Social network

This homework must be filed as a jar archive BEFORE April 2, 2029 on the ENT, indicating your full names. This archive must contain :

— java source codes
— installation documentation
— user documentation

The homework is to be done in groups of 2 or 3 people. You can take the initiative to complete or modify what is requested below.

The objective consists in creating a java application that analyses data from a social network. Data of the social network we consider are of two types : message and comment. A *message* initiates a thread (of discussion), a *comment* comments a message or a comment.

— A *message* is defined with the following items :
  — date : date of the message
  — idMessage : is the (unique) identifier of the message (integer)
  — idUser : is the (unique) identifier of the user (integer)
  — message : is the content of the message (string)
  — user : is the name of the user (string)
— A *comment* is defined with the following items :
  — date : date of the comment
  — idCommentaire : is the (unique) identifier of the comment (integer)
  — idUser : is the (unique) identifier of the user (integer)
  — comment : is the content of the comment (string)
  — user : is the name of the user (string)
  — pidCommentaire : is the identifier of the comment this comment comments (-1 if this comment comments a message)
  — pidMessage : is the identifier of the message this comment comments (-1 if this comment comments a comment)

The aim is to create a server that computes continuously the 3 messages that have the best **value of importance**. The value of importance is an integer that characterizes the importance (or the relevance, the accuracy, ...) of a thread of discussion (each new message initiates a new thread of discussion). The higher the value, the more important the message is. The value of importance of a message is computed as the sum of its score and the scores of its associated comments. A comment is associated to a message if this comment is a comment to the message or to a comment associated to this message.

A score is a positive or null integer. Each message or comment has an initial score of 20, the score decreases by 1 each 30s. If the total score of a message is 0 then this message is considered inactive, even if other comments are then associated to it. Only active messages can be among the 3 best.

The server is supposed to be connected to a social network and receive messages and comments from it. This will be simulated : the server reads data from a file.

1. Define classes `Comment` and `Message`.
2. Define a server that should perform the following operations :
   (a) A (java) thread reads from the file `reseauSocial.txt` messages and comments and computes continuously the 3 best messages (the 3 messages that have the best value of importance). The reading of a line of the file (i.e., a message or a comment) is done after a random time of between 1 and 3s.
   (b) A pool of (java) threads will be used for answering clients. If a client connects to the server, the server sends the client the 3 messages that have the best value of importance (at the time the client connects) in the following pattern :

```
idMessage|idUser|idMessage|idUser|idMessage|idUser
```

3. The server should be defined such that :
   (a) The java thread that reads data and computes the best messages should be on a virtal machine distinct from the one where the pool of threads is deployed (use RMI).

(b) The data file is read via a protocol handler.

(c) Results are sent in XML.

The file `reseauSocial.txt` contains data that you may use for testing your application. A line is either a message or a comment, hence of one of the three following patterns :

```
idMessage|idUser|message|user||
idCommentaire|idUser|commentaire|user|pidCommentaire|
idCommentaire|idUser|commentaire|user||pidMessage
```