For each of the four test we got a fellow student to test them out for us. This person is not taking the class, and we believe this will strengthen the tests and find errors we might have missed.

Test-case 1:

In the first case the person should check if there are any active alarms. Then schedule an alarm for so go back to the list of active alarms to see if it is registered.

This scenario test if an alarm is put in list of active alarms after creating it.

Expected result is first to see that there are no active alarms. For so creating one and see that it is put in the list of active alarms.

Test-case 2:

In the second test-case the person will try to schedule an alarm for a time that has passed. Then schedule an alarm with a valid time. After the alarm is scheduled, the person is going to cancel it. Then check list of active alarms to see it is removed.

In this case you test if the program handles time correctly. That you cannot schedule an alarm for a time that has already passed. Then the case check if an alarm is removed from the list of active alarms when you choose to cancel it.

Expected result is to tell user that you cannot schedule an alarm with a time that has already passed and ask them for a valid time. Finally when the alarm is cancelled, it should also be removed from list of active alarms.

Test-case 3:

Schedule two alarms with different time. Wait out the first alarm and check if it goes off at the scheduled time. Then the user should exit the program. After exiting the user should wait and see if the program is truly done and the last alarm does not ring.

In the third test-case we will check both if an alarms rings at the time it is scheduled for. And if you exit the program, that is truly cancelled.

Expected result is that the first alarms rings in real time. Then when the program is exited, the last alarm does not ring.

Test-case 4:

Try writing something that is not given as one of the commands. Then try to cancel an alarm. After that the user tries to schedule an alarm with invalid format.

In the last test-case you test out functionality that make no sense. By that we mean doing things in an order that is obviously incorrect. The second thing this case tests for is that you cannot have type-errors.

Expected result is that the person will be asked to write a valid input until a correct one is passed. Then you will be given feedback that you cannot cancel any alarms when there isn’t any active. Finally in this test-case when you try to schedule an alarm you will be prompted the given format until a correct one is passed.