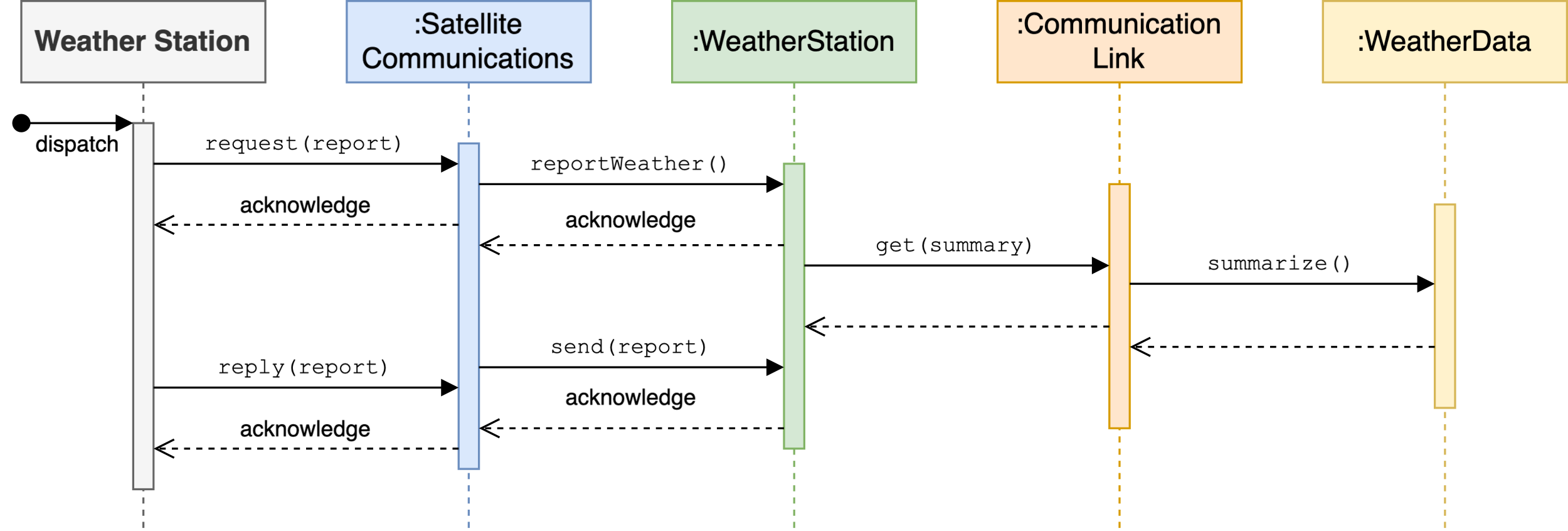
[**Homework 7-8**](https://github.com/hendraanggrian/IIT-ITM511/blob/assets/assignments/hw7-8.pdf)**: Design, implementation and software testing**

**Problem 1**

*Develop the design of the weather station to show the interaction between the data collection subsystem and the instruments that collect weather data. Use sequence diagrams to show this interaction.*

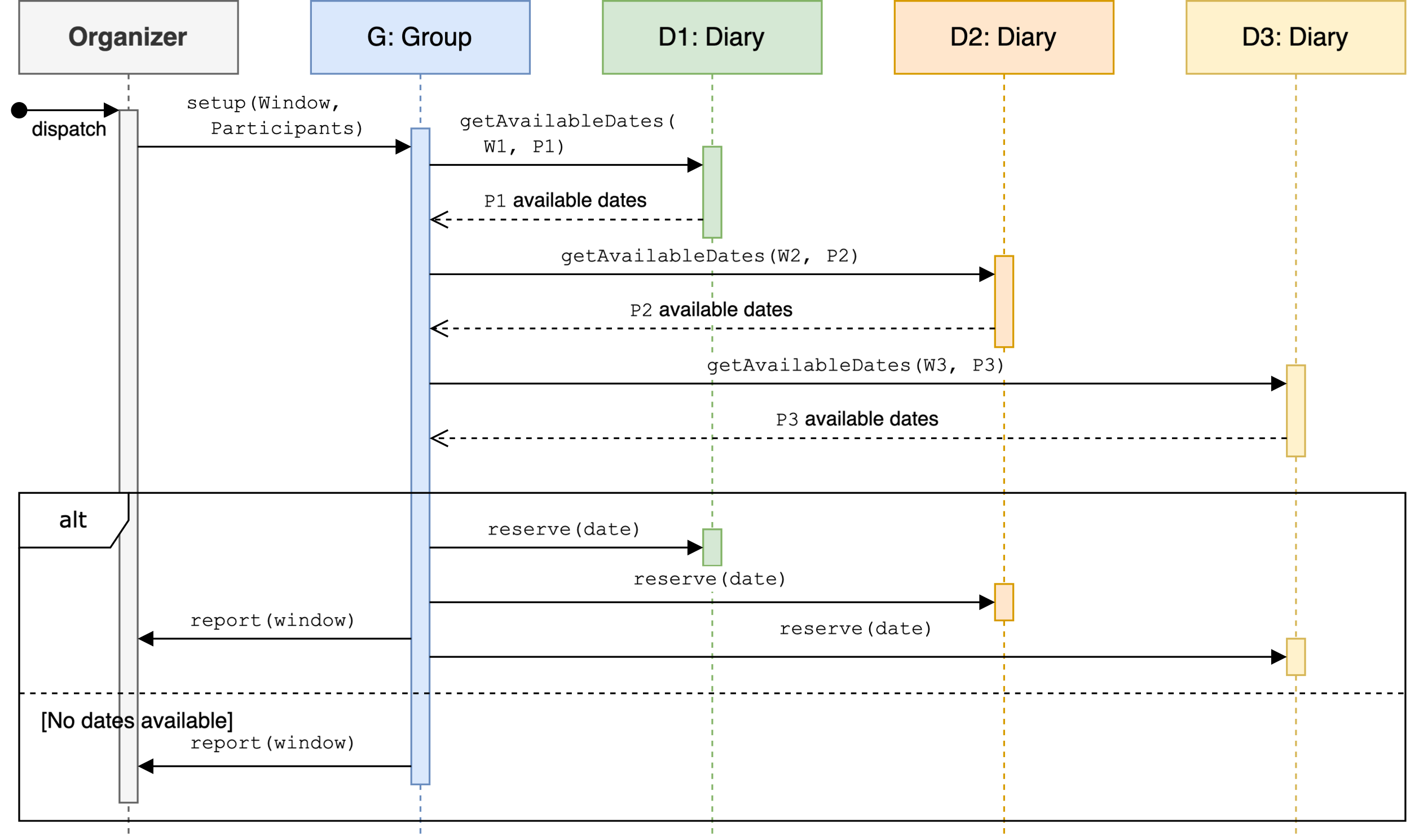
(Sommerville, 2016)



[View source](https://github.com/hendraanggrian/IIT-ITM511/blob/main/assignments/hw7-8/figures.drawio)

**Problem 2**

*Draw a sequence diagram showing the interactions of objects in a group diary system when a group of people are arranging a meeting.*



**Problem 3**

*When code is integrated into a larger system, problems may surface. Explain how configuration management can be useful when handling such problems.*

A software system is typically a culmination of many smaller components and external resources. While each piece may work individually, unexpected issues will likely appear when combining them all, especially in a large or fast-paced project. A configuration management tool helps reduce this risk by creating multiple checkpoints to manage changes, detect early mistakes and focus work on the most pressing issue (What is configuration management?, 2023). We can also revert progress to a previous state when the problem is beyond repair. Implementing these measures will improve team collaboration and productivity by minimizing repetitive work among developers.

**Problem 4**

*Testing is meant to show that a program does what it is intended to do. Why may testers not always know what a program is intended for?*

Independent software testers' lack of involvement in the development process of the main program may limit their understanding of specific design choices and mechanics. The problem is exacerbated when there is limited communication in the software team or when the solution by the original authors is poorly documented. If left untreated, the written tests could miss core functionality and lead to a flawed source code.

**Problem 5**

*Some people argue that developers should not be involved in testing their own code but that all testing should be the responsibility of a separate team. Give arguments for and against testing by the developers themselves.*

1. **Test own code:** Developers should know how to systematically test each software component by leveraging their in-depth knowledge of the software they wrote. Their initial knowledge could translate to a faster testing time, though they need to be proficient in software testing for such conditions to materialize. Moreover, trusting developers to conduct their testing promotes a sense of ownership within a software team.
2. **Test independently:** A software team may also decide to offload some – if not all – testing workload to a separate testing team. They are theoretically the best candidate to test the program for their expertise and are unlikely to be conflicted with personal relationships. However, their employment warrants more testing time and requires extra effort to get points across multiple teams.

# References

Sommerville, I. (2016). Software Engineering. In *Design models* (10 ed., p. 206). Pearson Education.

*What is configuration management?* (2023, June 22). Retrieved from Red Hat: https://www.redhat.com/en/topics/automation/what-is-configuration-management/