Patient Identification System

PatientIdentifier

- patientId: int
- patientRecords: List<PatientRecords>
- recordType: String
- mesurementValue: double

+ comparData(Records: List<PatientRecords>, startTime: long, endTime: long): Boolean
+ log(Records: List<PatientRecords>, startTime: long, endTime: long): List<String>
+ matchData(Records: List<PatientRecords>, startTime: long, endTime: long): List<PatientRecords>

PatientRecord

- patientId: int
- patientRecords: List<PatientRecords>
- recordType: String
- mesurementValue: double

+ addRecord(measurementValue: double, recordType: String, timeStamp, long): void
+ getRecords(startTime: long, endTime: long): List<PatientRecords>

IdentityManager

- patientId: int
- patientRecords: List<PatientRecords>
- recordType: String
- mesurementValue: double

+ unmatchData(Records: List<PatientRecords>, startTime: long, endTime: long): Boolean
+ log(Records: List<PatientRecords>, startTime: long, endTime: long): List<String>
+ matchData(Records: List<PatientRecords>, startTime: long, endTime: long): List<PatientRecords>

This UML describes how the Patient Identification System works. The main objective of

this system is to match the generated patient ID to real hospital records. And it ensures that the incoming data are linked to the right patient.

The system contains 3 classes, PatientIdentifier, PatientRecord and IdentityManager.

PatientIdentifier is the class responsible for checking the patient, to check is id number and to linked this id to the correct stream of data. It will compare the data from PatientRecord and see if the matching is correctly done.

PatientRecords will contain the data relative to one patient. It will be use to get the data that PatientIdentifier needed to compare and check if the matching is correctly done. PatientRecord will contain the data from precise timestamp or continuous stream of data and it will add this data to the database.

IdentityManager will handle every comparison. It will also handle every problem like unmatching data, wrong attribution or inexistent data. It will send to PatientIdentifier if everything is correct and if it's not, it will correct it and notify PatientIdentifier of the errors.

This UML show how the verification of the data is done and how effective the system is to handle different scenario.