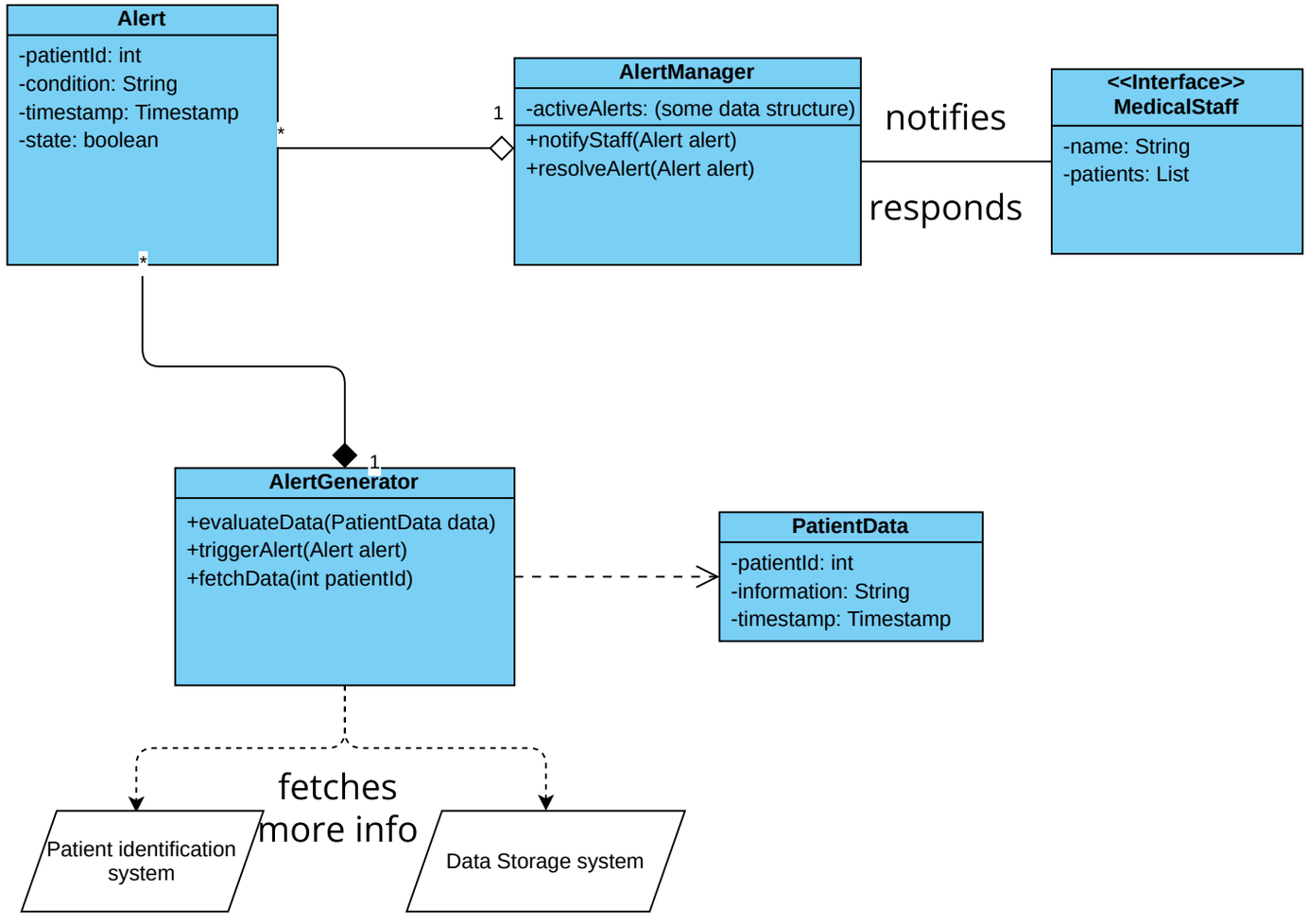
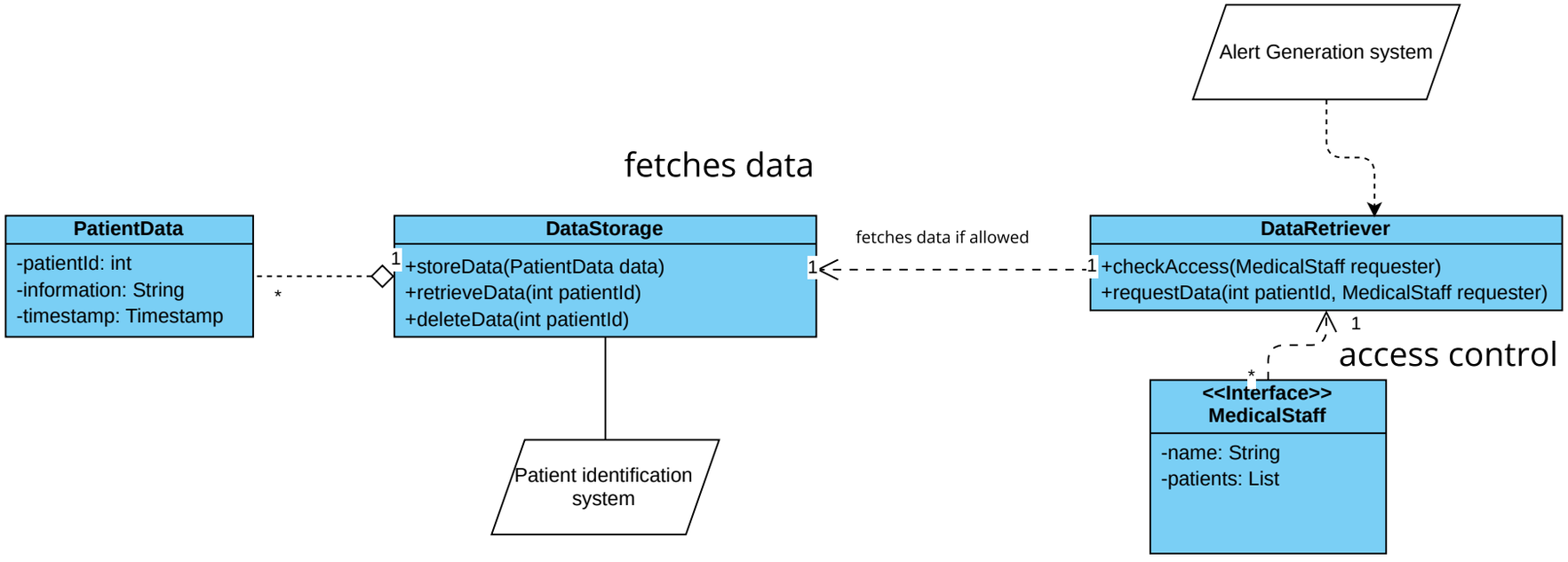
Alert Generation System



Alert Generator creates the alerts after evaluating the patients data, retrieved from Data Storage and Patient Identification systems. The active alert is then sent to the Alert manager, which holds all active alerts and notifies the appropriate medical staff until they respond to the alert.
Alert Generator needs the information from the other systems to make detailed judgement wether to trigger an alert or not, both the alert conditions set by the patient's doctors and to identify additional anomalies, therefore the relationship is dependant. I brought out Patient Data sepeartely, as it is crucial for the generator, it would serve no purpose without the data, so it is very much dependent. Alert Generator, as the name suggests, is also the one to generate alerts, which means it is responsible for the lifecycle of these alerts.
The purpose of Alert Manager is to take care of active alerts, but (inactive) alerts can exist outside it, so aggregation is an appropriate relationship. Alert Manager has a two-way association with MedicalStaff, as it

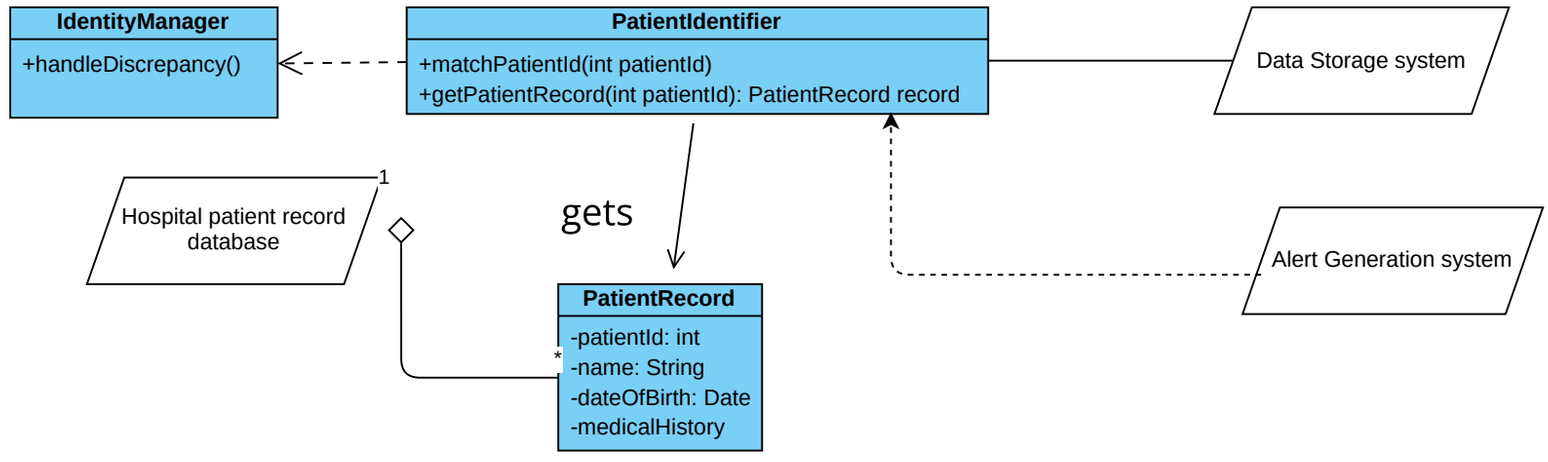
sends the alerts to the personell and they, in turn, can respond to those alerts.
Here and after, MedicalStaff is an interface, upon which classes for nurses, doctors, admir
more would be built.

Data Storage system



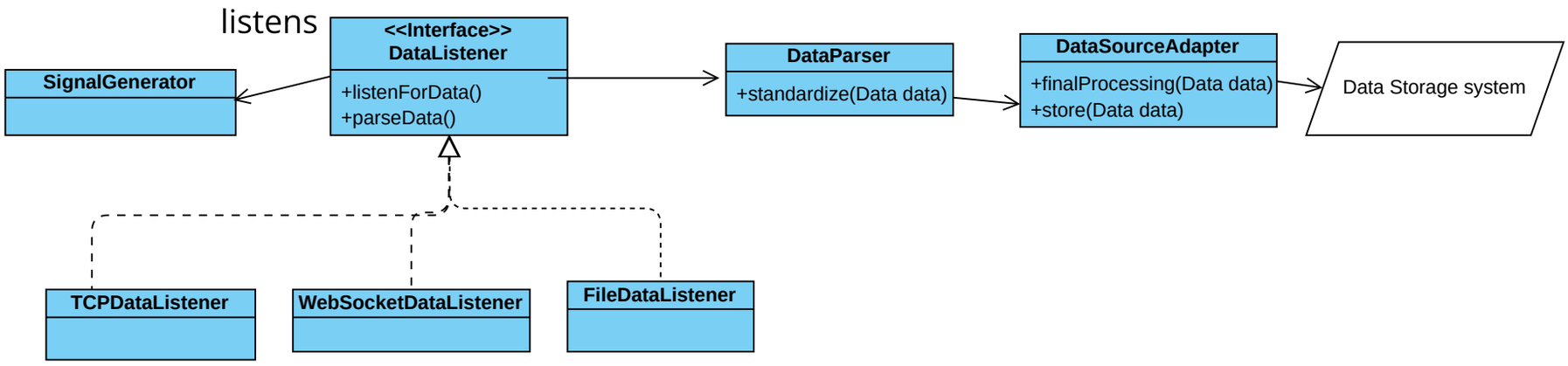
Data Storage is where the information about all patients is placed, but it is not created there, so the appropriate relationship is aggregation.
Alert Generation system uses the information in Data Storage in its decision-making processes, so alerts are dependent on data, but not the other way around.
Before retrieving data, Data Retriever needs to check whether the requester is authorized to access the stored data. Medical staff can only get the data if they pass Data Retriever's checks, so they're dependent on the Data Retriever. Data Retriever, in turn, is dependent on Data Storage, as it's main function is to get the information from this class.
The Alert Generation system is also using Data Retriever to fetch data, since that is the intended purpose of this class.
The data storage and patient identification system are associated, as together they contain the medical records with all the data, interacting constantly to maintain proper order.

Patient Identification system



The Alert Generation system uses the getPatientRecord method to fetch additional info for its processes, but Patient Identifier doesn't use the alert generator, thus the dependency arrow.
In case of anomalies, Patient Identifier uses the Identity Manager to handle the situation, thus another dependency.
The Data Storage system and Patient Identifier are associated, as they work together to store and get the record and update it.
All the patient records are hosted, but not created in the database, so association it is.

Data Access layer



The various Data Listeners implement the DataListener interface for uniform usage from various sources, so I used realization arrows.
Data listeners are made to retrieve data from its generator, which I show in my diagram.
How data gets passed along and proccesed from the listener to parser to adapter to the storage system can also be seen.