Unit Testing

Alarm > String: name, double: time, long: location, sound, String: AlertType > double getTime()

Alarm.getTime(...)

The Alarm class's main purpose is to create the alarm itself. In this case, we will be testing the getTime() function. The getTime() function is tasked to get the time at which the alarm was set off. In order to do this the function will take a double as a parameter and return that exact same double to the setTime() function. For instance, if the alarm went off at 17.30 the getTime() function will take 17.30 as its parameter and retrieve its value; so that it can be accessed by the setTime() function.

Alarm.getTime(double time)
return 17.30

Functional Testing

AlarmSystem & Alarm > String: detectionType, Alarm(): alert > void createAlarm(), boolean resolveAlarm()

AlarmSystem.createAlarm(...)

In the AlarmSystem class, the user can turn off the alarm and create an alarm. In this case, we are testing the createAlarm() function which creates and stores the alarm in the database. For this function to work, it must take in two parameters which are detectionType and alert. The alert parameter is an object from the Alarm class and detectionType is the string that explains the type of detection the user has categorized the detection to be. For example, an alarm goes off at 3.00, it is located in the northern part of the national park, and the sensor that detected the sound is called “Delta.” The ranger reviews the detection and categorizes it as a “suspected” detection. The createAlarm() method combines the information from the Alarm class with the ranger's classification of the detection.

AlamSystem.createAlam(Alarm alert, String detectionType)
Time: 3:00 am
Location: 11011°
Sensor: Delta
Detection Type: Suspected

System Testing

userRanger > ControlSystem > Alarm > AlarmSystem > Report

For the system testing, we will be seeing if the process of logging in and accessing the reports work. In order for this process to start the user must have access to the system, if the user has access he will then be sent to the ControlSystem() class. In the control system, the user will have control of the full system, however, in this scenario the user is turning off and categorizing the alarm that was set off. Afterward, he will be sent to the Report() class, where he can view the reports in chronological order. This is a great example of a testing scenario: Ranger #1827, with username "ranger7" and password "Peanut2003", has categorized the alarm Delta as "suspected" and is asking to view its report. The system will go through both the login() and checkID() functions and return true. Giving him access to the control system he will then turn off the alarm which will make the function resolveAlarm() return false and the createAlarm() method will combine both sensor and classification information and store it in the database. Lastly, udder will be sent to the Report() class where the function sort() will store each report in chronological order in a vector and display with the display() method.

Login:

- Username: ranger7, Password: Peanut2003, ID#: 1827

Alarm:

- Time: 3:00 am
- Location: 11011°
- Sensor: Delta
- Detection Type: Suspected

Reports:

- Jan 31, 2023 / 12:00 pm / Sensor: Comic / Location: 10001° / Type: Definite
- Feb 21, 2023 / 15:12 pm / Sensor: Bravo / Location: 11111° / Type: Definite
- Mar 24, 2023 / 3:00 am / Sensor: Delta / Location: 11011° / Type: Suspected