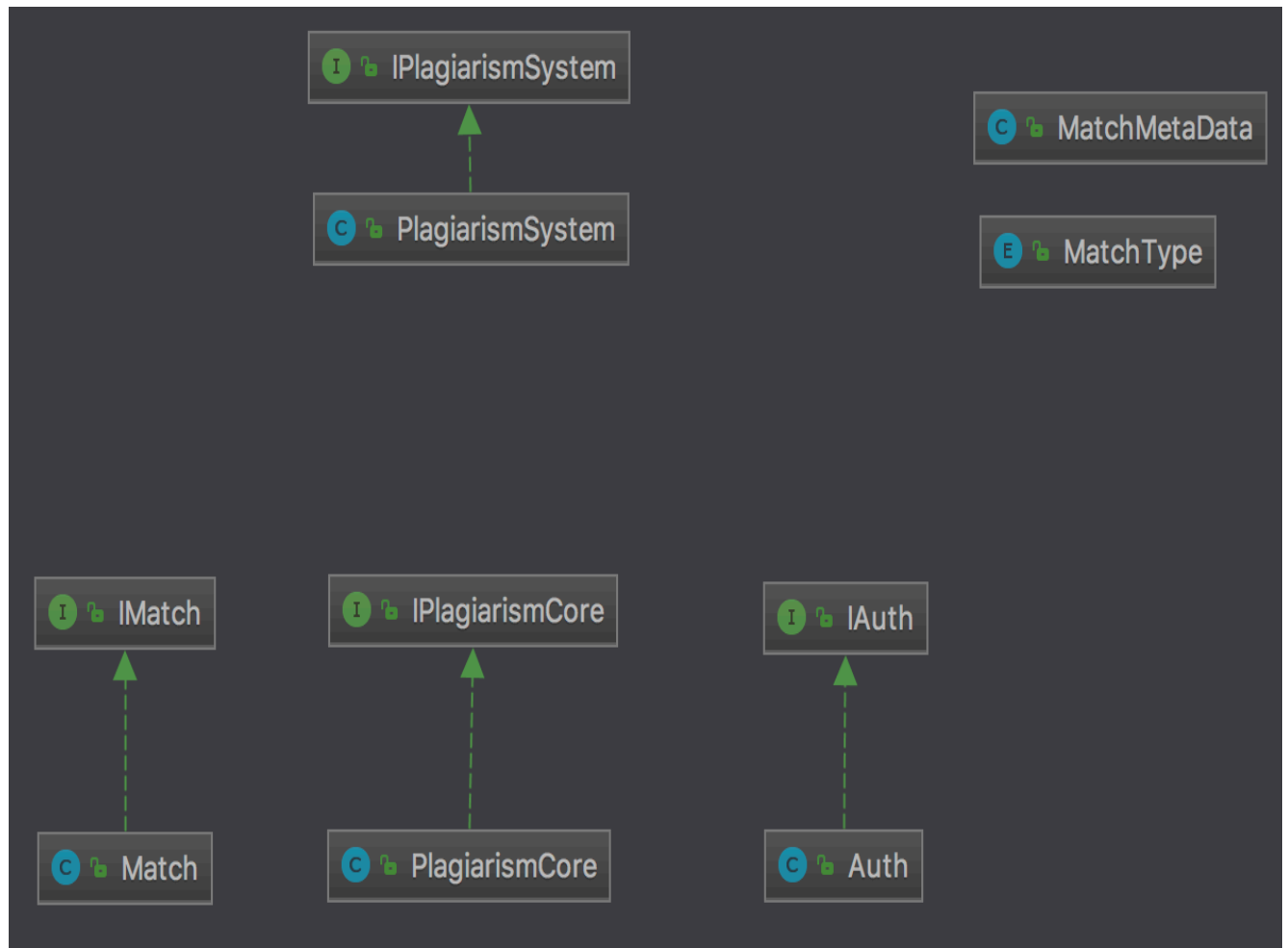


Phase B: Team 104 UML Diagrams & Data structure

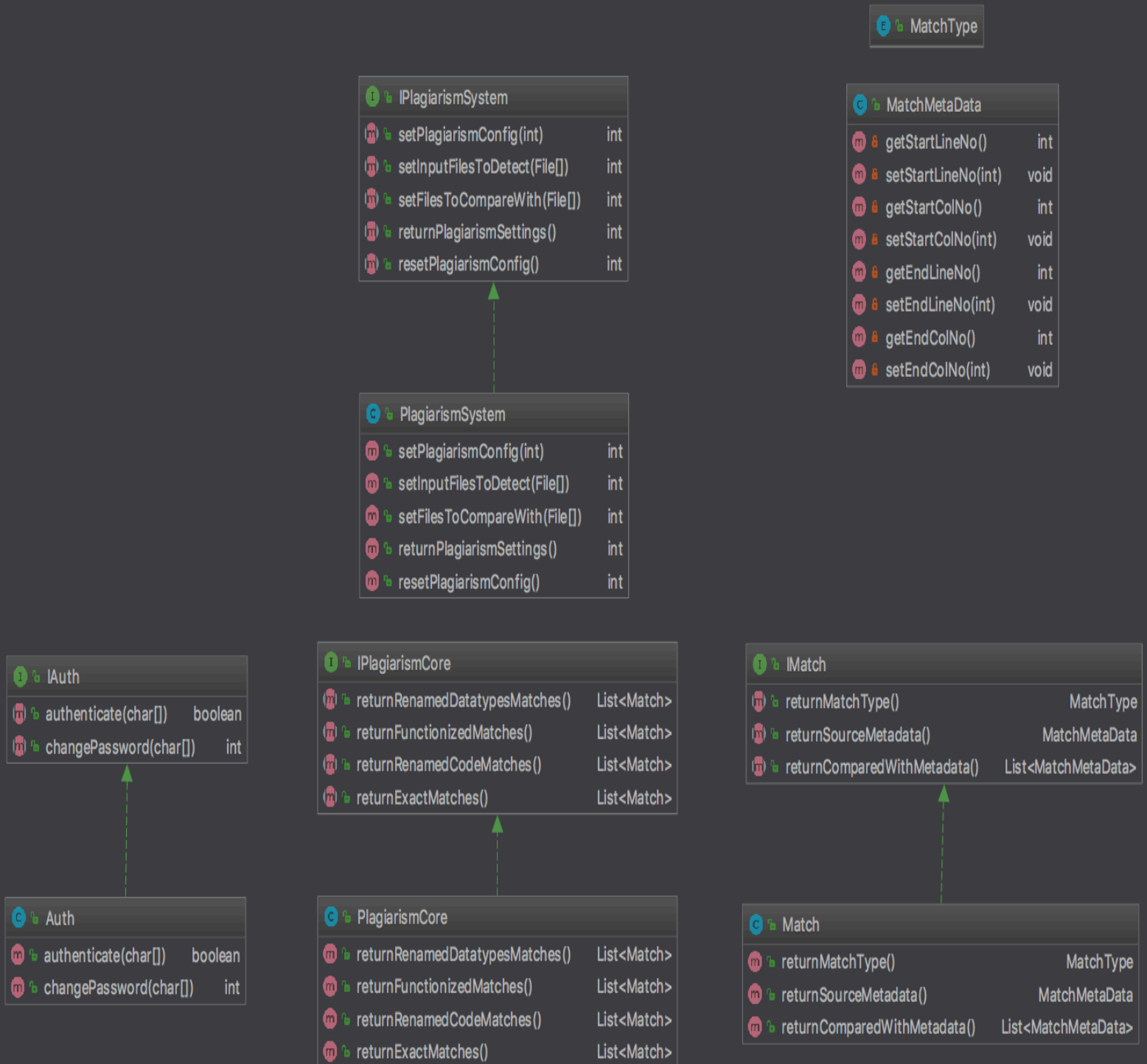


The plagiarism system class is similar to a centralized hub that connects to all other modules and interfaces.

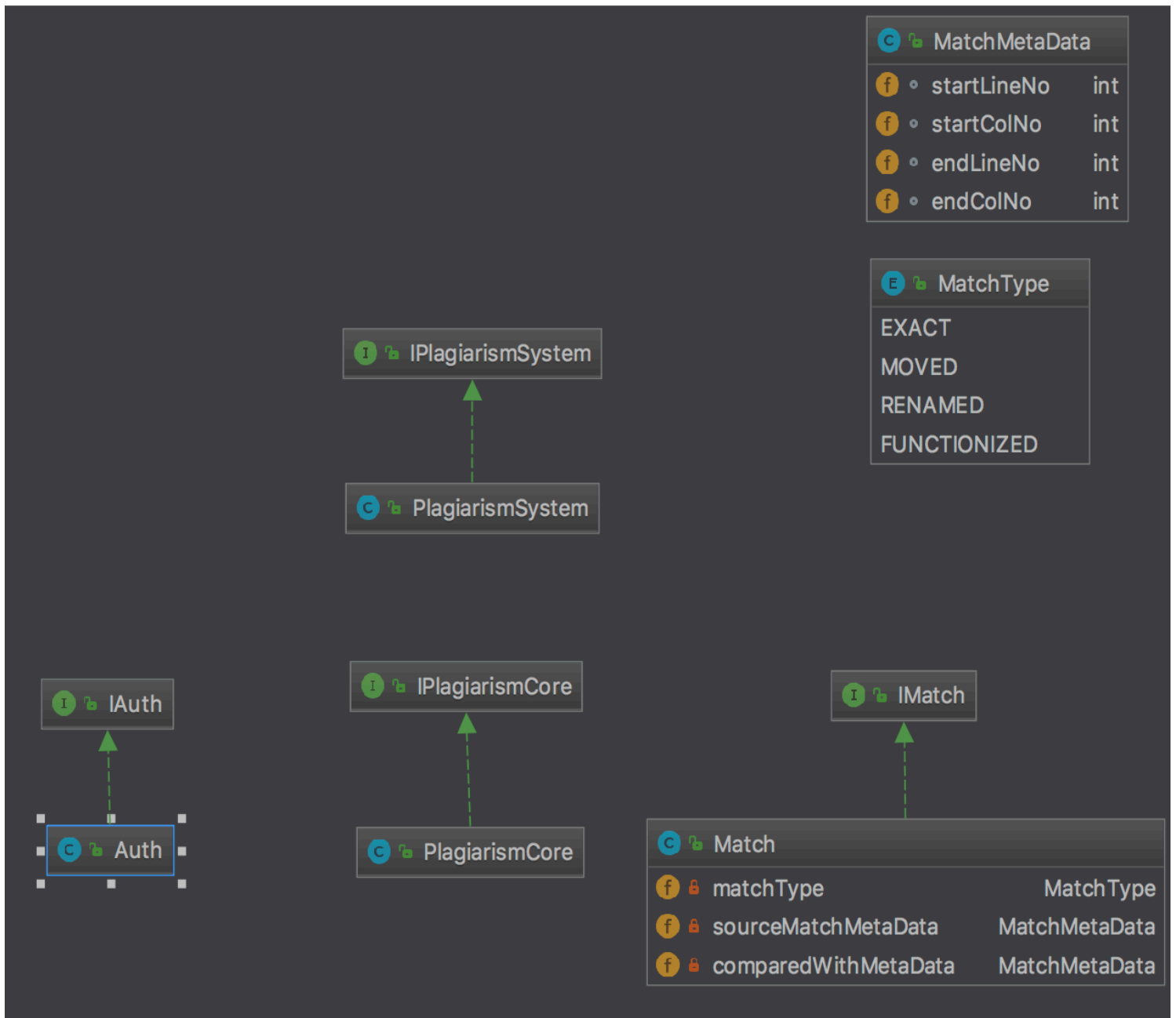
For each interface like **IPlagiarismCore**, **IAuth** etc. a singleton object is created and then the appropriate methods are called inside the methods to be implemented.

Design pattern to be used : Singleton

Detailed UML Design (reflecting basic methods)



UML with User defined data structures & fields



In the above picture, each (f) represents a field;
The corresponding type can be found next to it.

Major Data Structures Used:

User Defined Data Structure:

Public class MatchMetadata

- This is one of the key types in this architecture.
- Holds data for matches
- Each match can be defined as a plagiarism occurrence.
- Based on the requirement given, plagiarism occurrences are categorized into types as defined in the enumerated type Match Type

- **MatchType matchType;**

Can be

- EXACT
- MOVED
- RENAMED
- FUNCTIONIZED

- **MatchMetaData sourceMatchMetaData;**

- startLineNo – holds the line number of the occurrence beginning in the file which is being compared
- endLineNo – holds the line number of the occurrence ending in the file which is being compared
- startColNo - holds the Column number of the occurrence beginning in the file which is being compared
- endColNo – holds the Column number of the occurrence ending in the file which is being compared

- **MatchMetaData comparedWithMetaData;**

- startLineNo – holds the line number of the occurrence beginning in the file against which the source is being compared
- endLineNo – holds the line number of the occurrence ending in the file against which the source is being compared
- startColNo - holds the Column number of the occurrence beginning in the file against which the source is being compared
- endColNo – holds the Column number of the occurrence ending in the file against which the source is being compared

Other Built in Data Structures Used :

1. List<Interface> - for storing a list of matches
2. Map - for storing key-values in case of interaction with UI
3. JSONObject – for receiving and sending data to UI elements
4. Char[] for storing hashed passwords

And many other commonly used types such as String, int, etc.

How it works?

- The plagiarism detection is based on a confidence level, a cumulative aggregate of various other parameters present inside the system.
- The plagiarism system has a configuration stored, which can be changed if the user successfully enters the password. Else, the system can be used without the ability to change the configuration.
- Then the files are added
 - To compare or source – the files for which the detection is performed
 - comparedWith – the files against which the system runs a check.
- Once the files are fed into the system, the plagiarism matches are found and stored categorically. The UI will be displaying the data in the chosen interface based on the data available.

Note:

- (i) This architecture focuses on the sections of storing and categorizing the data but not on the UI.
- (ii) All the modules described above constitutes the basic skeleton of the system. It has been kept as simple as possible to ease the understanding for
- (iii) The support libraries such as ANTL4 and other language processing tools will be utilized in order to enhance and simplify the development process
- (iv) All the core features of the plagiarism detection will be developed in Java according to the requirement.