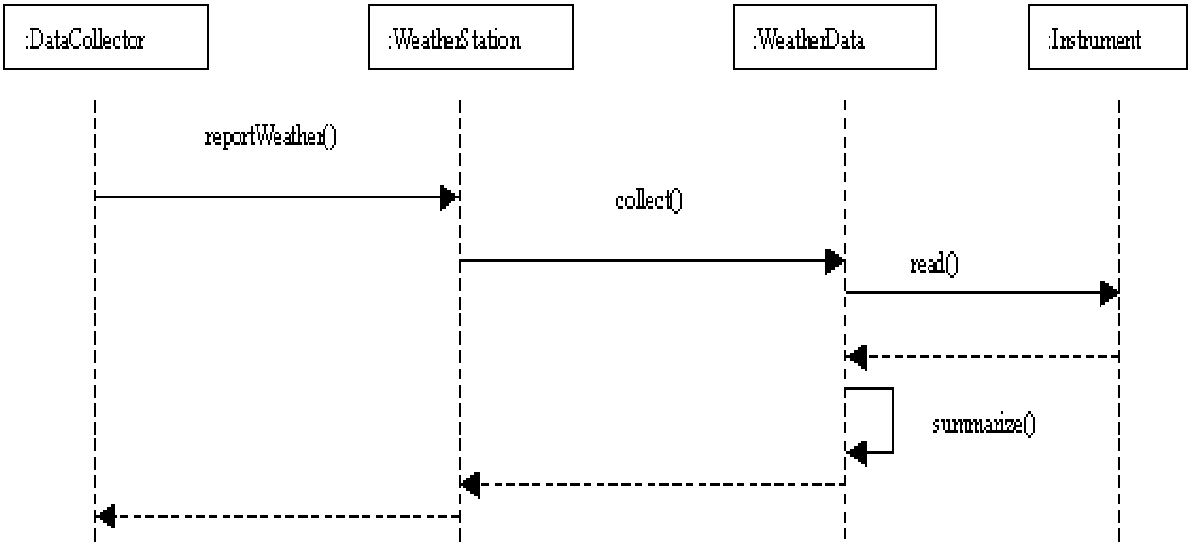
Lab 2: Solutions

A.

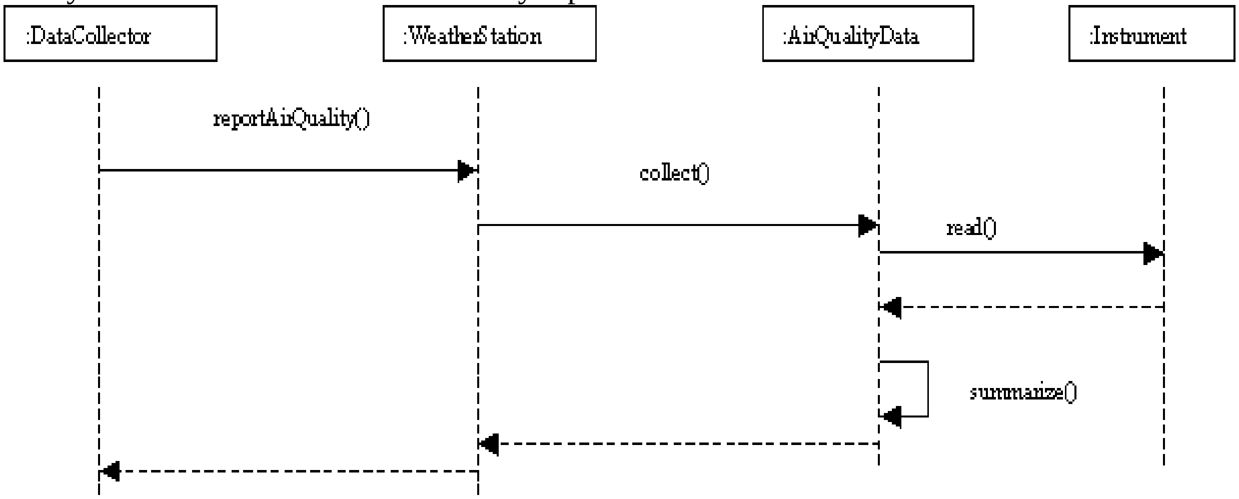
1.a

- *Report Weather Readings Scenario*



- *Report Air Quality readings Scenario*

Both diagrams are similar: you may describe both scenarios using one diagram, in which case you should describe the concurrency aspects.



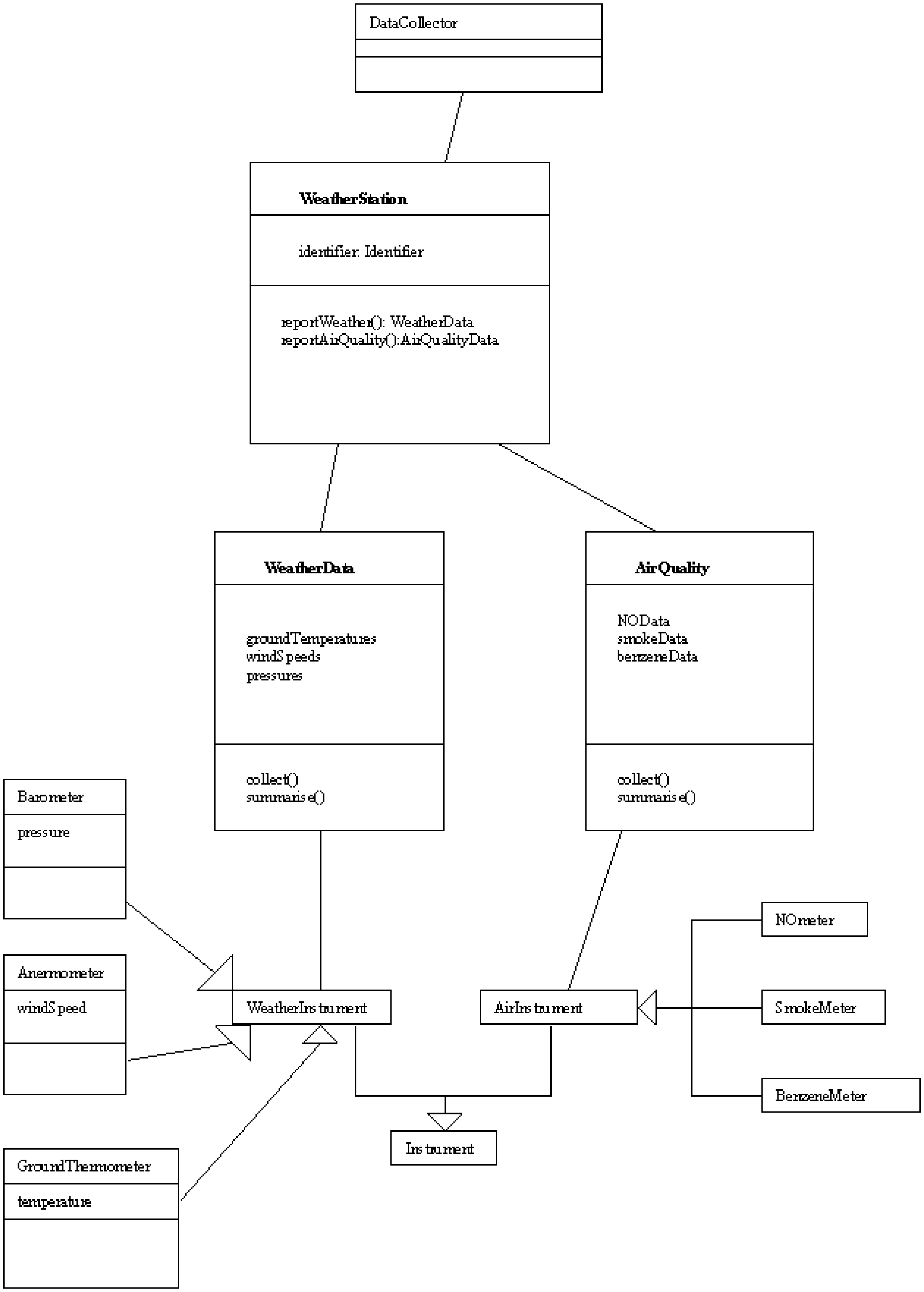
b. Collaboration and sequence diagrams are dual; using Rose, you can generate either diagram from the other automatically. In any case, you must find the same objects, messages, and communication paths.

2.

a. Entity class models information and associated behaviour that is generally long lived. This type of class is typically independent of their surroundings; that is they are not sensitive to how the surroundings communicate with the system. Boundary classes handle the communication between the system surroundings and the inside of the system. They can provide the interface to a user or another system. Control classes model sequencing behaviour specific to one or more use cases: they represent the dynamics of the use case.

- **Boundary classes**: this use case interacts only with the data collection instruments. Hence, the boundary classes would be the classes encapsulating the instruments (weather and air data collection instruments: GroundThermometer, Anemometer, Barometer,Nometer, Smokemeter and Benzenmeter).
- **Entity classes**: this scenario deals with weather and air data collection. We can identify two entity classes*: WeatherData* and *AirQualityData*. *AirQualityData* class represents the air quality data.
- **Control classes**: there are two control classes that handle the flow of events for the use case: *WeatherStation* and *DataCollector*.

b. and c. We derive the classes participating to the use case from the interaction diagrams. We shall find in the class diagram the same relationships (correspond to the message flows) and the same operations.



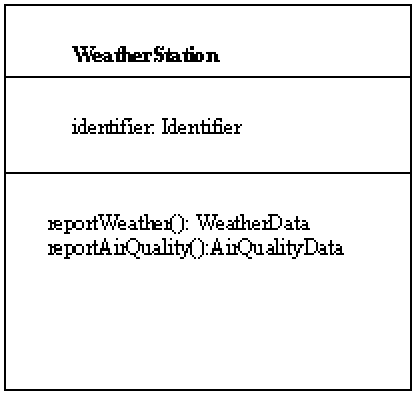
3.

Operations:

Operations may be identified using interaction diagrams: operations are associated to the messages exchanged.
Documentation for the collect() operation of the WeatherData class:
Collect weather information from the weather instruments.
Inputs: request from the weather data collection subsystem
Outputs: summarised weather data sent to the weather collection subsystem.

Attributes:

Example: attributes and operations for WeatherStation class:


WeatherStation
identifier: Identifier
reportWeather(): WeatherData
reportAirQuality():AirQualityData

Documentation for the *identifier* attribute of the *WeatherSation* class:
The unique reference used to distinguish any weather station in the WMS system.

The documentation should state briefly the purpose of the attribute, not the structure of the attribute. For instance a bad definition for the *identifier* attribute would be *"a character string of length 15"*.