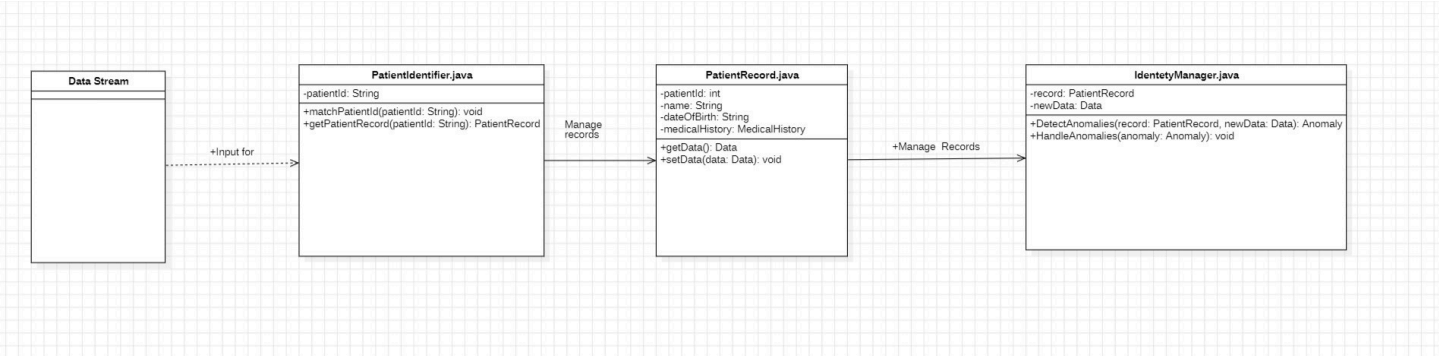

The UML diagram presents a system that processes and manages patient records. The process starts with the 'Data Stream', which acts as an input source for the 'PatientIdentifier.java' class. The 'PatientIdentifier' class holds a 'patientId' of type String and provides methods to match a patient ID with incoming data and to retrieve a patient record.

The 'PatientRecord.java' class manages the patient's information, including an integer 'patientId', the patient's 'name', 'dateOfBirth', and 'medicalHistory', which is of type 'MedicalHistory'. It offers methods to get and set data, suggesting that this class can both retrieve and update patient information.

Finally, there's the 'IdentityManager.java' class (despite the misspelling in the class name as 'IdentetyManager'). It is responsible for detecting and handling anomalies between existing patient records and new data. It has access to the patient records and contains methods to detect anomalies in the data and handle them accordingly. This indicates a layer of the system designed for data integrity and consistency checks.

The 'Manage records' label that connects 'PatientRecord' and 'IdentityManager' implies that 'IdentityManager' performs its operations by managing instances of 'PatientRecord'.