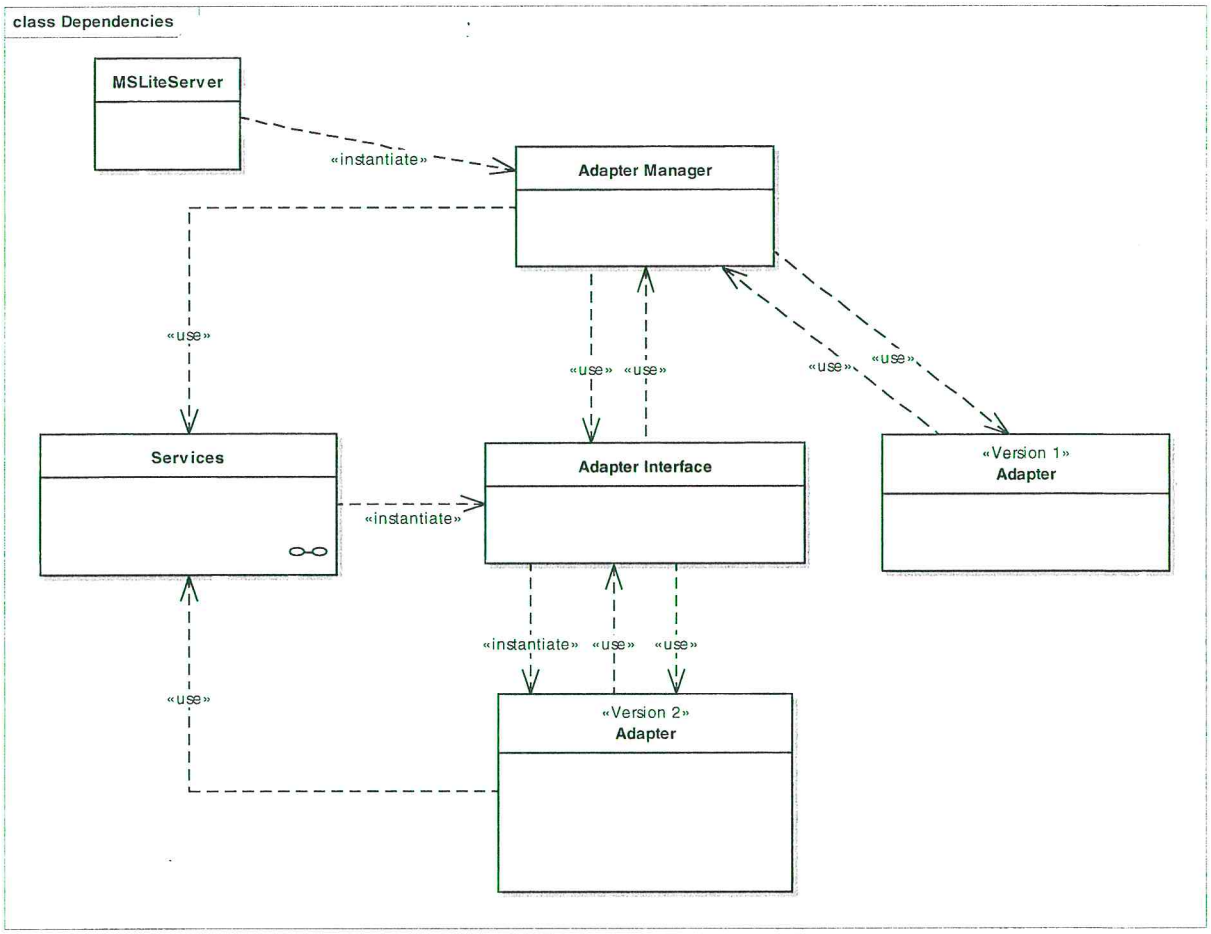
Documentation Template

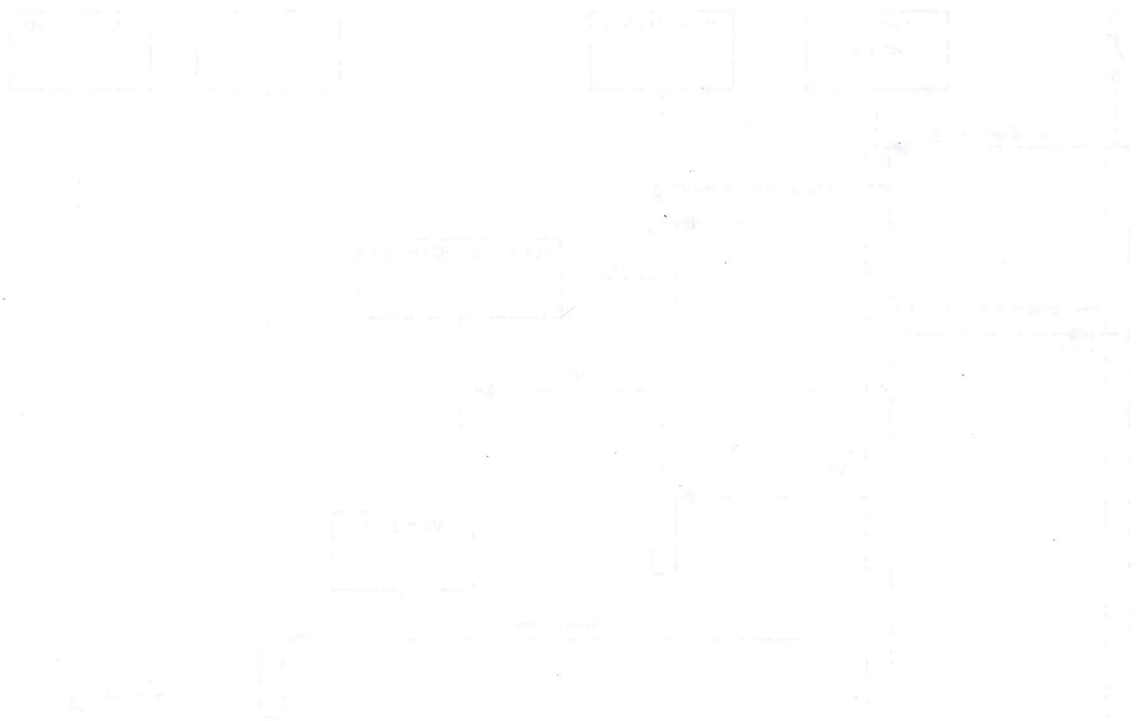
1) Create a dependency view that includes at least all the modules that need to be adjusted.
2) Create one or more sequence diagrams showing how the new modules would interact in fault situations, such as buffer overflow, threads not returning, etc.
3) For every module that needs to be changed or is newly created provide a module description with operations and responsibilities (changes only)
4) For every changed/new module create a module context diagram
5) Create the detail design hierarchy for the changed/ new modules.

DEPENDENCY VIEW

Create a dependency view that includes at least all the modules that need to be adjusted.
Similar to this one:

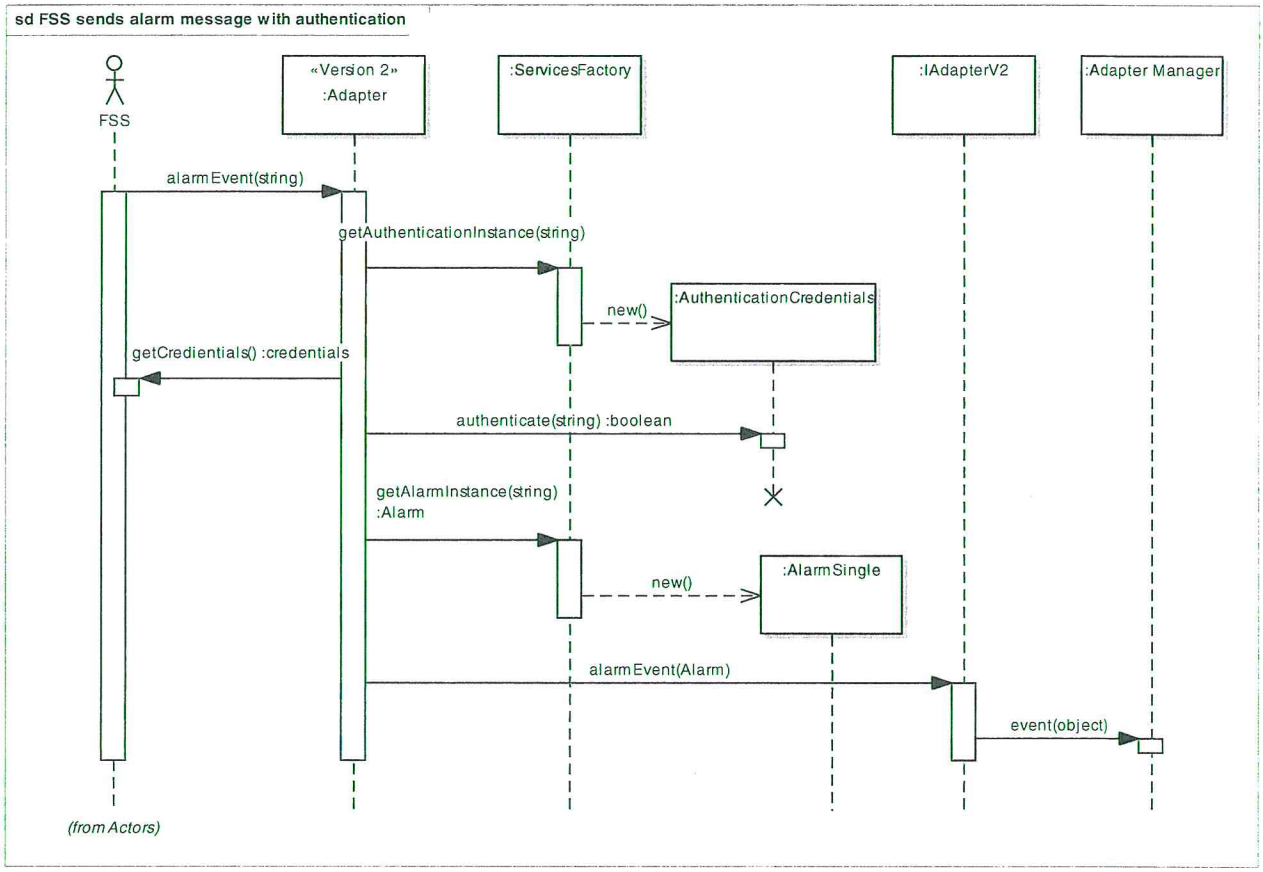


Exercise 4: See whiteboard picture

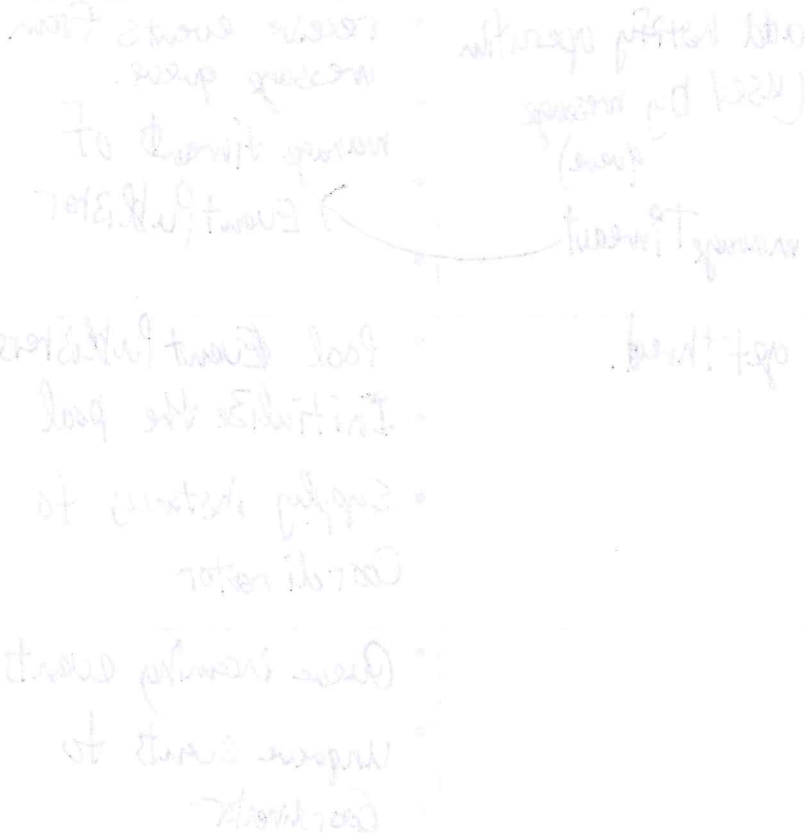


SEQUENCE DIAGRAMS

Create one or more sequence diagrams showing how the new modules would interact in fault situations, such as buffer overflow, threads not returning, etc.
Similar to this one:



Ex. 4: See whiteboard pictures



MODULE DESCRIPTION

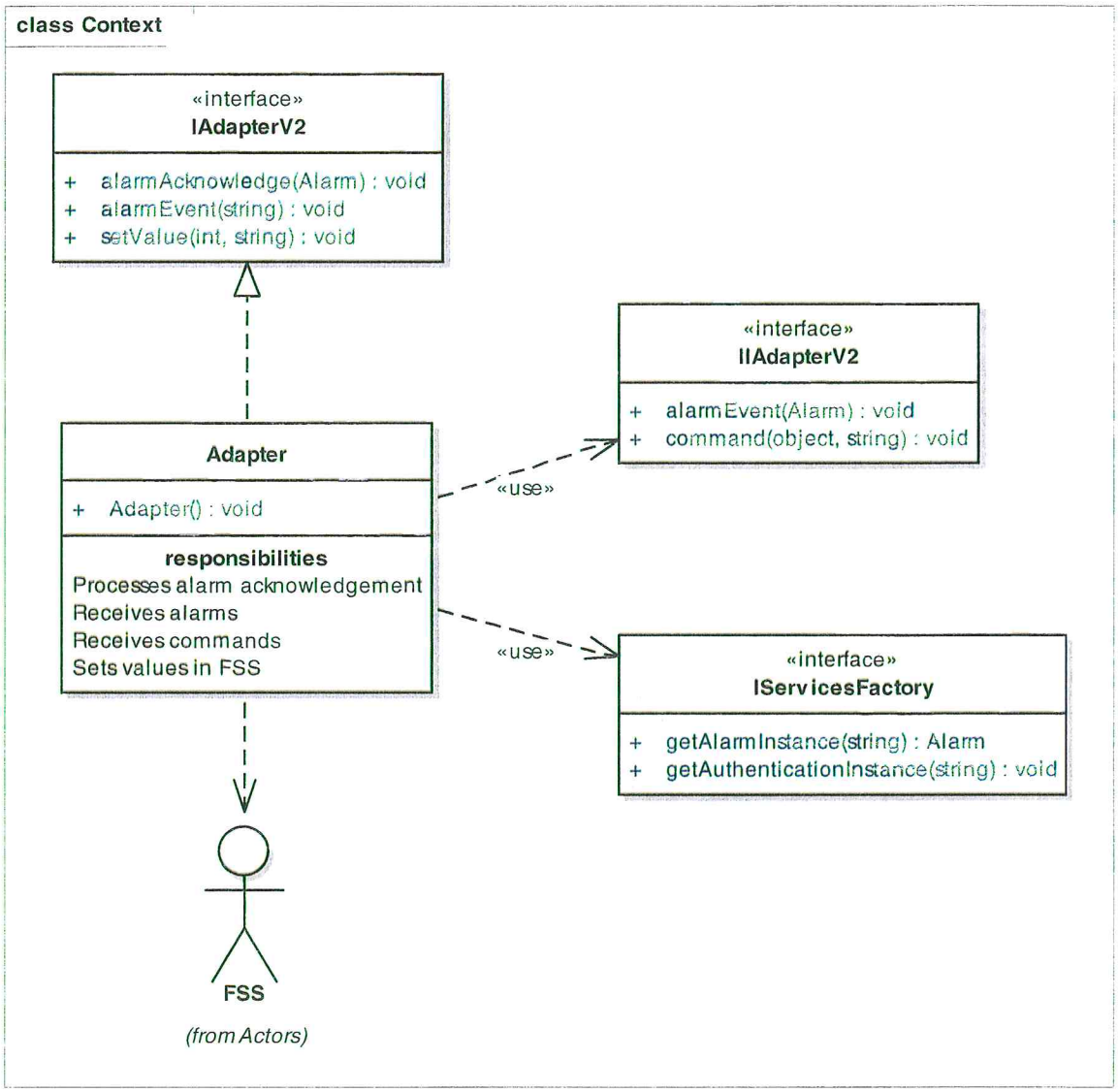
For every module that needs to be changed or is newly created provide a module description with operations and responsibilities (changes only).

Module	Operations	Responsibilities
M1 Coordinator	• add notify operation (used by message queue) • manage Timeout	• receive events from message queue. • manage timeouts of EventPublisher
M2 ThreadPool	• getThread.	• Pool EventPublishers. • Initialize the pool • Supply instances to Coordinator.
M3 MessageQueue		• Queue incoming events • Unqueue events to Coordinator.
M4		

Module	Operations	Responsibilities
M5	•	•
	•	•
	•	•
	•	•
M6	•	•
	•	•
	•	•
M7	•	•
	•	•
	•	•
M8	•	•
	•	•
	•	•

MODULE CONTEXT DIAGRAMS

For every changed/new module create a module context diagram.
Similar to this one:



DETAILED DESIGN HIERARCHY

Create the detail design hierarchy for the changed/ new modules
For example something like this:

1. PublishSubscribe
   1.1. BufferReceiver
      1.1.1. Context diagram
      1.1.2. Class diagram
      1.1.3. Side-by-side view
      1.1.4. Classes