DeletionPolicy

- numOfDays: int

+deletionPolicy(numOfDays): int
+toDelete(PatientData): boolean

DataStorage

- patientMap Map<Integer, Patient> ;

+ addPatientData(patientId:int, measurementValue:double, recordType:String , timestamp:long): void

+ getRecords(patientId:int, startTime:long, endTime:long):List<PatientRecord>

+ getAllPatients():List<Patient>

DataRetriever

- dataStorage: DataStorage

+ dataRetriever(DataStorage)
+ getPatientHistory(String userId, String patientId) : List<PatientData>

AccessController

- accessInfo: Map<String, List<String>>

+ hasAccess(String userId, String patientId) : boolean
+ allowAccess(String userId, String patientId) : void

updates — 1

retrieve — 1

send request

uses 0..*

PatientData

-patientId: int
-patientdata: List<PatientRecord>

+addPatientData(patientId: int):

uses — 0..*

PatientRecord

-patientId: int
-recordType: String
-measurementValue: double
-timestamp: long

+**PatientRecord**(patientId: int, measurementValue: double, recordType: String, timestamp: long)
+**getPatientID**(): int
+**getMeasurementValue**(): double
+**getTimestamp**(): long
+**getRecordType**(): String

The following UML-diagram depicts a secure data storage system of patients' healthcare information (patients records) that has deletion policy and access controller to check users who want to retrieve data.

In PatientData class each patient has personal ID and related medical records with health measurement values, information about timestamps and up-to-date notes about the current condition. Patient records store all the updates about health of a certain patient(such indicators as heart rate and diagnoses from specialized doctors).

Data storage is the main component of this system as it accumulates patient data that is mapped with unique IDs for all patients in a hospital and securely stores it until retrieval request by users that are allowed to access this data.

Deletion policy is provided by DeletionPolicy class that gets rid of outdated PatientData every X days, number of days after which data should be deleted can be defined by user in the private field.

To retrieve the data user should first ask for access to the storage with medical histories. AccessController uses private field accessInfo to check whether the information can be granted to the user or not and return has Access value that allows (or not) access. This confirmation or rejection is used in DataRetriever class to answer user's request and return patients' medical history from the storage.