Second assignment

The main **goal** of the second assignment is to explore property design decisions, their implementation, and their architectural knowledge. Specifically, we aim to answer the following research questions for each quality attribute:

1. What tactics or patterns are used to achieve quality attributes?
2. How do software engineers implement architectural tactics or patterns to achieve quality attributes?
3. What architectural knowledge concepts were behind making the design decisions?

Each group will focus on a specific project. The table below shows which project belong to which group and provides the required information for each project.

Group	Project	Method
Group 1	Struts2	All
Group 2 + 7	ActiveMQ	Mailing-List to issue tracker + Archie
Group 3	ActiveMQ	Issue tracker to mailing lists
Group 4	Axis2	Mailing-List to issue tracker + Archie
Group 5	Axis2	Issue tracker to mailing lists
Group 9 + 12	Wicket Log4j2	Issue tracker Mailing lists
Group 10 + 8	Derby	Mailing-List to issue tracker
Group 11	Derby	Issue tracker to Mailing lists.

To achieve our goal and answer the research questions, we will follow two steps in the next weeks:

- Step 1: Search for discussed architectural tactics and patterns to achieve quality attributes and analyze their implementation.

For each quality attribute, you need to make two sub-steps: **search** and **analyze**.

1) You will **search** for tactics and patterns for each quality attribute in multiple sources: issues, emails, and source code. There are three multiple methods to search:
   a. Start from issues to code and emails, check the issues for ones that contain property decisions, determine their respective commits, then search for related emails in mailing lists that discuss this decision. It is expected that the emails should be at the same time or close to the issue time. You can try to search using the issue ID or some keywords related to this issue.
   b. Start from emails to issues and code, search in the database of your project's mailing list using keywords, such as names of the quality attributes and names of tactics or patterns. See attached examples of keywords to search with. In

certain emails, you can find related issues directly. Another option is to search in the issue tracker Jira search for issues with similar keywords. Related issues should happen in closer time to the emails. If you found related issues, then you can find the respective commits as well.

c. Start from source code to issue and emails, you can search using the Archie tool (attached) and determine classes that implement security and availability tactics. Another way is to search using keywords in the browser. Verify the instances that they are correct and implement tactics or patterns. If you found the tactic or pattern through source code search, then use the Git command git blame -L to determine the commits responsible on the implementation and back trace its issue, and then emails.

If you found the tactic or pattern through an issue, then try to search if there is a respective email discussion on this tactic or pattern.
If you found the tactic or pattern through an email, then try to search if there is a respective issue discussion on this tactic or pattern, and if there is a respective implementation.

For each found tactic or pattern, record the following in a table:

a. How you found the tactic or pattern.
b. Respective issue ID.
c. Respective email ID.
d. Respective implementation in source code: Classes and/or methods that implement the tactic or pattern.

If you cannot find one of these fields, then leave it blank. But record all instances you found, even if they are not implemented or have no issues or emails.

2) You will **analyze** the implementation of the found tactic or pattern.

Check how the tactic or pattern is implemented. Explain using words and/or models how the tactic or pattern is implemented. There are different options: class diagram, using reflection, using certain library, using parameters, ….

By the end of Step 1, you can answer the research question:

- What tactics or patterns are used to achieve quality attributes?
- How do software engineers implement architectural tactics or patterns to achieve quality attributes?

- Step 2: Explore architectural knowledge of property decisions.

Analyze the architectural knowledge of design decisions. This should be either in the issues or emails. Architectural knowledge includes, its design issue, alternative solutions, rationale of design decisions, such as assumptions, trade-offs, and benefits and drawbacks.

By the end of Step 2, you can answer the research question:

- What architectural knowledge concepts were behind making the design decisions?

Week 1:

Perform step 1 on availability, security, and performance.

Week 2:

Perform step 1 on maintainability and interoperability.

Week 3:

Perform step 2 on a sample of 15 design decisions from all quality attributes. Choose decisions that have more discussions in their issues or emails or both to analyze their architectural knowledge.