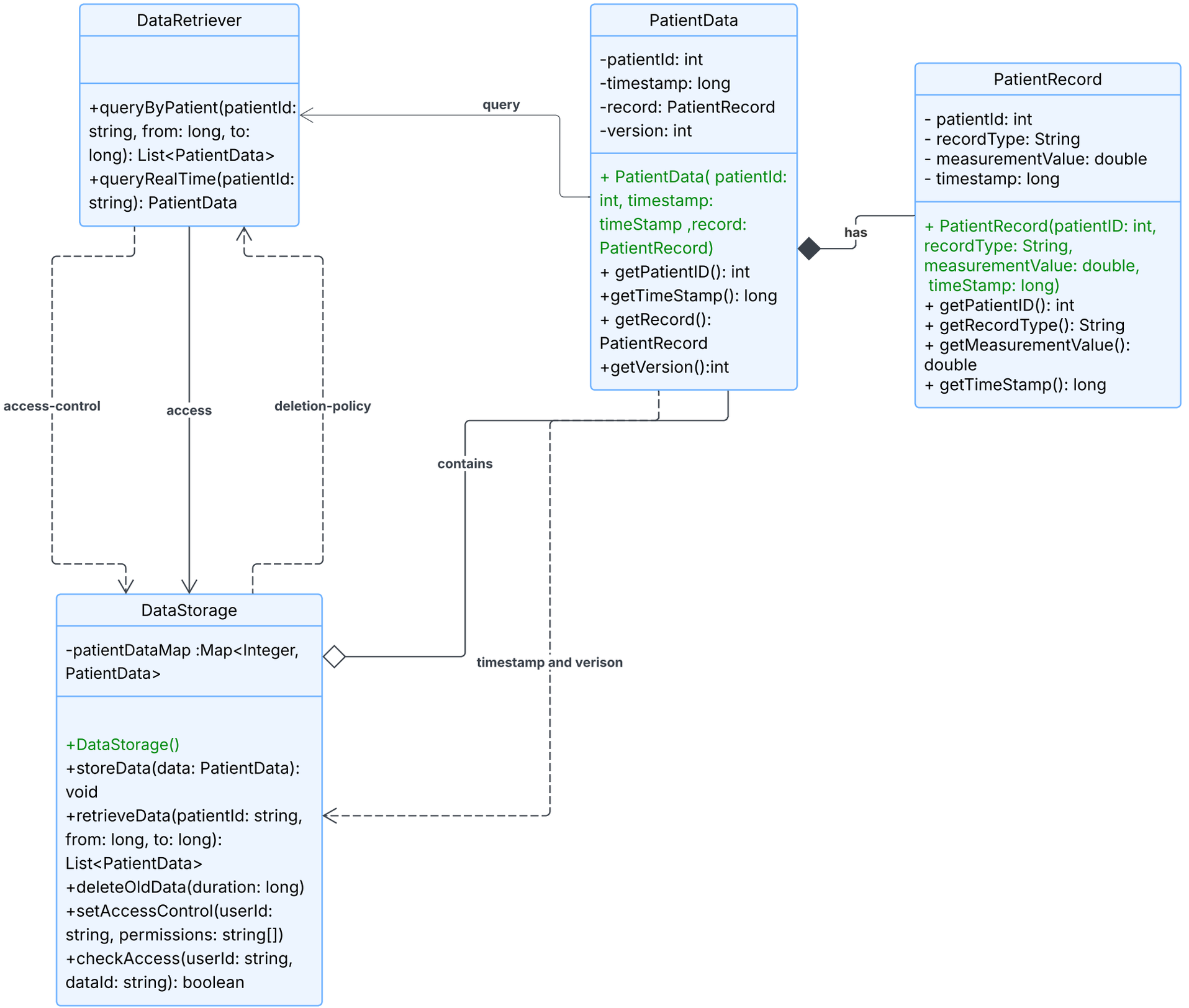
DataRetriever
+queryByPatient(patientId: string, from: long, to: long): List<PatientData>
+queryRealTime(patientId: string): PatientData
query
PatientData
-patientId: int
-timestamp: long
-record: PatientRecord
-version: int
+ PatientData(patientId: int, timestamp: timeStamp ,record: PatientRecord)
+ getPatientID(): int
+getTimeStamp(): long
+ getRecord(): PatientRecord
+getVersion():int
has
PatientRecord
- patientId: int
- recordType: String
- measurementValue: double
- timestamp: long
+ PatientRecord(patientID: int, recordType: String, measurementValue: double, timeStamp: long)
+ getPatientID(): int
+ getRecordType(): String
+ getMeasurementValue(): double
+ getTimeStamp(): long
access-control
access
deletion-policy
contains
DataStorage
-patientDataMap :Map<Integer, PatientData>
+DataStorage()
+storeData(data: PatientData): void
+retrieveData(patientId: string, from: long, to: long): List<PatientData>
+deleteOldData(duration: long)
+setAccessControl(userId: string, permissions: string[])
+checkAccess(userId: string, dataId: string): boolean
timestamp and verison

2.Data Storage subsytem:

Here is a UML diagram of a system built for managing and accessing patient health data in a reliable and secure way.

The DataStorage class is the main part of this system. It stores and manage the health-related data through a map that links patient IDs to their data. It can store new data, retrieve it based on time ranges, delete older entries ,settup access controls for different users and check autorisation to access the data so that only the right people can access certain info.

Each record that goes into the system is wrapped up in a PatientData object. It has a patient ID, a timestamp, the actual health data (in a PatientRecord), and a version number to keep track of changes in condition.

The PatientRecord class holds the actual health info: things like the type of measurement (e.g., heart rate) and when it was recorded.

Lastly, there's DataRetriever, a class that helps pull data out of the system. It allows two type of queries, over a time range and real-time data requests, which is useful for live monitoring and dashboards.

This subsystem will allow secured access to the patient data for the staff as well as for the patients.