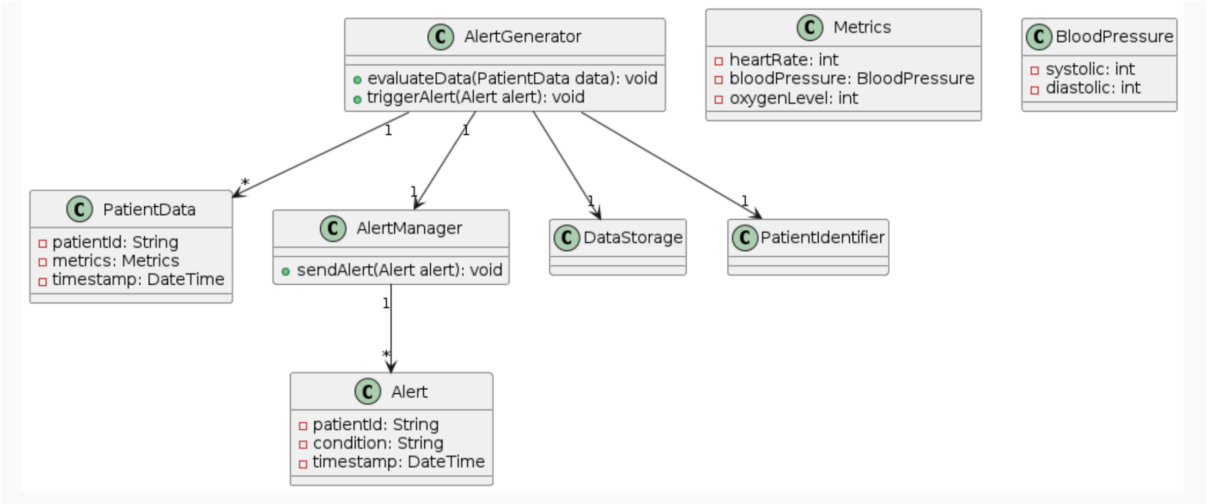
Alert Generation System



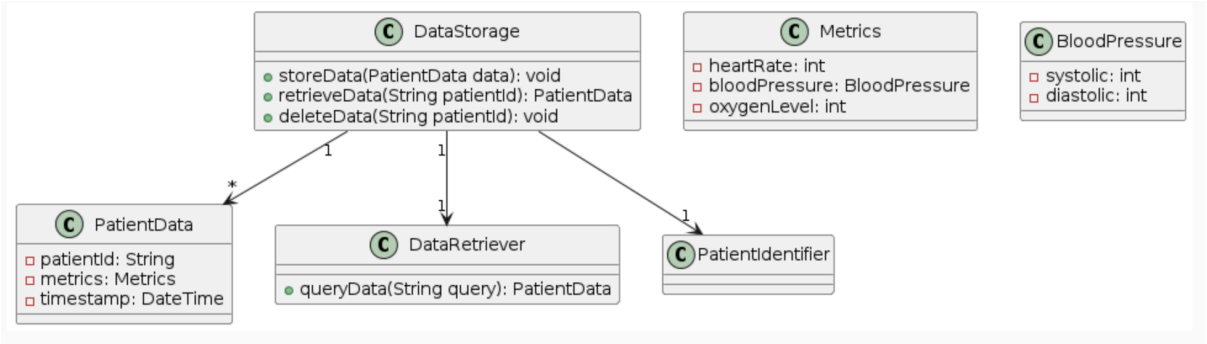
Purpose:
The Alert Generation System is responsible for continuously monitoring incoming patient data, evaluating it against predefined criteria and generating alerts when specific conditions indicating patient distress are detected.

Functionality of Each Class:

- PatientData: Represents the data for a patient at a specific point in time, including patient ID, metrics (such as heart rate, blood pressure, oxygen level), and timestamp.
- Metrics: Represents the various metrics measured for a patient, such as heart rate, blood pressure, and oxygen level.
- BloodPressure: Represents the blood pressure metrics, including systolic and diastolic readings.
- AlertGenerator: Monitors incoming patient data streams, evaluates the data against predefined criteria, and triggers alerts when specific conditions are met.
    - evaluateData(PatientData data): Method to evaluate incoming patient data against predefined criteria.
    - triggerAlert(Alert alert): Method to trigger an alert based on evaluated data.
- Alert: Represents an alert triggered by the system, including patient ID, the specific condition that triggered the alert, and a timestamp.
- AlertManager: Manages alerts generated by the system and ensures they are sent to appropriate medical staff's mobile devices and workstations.

Interactions:
AlertGenerator interacts with PatientData to receive incoming patient data streams.
AlertGenerator evaluates incoming data using methods defined in Metrics and BloodPressure classes.
If a condition meets predefined criteria, AlertGenerator triggers an alert by creating an instance of Alert.
AlertManager receives alerts generated by AlertGenerator and manages their dissemination to medical staff.

Data Storage System



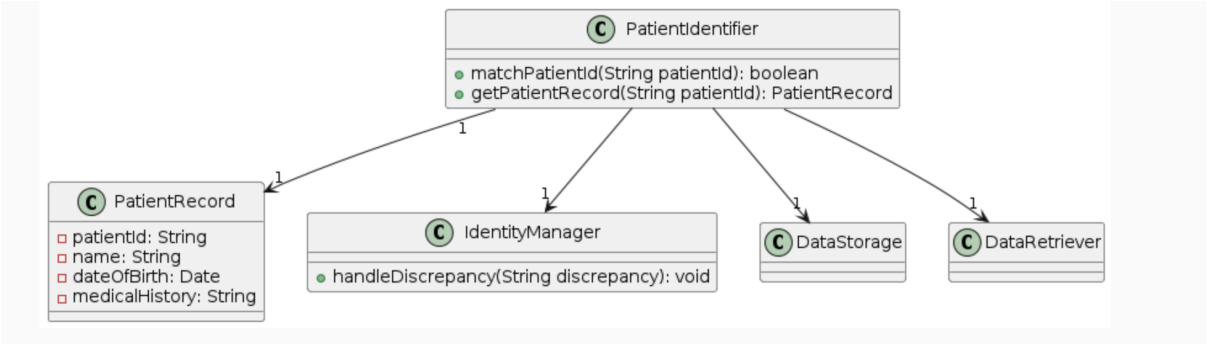
Purpose:
The Data Storage System is responsible for storing, managing, and retrieving patient data, both real-time vital signs and historical medical data, in a secure and encrypted manner.

Functionality of Each Class:

- PatientData: Same as described in the Alert Generation System.
- Metrics: Same as described in the Alert Generation System.
- BloodPressure: Same as described in the Alert Generation System.
- DataStorage: Stores patient data and provides operations for storing, retrieving, and deleting patient data.
    - storeData(PatientData data): Method to store patient data in the system.
    - retrieveData(String patientId): Method to retrieve patient data based on patient ID.
    - deleteData(String patientId): Method to delete patient data based on patient ID.
- DataRetriever: Queries and fetches patient data as needed, applying security checks to ensure authorized access.
- queryData(String query): Method to query and fetch patient data.

Interactions:
DataStorage interacts with PatientData to store and retrieve patient data.
DataStorage interacts with DataRetriever to provide querying and retrieval capabilities.
DataRetriever ensures that only authorized personnel have access to sensitive patient information.

Patient Identification System



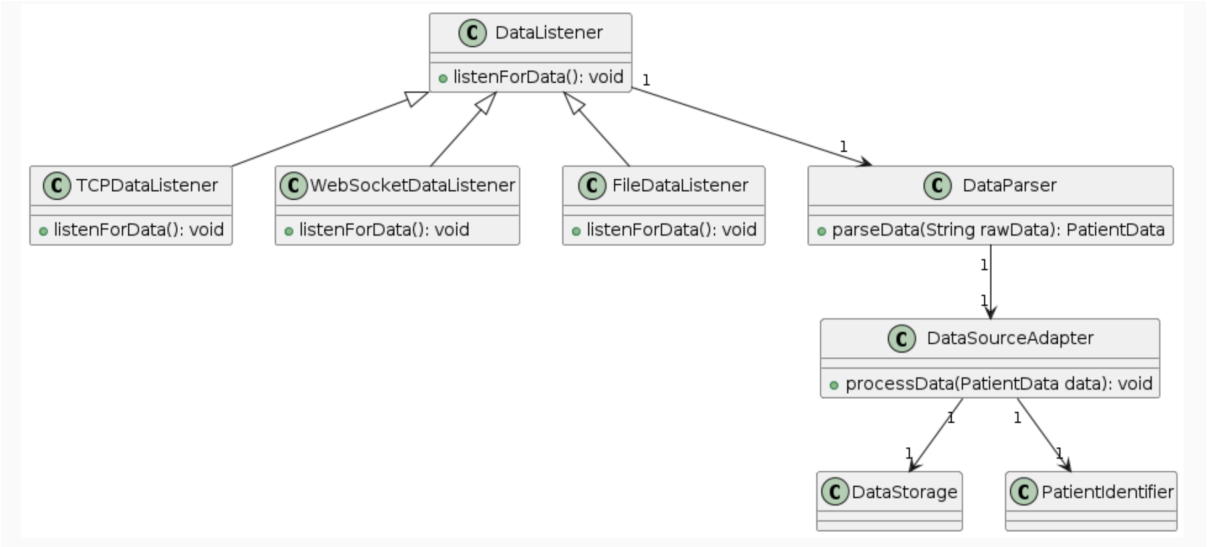
Purpose:
The Patient Identification System ensures that incoming patient data is accurately linked to the correct patient profile, preventing misidentification and ensuring the integrity of patient records.

Functionality of Each Class:

- PatientIdentifier: Matches incoming patient IDs with patient records in the database and retrieves patient records based on patient ID.
    - matchPatientId(String patientId): Method to match incoming patient IDs with patient records.
    - getPatientRecord(String patientId): Method to retrieve patient records based on patient ID.
- PatientRecord: Represents a patient's comprehensive record, including personal information and medical history.
- IdentityManager: Handles discrepancies or anomalies in data linkage, ensuring the integrity of patient records.

Interactions:
PatientIdentifier interacts with PatientRecord to match patient IDs and retrieve patient records.
PatientIdentifier interacts with DataStorage and DataRetriever to access and retrieve patient data.
IdentityManager oversees the process of matching patient IDs and handling discrepancies in data linkage.

Data Access Layer



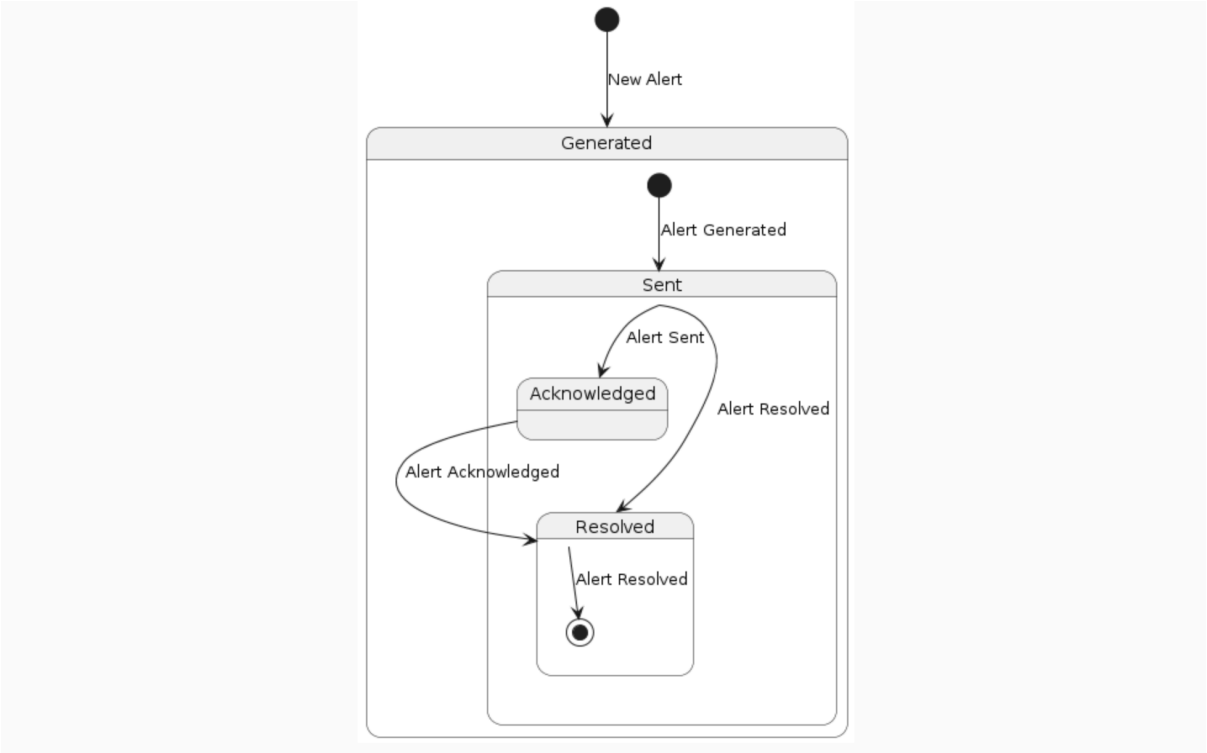
Purpose:
The Data Access Layer serves as a bridge between the Health Monitoring System and external data sources, ensuring effective communication and data retrieval.

Functionality of Each Class:

- DataListener: Listens for incoming data from different sources and forwards it for processing.
- TCPDataListener: Listens for incoming data via TCP/IP connections.
- WebSocketDataListener: Listens for incoming data via WebSocket connections.
- FileDataListener: Listens for incoming data from files.
- DataParser: Parses incoming data into a uniform format for processing.
- DataSourceAdapter: Processes parsed data and interacts with internal systems for storage and further processing.

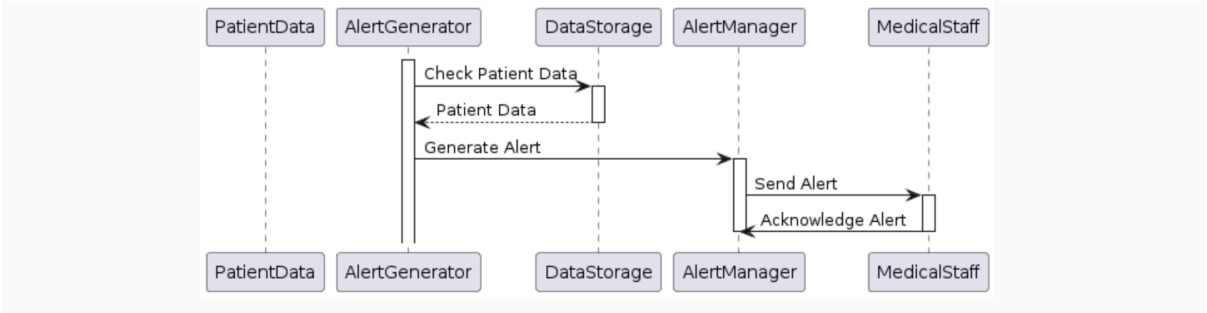
Interactions:
DataListener interacts with different types of data sources, including TCP/IP connections, WebSocket connections, and files.
DataParser standardizes incoming data into a uniform format.
DataSourceAdapter interacts with DataStorage and PatientIdentifier for storage and processing of parsed data.

UML State Diagramm for Alerts



Alerts progress from the Generated state to Sent when they are sent to medical staff, then to Acknowledged when they are acknowledged, and finally to Resolved when they are resolved, either automatically or manually

Sequence Diagramm



This uml visualizes the sequence of actions involved in the Alert Generation System