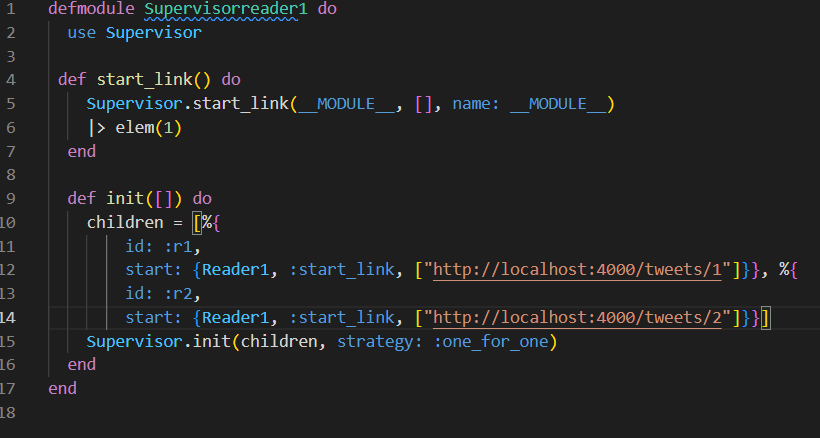
FAF.PTR16.1 -- Project 1

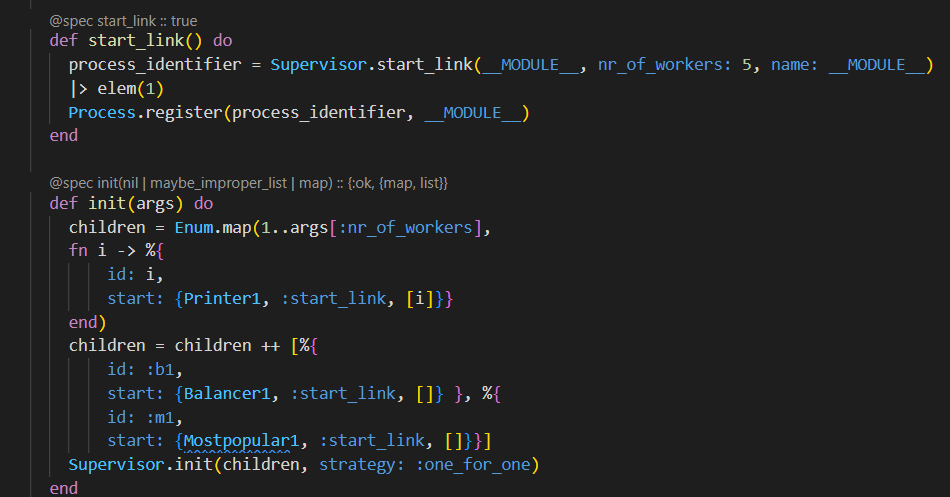
**Performed by:** Covtun Serghei, group FAF-203

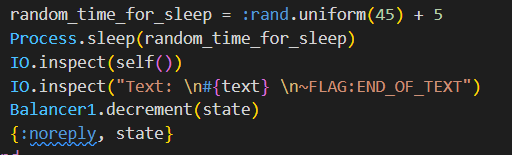
**Verified by**: asist. univ. Alexandru Osadcenco

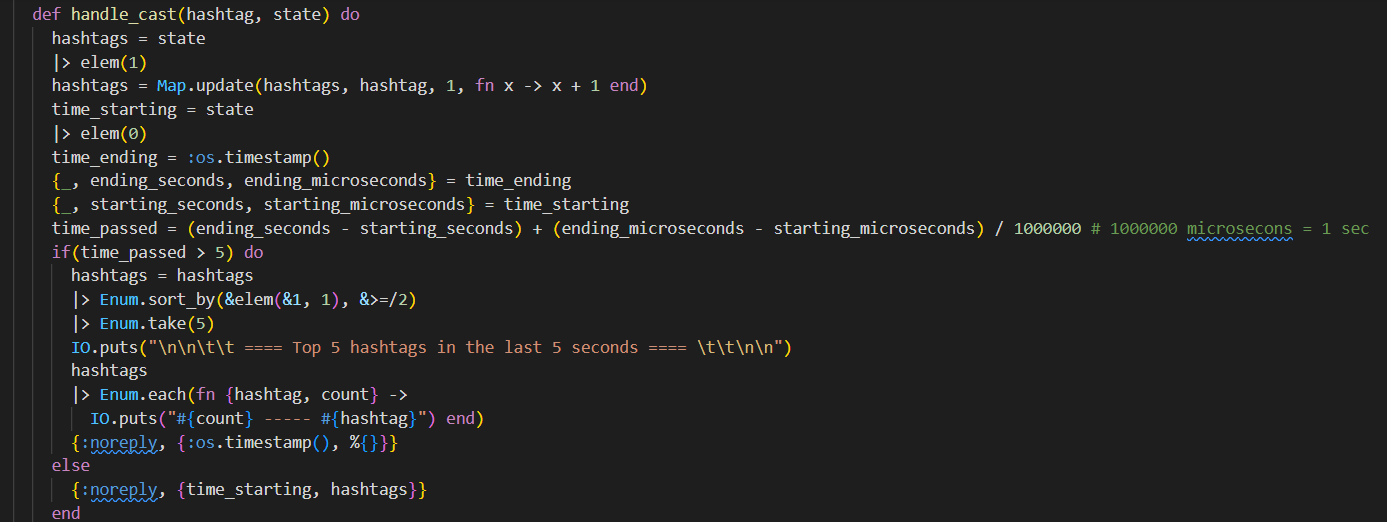
**P0W1**

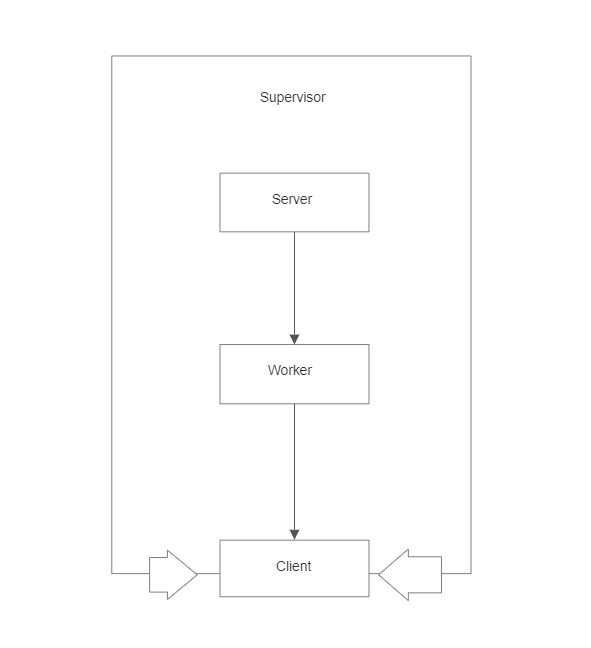
**Task1:** Write an actor that would read SSE streams. The SSE streams for this lab are available on Docker Hub at alexburlacu/rtp-server, courtesy of our beloved FAFer Alex Burlacu. Create an actor that would print on the screen the tweets it receives from the SSE Reader. You can only print the text of the tweet to save on screen space. Create a second Reader actor that will consume the second stream provided by the Docker image. Send the tweets to the same Printer actor. Continue your Printer actor. Simulate some load on the actor by sleeping every time a tweet is received. Suggested time of sleep – 5ms to 50ms. Consider using Poisson distribution. Sleep values / distribution parameters need to be parameterizable. Create an actor that would print out every 5 seconds the most popular hashtag in the last 5 seconds. Consider adding other analytics about the stream.



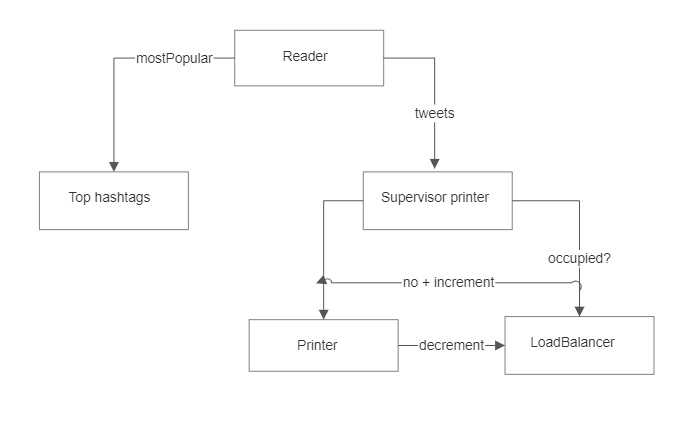








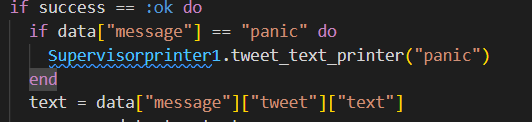
Message flow diagram

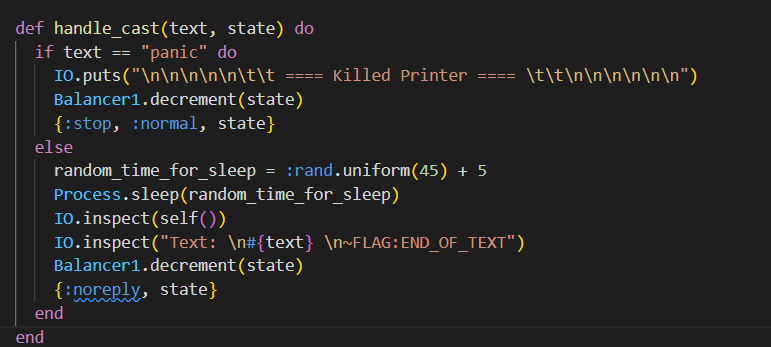


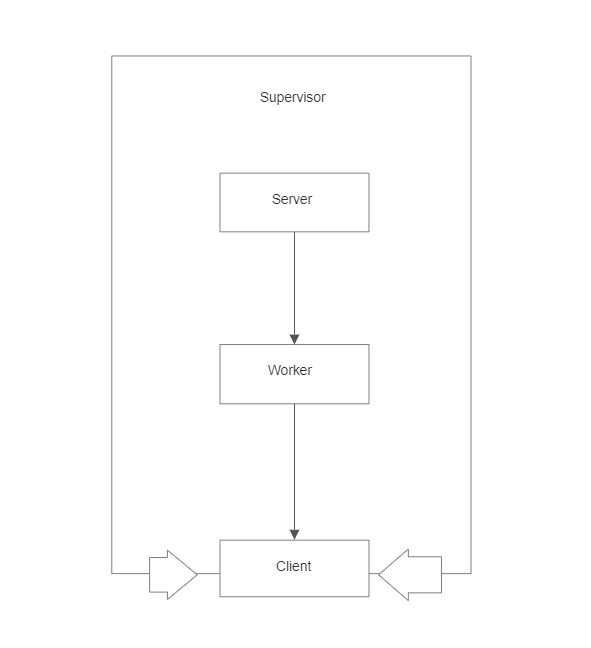
Supervision Tree Diagram

**P0W2**

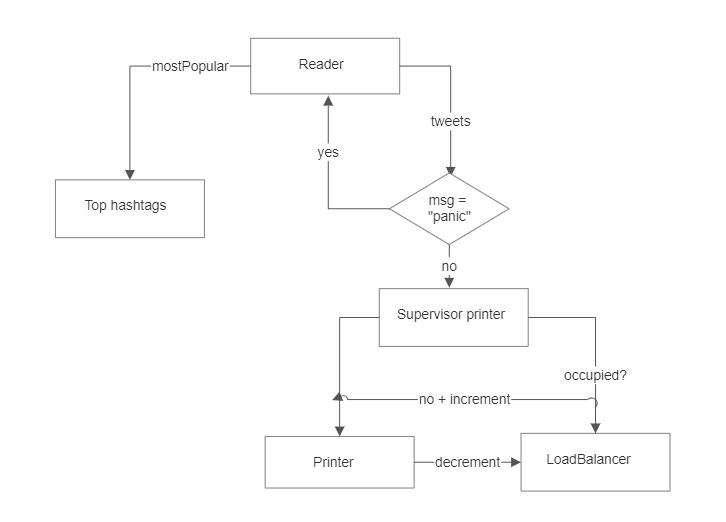
**Task:** Create a Worker Pool to substitute the Printer actor from previous week. The pool will contain 3 copies of the Printer actor which will be supervised by a Pool Supervisor. Use the one-for-one restart policy. Create an actor that would mediate the tasks being sent to the Worker Pool. Any tweet that this actor receives will be sent to the Worker Pool in a Round Robin fashion. Direct the Reader actor to sent it’s tweets to this actor. Continue your Worker actor. Occasionally, the SSE events will contain a “kill message”. Change the actor to crash when such a message is received. Of course, this should trigger the supervisor to restart the crashed actor.







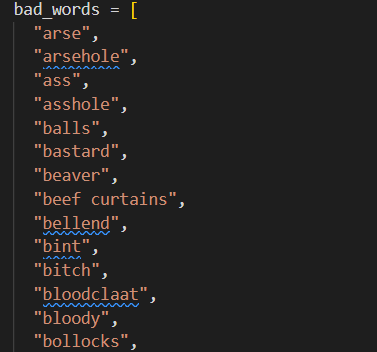
Message flow diagram

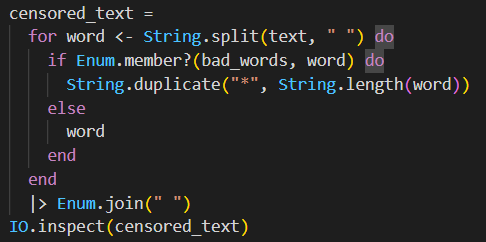


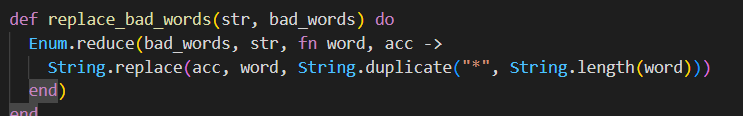
Supervision Tree Diagram

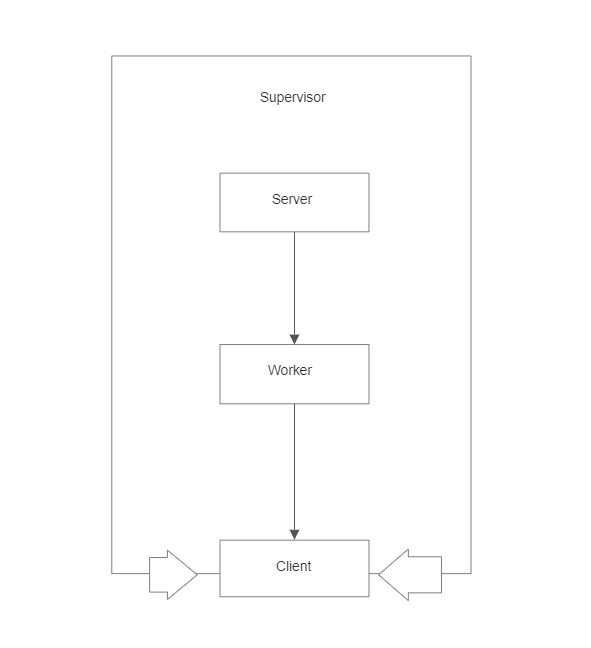
**P0W3**

**Task1:** Continue your Worker actor. Any bad words that a tweet might contain mustn’t be printed. Instead, a set of stars should appear, the number of which corresponds to the bad word’s length. Consult the Internet for a list of bad words.

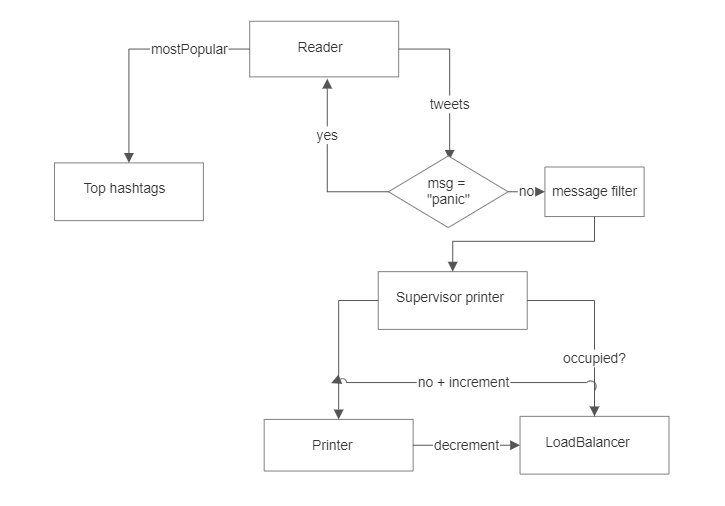








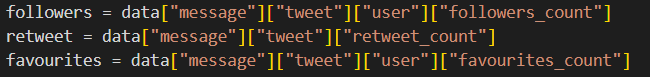
Message flow diagram

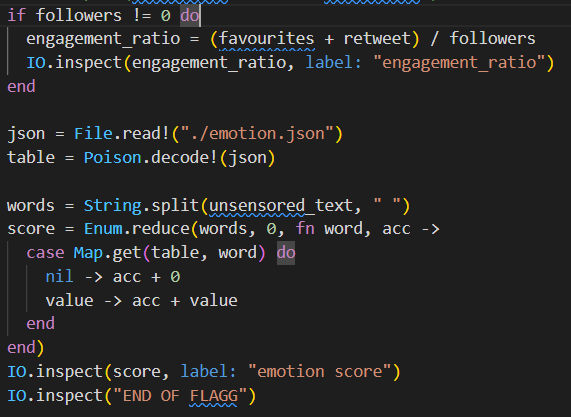


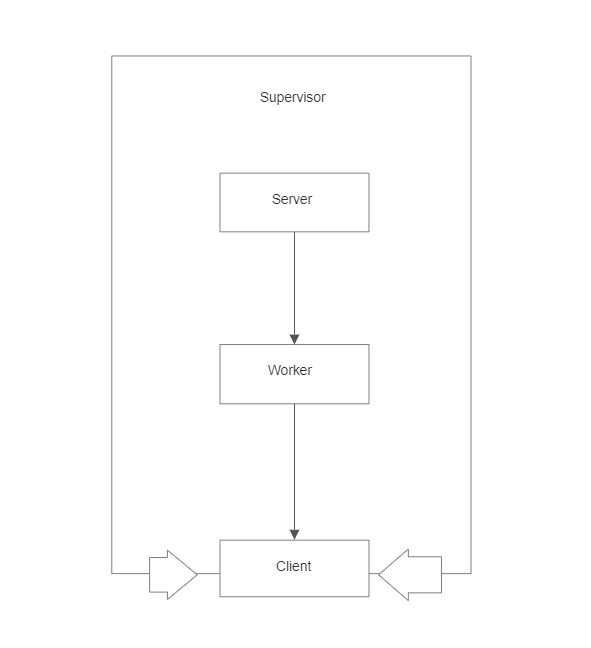
Supervision Tree Diagram

**P0W4**

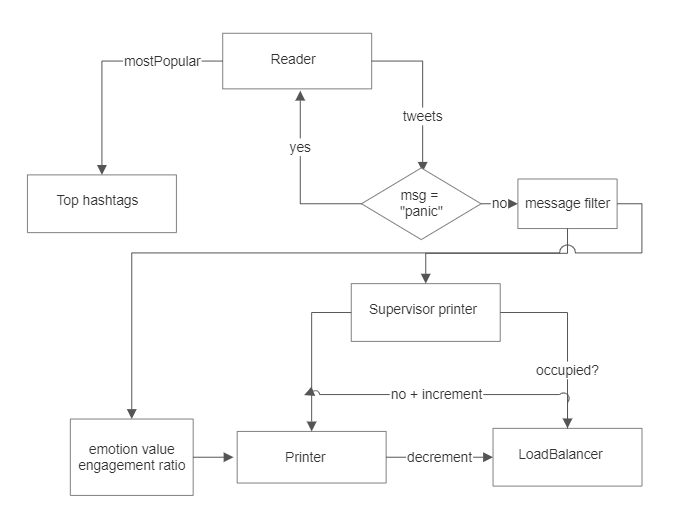
**Tast1:** Continue your Worker actor. Besides printing out the redacted tweet text, the Worker actor must also calculate two values: the Sentiment Score and the Engagement Ratio of the tweet. To compute the Sentiment Score per tweet you should calculate the mean of emotional scores of each word in the tweet text. A map that links words with their scores is provided as an endpoint in the Docker container. If a word cannot be found in the map, it’s emotional score is equal to 0. The Engagement Ratio should be calculated as follows: engagement ratio = #f avourites + #retweets / #followers







Message flow diagram



Supervision Tree Diagram

**Conclusion.**

During this laboratory work we learned more of elixir language, also we tried to write programs in functional way. I learned how to work with SSE streams and also to create stream processing system.

**Bibliography.**

<https://elixir-lang.org/docs.html>

<https://hexdocs.pm/elixir/1.12.3/Kernel.html>

<https://elixirschool.com/en>