

17-708: Software Product Lines

Assignment 4: Chat Framework

Task 1

Reimplement the Chat system from Assignment 3 as a (black-box) framework in which each feature is implemented as a plugin that can be compiled separately. You may either refactor your existing implementation or redesign it from scratch.

Provide a simple plugin loader such that different configurations can be created without any modifications to the source code. Ideally, plugins are placed as .jar files in a folder and the system loads all plugins in that folder.

Task 2

Add a new feature *Spam filter* as additional plugin. Again, a simple implementation is fine, for example one rejecting all messages from a user who posted more than 10 messages within 2 seconds for the next 5 minutes.

Task 3

Reflect on the implementation, especially the following three aspects:

1. **Effort:** Was the creation of the framework a difficult task? What were the challenges?
2. **Modularity:** Are (all) plugins understandable in isolation? Are there interactions among the plugins possible? How does this implementation style affect code quality and maintenance?
3. **Extensibility:** How flexible is the framework? Which other features would be easy to add, which features would be challenging and why?

A good reflection document does not only present facts but also interprets them in depth and adds additional value by adding judgments from experience or opinions. Do not merely recite textbook opinions but discuss whether your experience aligns with them. A good reflection document will include concrete statements about lessons learned, with clear supporting evidence, such as examples, to support the claims. For example, "It is easy to understand in isolation." is weakly supported. One could strengthen it with examples from

the development experience as follows: “One source of difficulty was the integration of components A and B, because the API for A was similar but slightly different from that of B... In the a larger system, this could have the following consequences.. Once could try to use ... instead.”

Deadlines, Technicalities, and Hints

Finish the implementation and reflection document by Oct 12, 11:59pm.

Create a branch “framework” in the provided GitHub repository. Provide brief instructions of how to start different configurations of your implementation in a README file in the root of the repository.

Add the reflection document to the root of the directory. Both .pdf files and pure text files (e.g. Markdown) are acceptable. Create explicit subsections in that document for the three aspects Effort, Modularity, and Extensibility. The entire reflection document should not exceed 1500 words (soft limit).

At the deadline, we will take a snapshot of your GitHub repository.

As usual, follow reasonable development practices, e.g. using Java naming conventions, making incremental cohesive commits with meaningful commit messages, including documentation or tests where appropriate.

Grading

We expect

- A black-box framework for the chat (both client and server if needed)
- A modular implementation of all features as plugins, including the new spam filter feature.
- A plugin loading mechanism that allows to load different configurations without any source code changes.
- A README file containing build and start instructions.
- A reflection document with three subsections discussing in-depth the stated questions.