**Problem Statement 1**

**Install a pyunit/nose framework which adds/updates/removes the Alarms implementation**

*a) Add an alarm with alarmId as Key, and (Title, Time, Repeat = false/true) as value etc*

*b) Update an existing alarm given by alarmId, where Title/Time/Repeat can be edited*

*c) Remove/delete an alarm by its alarm ID*

*d) Clear / Remove all the alarms at a time*

*e) Keep the count of the alarms*

*f) Get the details of an alarm by its alarm id*

This POC will focus on Unit Testing of Python based applications by using Open source framework Pyunit. The designer will learn the basic concepts of Framework and developing simple test scripts to run it on an open source Test framework.

The Python unit testing framework, dubbed 'PyUnit' by convention, is a [Python](http://www.python.org/) language version of [JUnit](http://members.pingnet.ch/gamma/junit.htm),

**Design Approach**

* The designer has to follow the Web page to understand the Pyunit Features/Functions.

<http://pyunit.sourceforge.net/pyunit.html>

* Install the Python 2.7 Version from <https://www.python.org/downloads/>
* Create a simple test Case to test the Alarm features covering a,b,c,d,e, f as specified above.
* Step by Step approach on how to create a test case is provided in the Web Page
  + The basic building blocks of unit testing are 'test cases
  + In PyUnit, test cases are represented by the TestCase class in the unittest module.
  + To make your own test cases you must write subclasses of TestCase.
* Concept of Re-using set-up code, Test Case Code and new Code will be learnt with this use case.