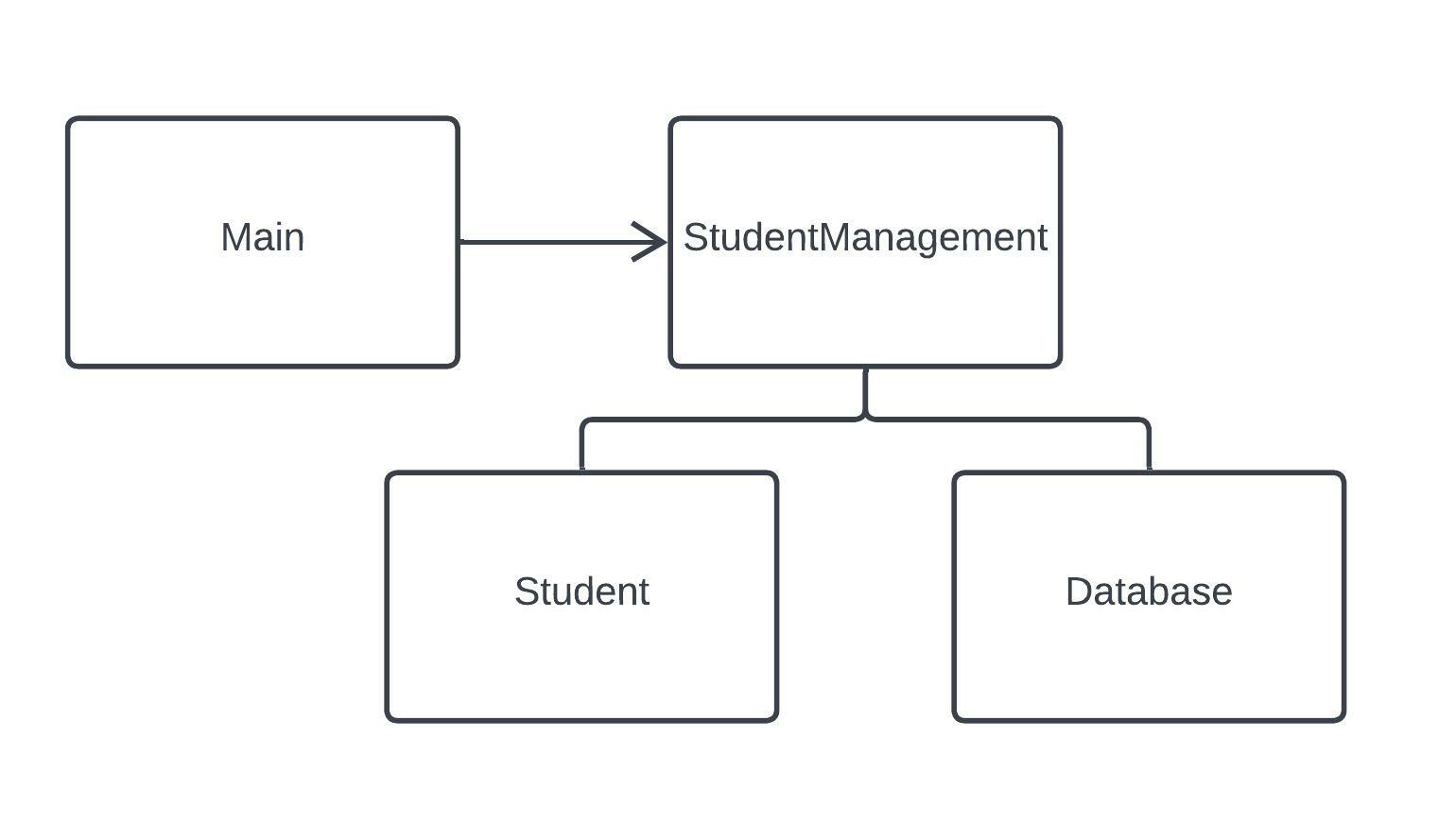
**Educational System**

**Propose:**

This system is for manipulating student personal data in an education institute. The application provides a consistent database of student details. Also, it can handle the registration process, data updating functionality, student delete functionality, and all student data printing functionality.

**Structure:**

This system has four classes and one data file. Classes are arranged as follows. Database class access data file (Student.xml). StudentManagement class is the data manipulating class. It communicates with all other classes. Main class and Database class, only communicate with StudentManagement class. When the application start, read all data from the XML file and keep in Database instance. All data updates are happened to the Database instance class. When application is exiting, all data of the Database instance, will be save to the XML file.

**Student Class**

This class represents the Student entity of the application. This class helps to make student data handling proceed easier.

* Attributes
  + registrationNumber
  + firstName
  + lastName
  + age
* Functions
  + Getters of all attributes

**Database Class**

This class is used for keeping data in the application. This is a singleton class.

* Attributes
  + studentHashMap
  + FILE\_PATH: Path of the XML file
* Functions
  + Read student from the XML file
  + Keep student data in a Map
  + Add a student to the Map
  + Delete student from the Map
  + Save Map data to the XML file

**StudentManagement Class**

This is the class that handles all functions of the Education Institute. This class is responsible for manipulating all data of the application.

* Functions
  + Provide all student data to the user
  + Handle student registration
  + Handle student data modifying process
  + Delete students
  + System exit function

**Main class**

The main class handles the runtime facilities of the application.

* Functions
  + Provide function Manu to the user
  + Take user inputs
  + Provide data according to user requests
  + Keep application in run state
  + Communicate with StudentManagement class

**XML File**

Keep student data of the application.

* <EducationInstitute>: Root element of the XML file
* <Student>: Parent tag of every student entity
* <registrationNumber>: Keep registration number of the student.
* <firstname>: Keep first name of the student.
* <lastname>: Keep last name of the student
* <age>: Keep age of the student