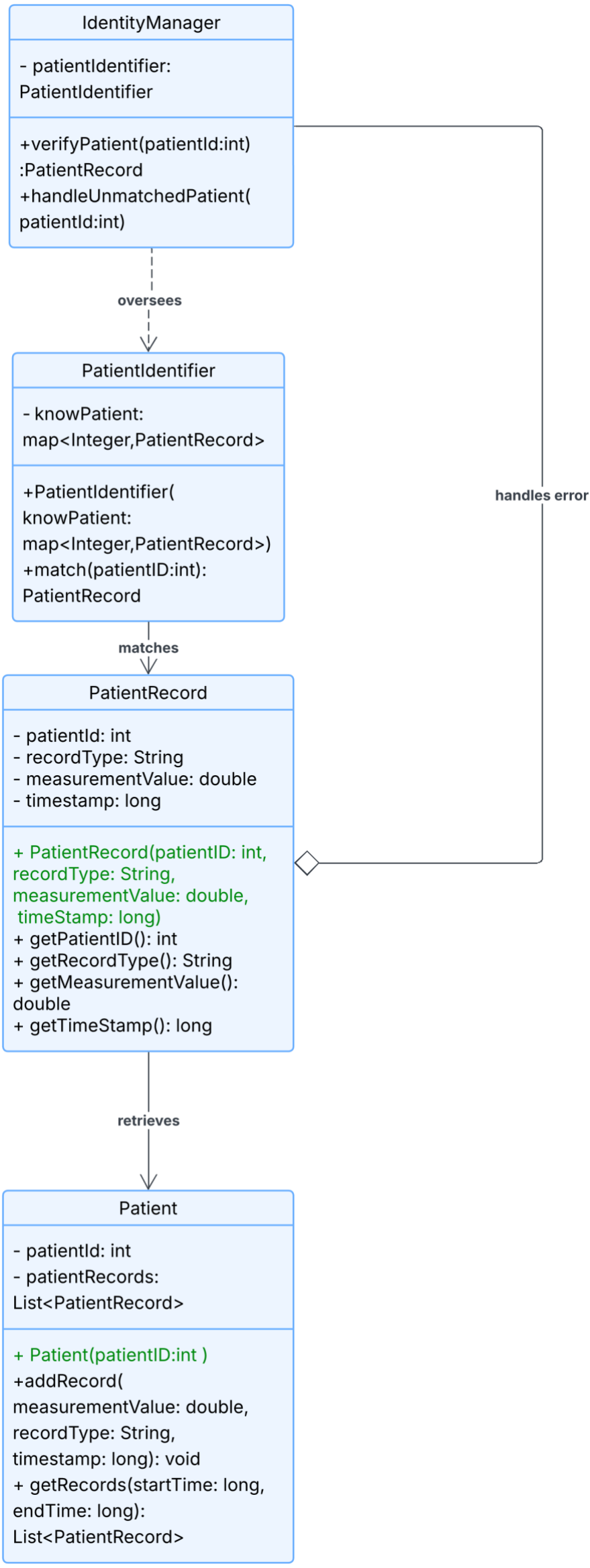
IdentityManager
- patientIdentifier: PatientIdentifier
+verifyPatient(patientId:int) :PatientRecord
+handleUnmatchedPatient(patientId:int)
oversees
PatientIdentifier
- knowPatient: map<Integer,PatientRecord>
+PatientIdentifier(knowPatient: map<Integer,PatientRecord>)
+match(patientID:int): PatientRecord
handles error
matches
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
retrieves
Patient
- patientId: int
- patientRecords: List<PatientRecord>
+ Patient(patientID:int)
+addRecord(measurementValue: double, recordType: String, timestamp: long): void
+ getRecords(startTime: long, endTime: long): List<PatientRecord>

3. patient identification subsystem:

this UML setup describe how patient data is managed, identified, and retrieved. the idea is that every patient has a unique identity, and we want to be able to match them with their medical records efficiently and accurately.

The PatientIdentifier class acts as a directory .It keeps track of all known patients using a map where the key is an integer ID and the value is the actual PatientRecord. This way, when someone inputs a patient ID, the system can try to find a match using the match() method.

The IdentityManager is the supervisor. It uses the PatientIdentifier to verify if a patient exists, and if it can't find a match, it calls handleUnmatchedPatient() to deal with errors. Therefore it ensure that you don't try to access data for a non-existent patient.

The Patient class contains patient's data. Each Patient has a list of PatientRecords (for each record there are blood pressure readings, lab test results, etc.). You can add new records and also query them based on a time range. It allow to track trends or viewing patient history over a certain period.

Lastly, the PatientRecord class is just the individual record details.it stores what the measurement was, when it was taken, what type of data it is, etc.

This subsystem allows to access the patient data while being sure not to cause any errors.