Data Storage System Requirements

DataStorage class is responsible for storing patient data and managing it. Regarding privacy, only authorized pesonnel should have access to sensitive patient data, which is why the instances of the class remain mostly private. The DataStorage class interacts with instances of PatientData to store, retrieve and delete patient data. It is represented by the composition relation, because the class DataStorage doesn’t exist if we don’t have any patient data. DataRetriever allows the stuff to quickly access basic information about a patient, such as their name, date of birth or medical history. It has methods to search those attributes based on the patient’s unique ID, which makes the information easily accessible. Data storage can call PatientRecord’s methods, however not vice versa. That’s because PatientRecord does not need to interact directly with the data storage mechanism, as it merely provides the necessary details for identification and retrieval purposes. By that design, we ensure that patient identification details remain separate from the storage logic, enhancing encapsulation and protecting data integrity. The HistoricalDataAPI class serves as an interface for accessing historical patient data stored in DataStorage. DataStorage and AlertGenerator are concrete implementations of HistoricalDataAPI.

Data Access Layer

The UML class diagram shows the data access later, starting with the DataListener class, which serves as a generic data listener. DataListener is responsible for actively listening for incoming data with a listen() method. It is an abstract class that provides a common base for different implementations tailored to specific data sources. WebSocketDataListener, FileDataListener and TCPDataListener are all concrete implementations of this abstract class. After receiving data from different sources, DataParser converts this raw data into a standardised format. Next, DataSourceAdapter **t**akes standardised data from DataParser and processes it, preparing it for storage in DataStorage. DataStorage stores processed data and provides methods for storing, retrieving, and deleting patient data. The relationships in the diagram are mostly dependencies, because in that way each class encapsulates its own functionality and relies on other classes to perform a specific task. For example, the DataSourceAdapter focuses on processing data, while the DataStorage is responsible for storing it. This separation enhances code readability, understandability, and maintainability.