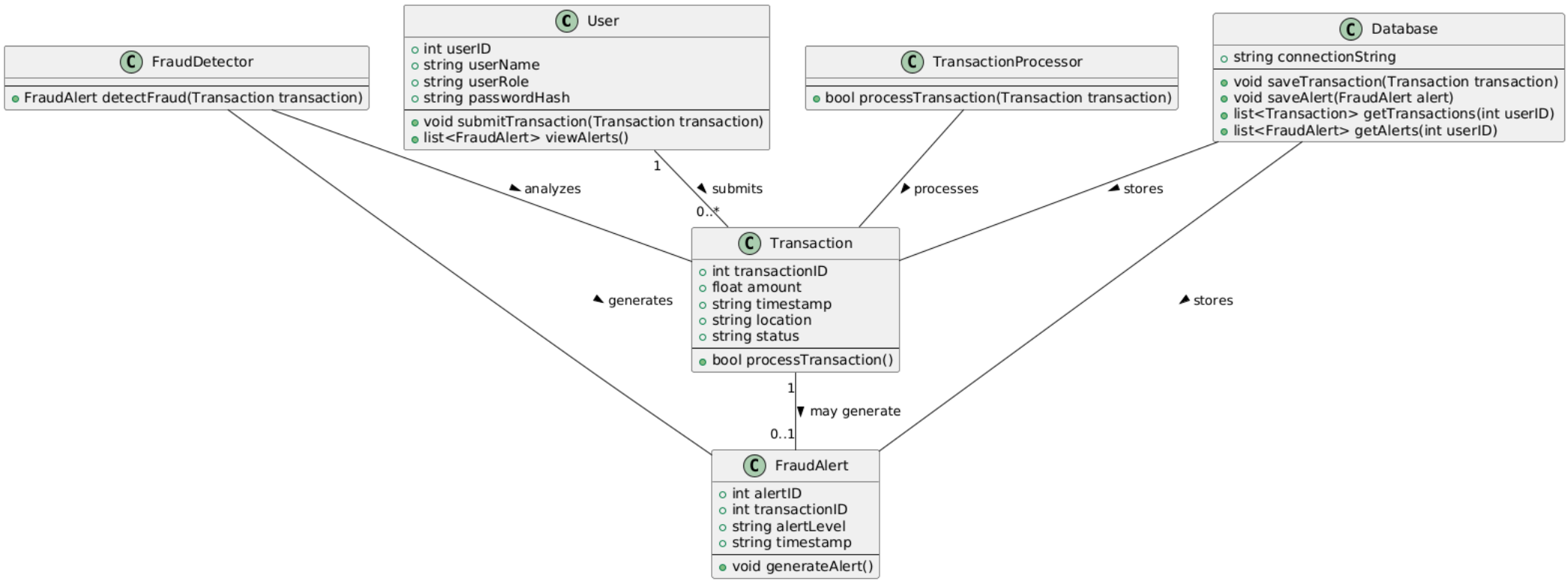
8a: Class Diagram



8b: Data Types and Operation Signatures

For each class, we define the data types of all attributes and the signatures of operations.

1. **User Class**:
    - **Attributes**:
        - userID: int
        - userName: string
        - userRole: string
        - passwordHash: string
    - **Operations**:
        - submitTransaction(transaction: Transaction): void
        - viewAlerts(): list<FraudAlert>
2. **Transaction Class**:
    - **Attributes**:
        - transactionID: int
        - amount: float
        - timestamp: string
        - location: string
        - status: string
    - **Operations**:
        - processTransaction(): bool
3. **FraudAlert Class**:
    - **Attributes**:
        - alertID: int
        - transactionID: int
        - alertLevel: string
        - timestamp: string
    - **Operations**:
        - generateAlert(): void
4. **TransactionProcessor Class**:
    - **Operations**:
        - processTransaction(transaction: Transaction): bool
5. **FraudDetector Class**:
    - **Operations**:
        - detectFraud(transaction: Transaction): FraudAlert
6. **Database Class**:
    - **Attributes**:
        - connectionString: string

- **Operations**:
  - saveTransaction(transaction: Transaction): void
  - saveAlert(alert: FraudAlert): void
  - getTransactions(userID: int): list<Transaction>
  - getAlerts(userID: int): list<FraudAlert>

8c: Traceability Matrix

The **Traceability Matrix** outlines how each class evolved from the domain concepts identified in the earlier stages of the project. Here, we map domain concepts to the corresponding classes in the system:

Domain Concept	Class	Rationale
User	User	Represents individuals interacting with the system
Transaction	Transaction	Each transaction corresponds to a financial operation
Fraud Alert	FraudAlert	Alerts generated when suspicious activity is detected
Transaction Processor	TransactionProcessor	Handles the processing of transactions
Fraud Detection	FraudDetector	Analyzes transactions for suspicious patterns
Data Storage	Database	Manages persistent storage of user, transaction, and alert data

Explanation:

- The **User** class directly evolved from the domain concept of "User" who submits transactions.

- The **Transaction** class evolved from the concept of financial transactions submitted by the users.
- The **FraudAlert** class corresponds to the fraud detection concept, which was originally derived from the need to alert suspicious transactions.
- **TransactionProcessor** and **FraudDetector** emerged from the need to separate processing and detection responsibilities, adhering to the **Single Responsibility Principle**.
- **Database** was introduced to handle persistent storage, evolving from the need to store data long-term.

This matrix shows the traceability from domain concepts to class structure, ensuring that all requirements are met effectively through the system's design.