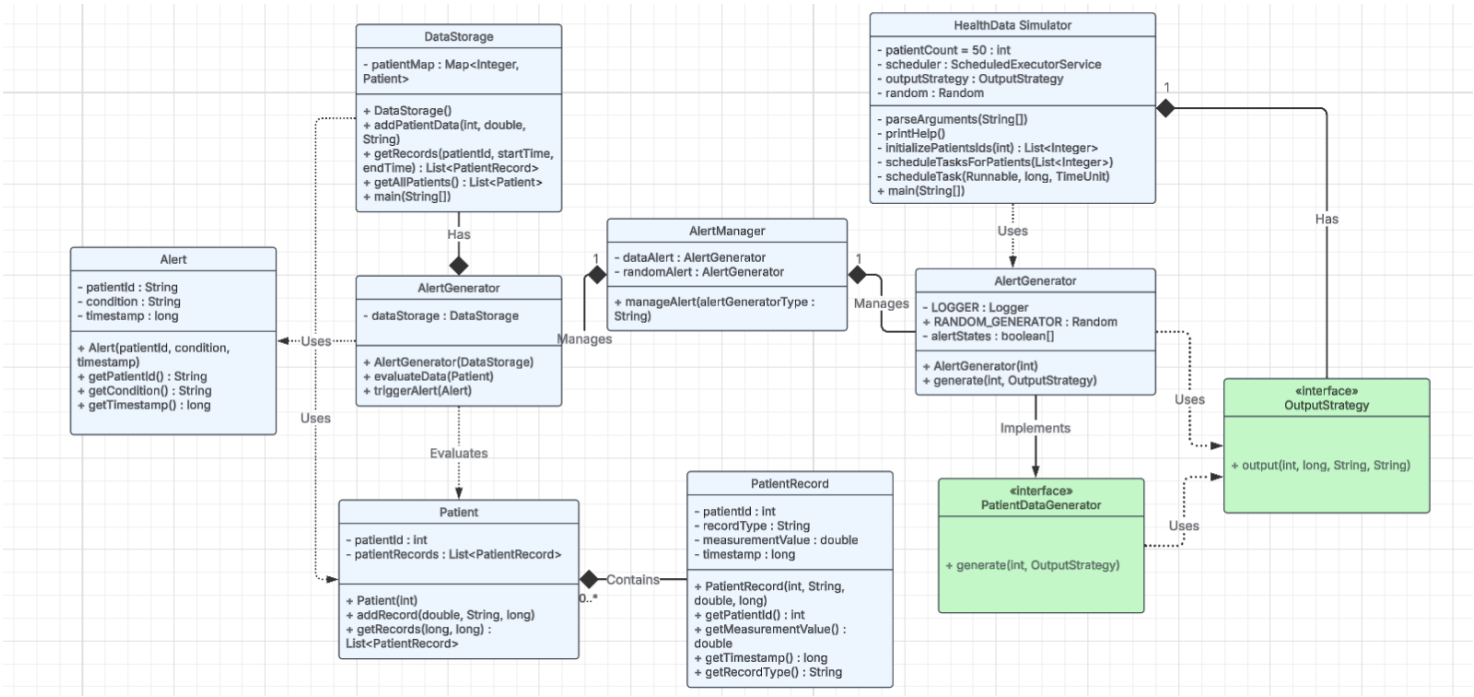
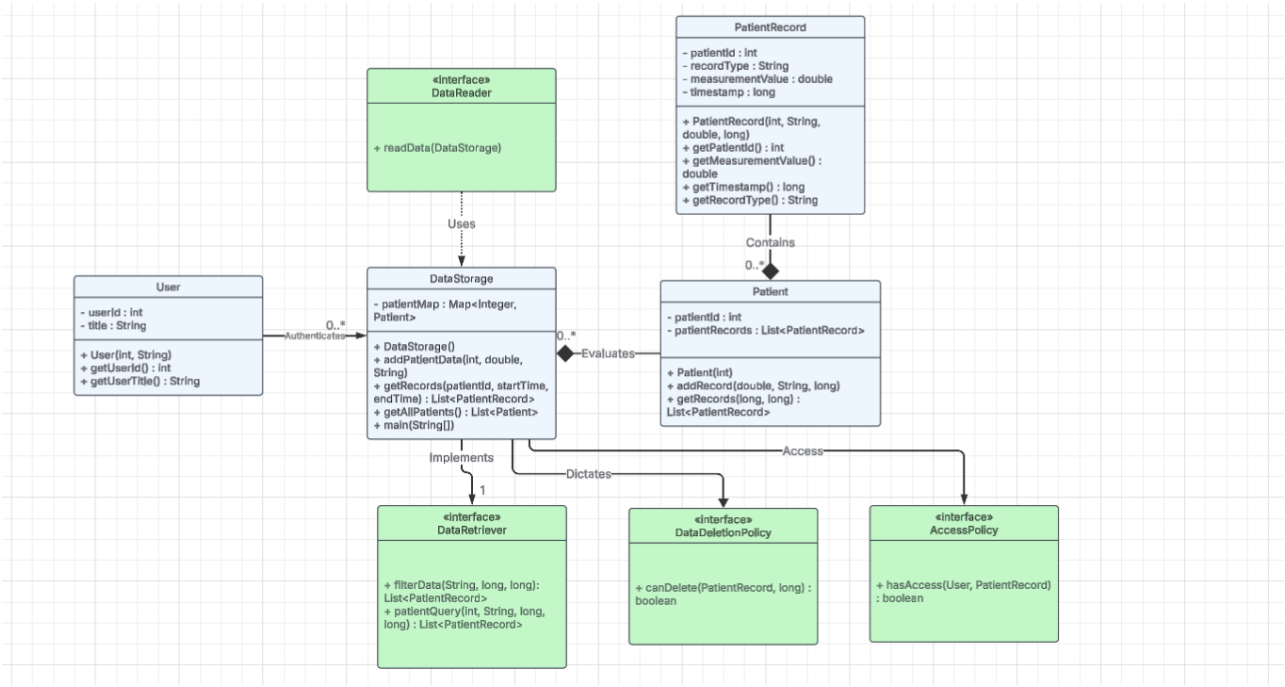
The 4 Sub-systems

1. Alert Generation System



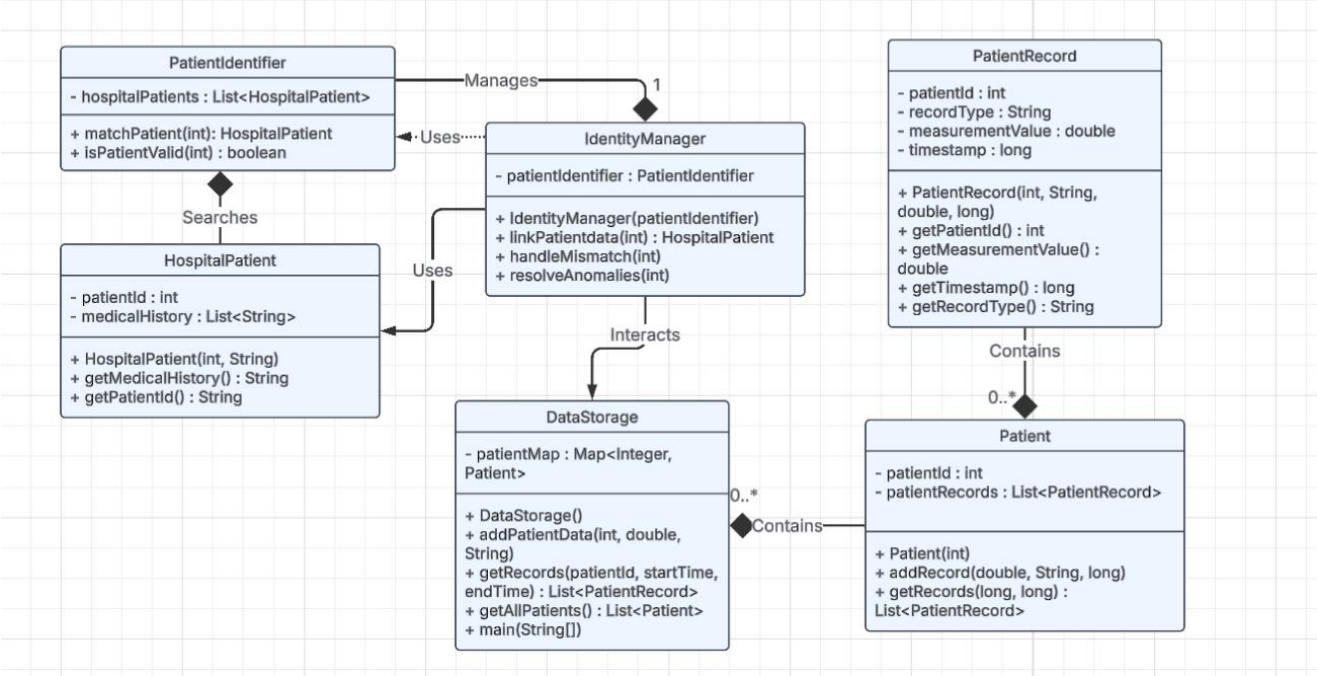
The following UML diagram illustrates the Alert Generation System comprising the flow and structure of the information passed. This diagram also depicts how patient vital signs are continuously evaluates against predefined thresholds to generate timely medical alerts. At the center of the system lies the AlertGenerator class, of which evaluates patient data retrieved from the DataStorage class holding Patient and PatientRecord data, all of which allow to model patient health. The AlertGenerator also creates an Alert object containing encapsulated patient details and condition to maintain integrity, and then passes it to the AlertManager. The AlertManager manages the distribution of these alert based on the type of AlertGenerator. The second AlertGenerator is seen to implement and use both the PatientDataGenerator and OutputStrategy interfaces to communicate alerts and allow for flexible inputs. Finally, the HealthDataSimulator class plays a significant role in orchestrating the entire process by scheduling alert generation tasks which completes the architecture that enables real-time monitoring of patient vitals.

2. Data Storage System



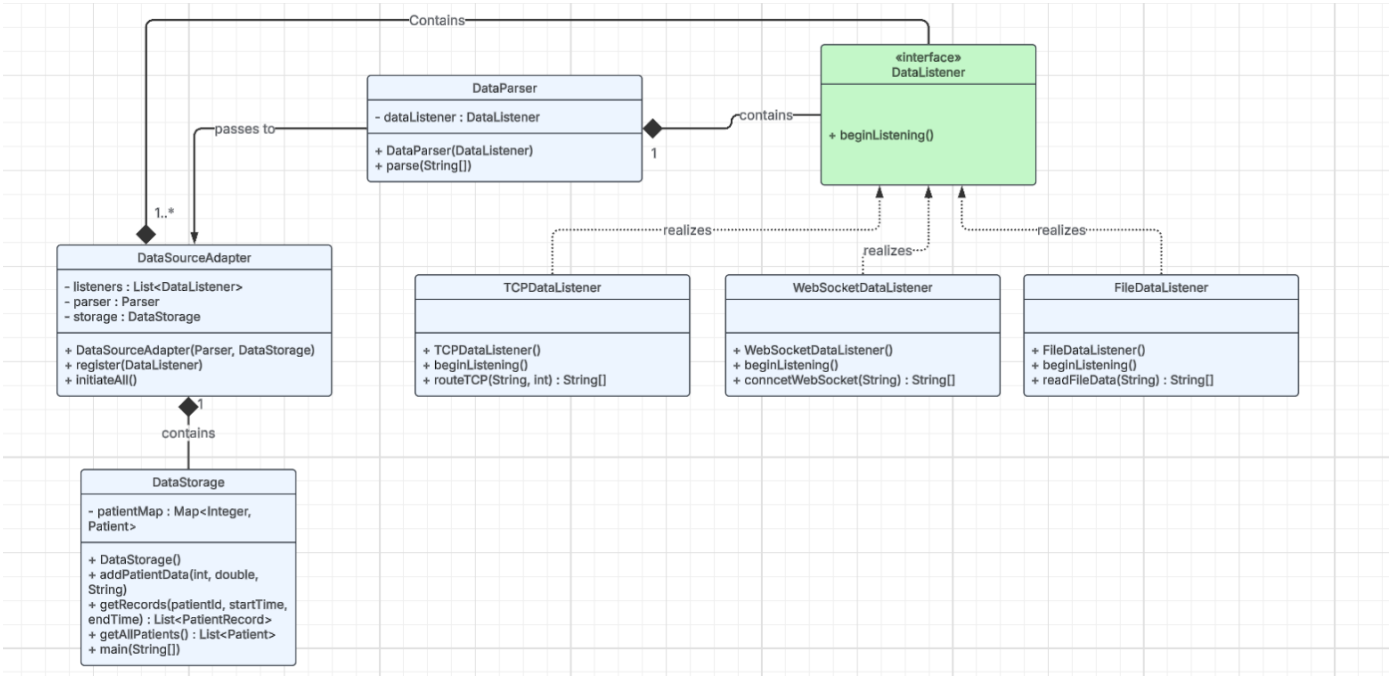
The DataStorage class lies at the core of the Data Storage System, serving as the central repository for managing patients and their associated records. It organizes patient data by mapping Patient objects to their unique IDs, with each Patient containing a collection of PatientRecord objects that store time-stamped health measurements. To ensure modularity and extensibility, the system integrates interfaces such as DataRetriever for advanced querying capabilities, DataDeletionPolicy for defining rules to remove outdated records, and AccessPolicy for enforcing security and integrity by verifying user permissions. The User class complements this by providing a template for identifying and authenticating individuals accessing the data. Together, these components create a robust, secure, and flexible system for storing, retrieving, and managing patient health data.

3. Patient Identification System



Modeling the Patient Identification System consists of a few core principles. First, the PatientIdentifier class serves as a management tool for the HospitalPatient objects (which also separates concerns from the actual Patient & PatientRecord), and it provides methods to match patients by ID and verify their validity. Then, the IdentityManager class, of which coordinates the overall identification process, handles patient data by linking and resolving anomalies if they occur. The diagram also showcases how the IdentityManager interacts with DataStorage system, which maintains the relationship between patients and their health records. Overall, definitions such as “manages”, “searches”, “contains”, etc, highlight how classes interact and maintain a robust architecture of ensuring incoming patient data is properly associated with the correct hospital records.

4. Data Access Layer



The final sub-system, Data Access Layer, serves as the crucial interface between external data sources and the monitoring system as it enables the ingestion and standardization of patient data. At the core lies the DataListener interface, responsible for defining common protocols for receiving data from distinct external sources. These protocols are namely TCPDataListener, WebSocketDataListener, and FileDataListener. The DataParser class works alongside with these protocols to transform raw incoming data into a standardized form of which the system can process. The DataSourceAdapter class acts as the orchestrator of the entire layer system, managing multiple listeners, coordinating the parsing process, and ultimately feeding the parsed patient data into the DataStorage system. Overall this structure enables for a data pipeline that can easily accommodate distinct data sources and formats while separating concerns from mutating the core system.