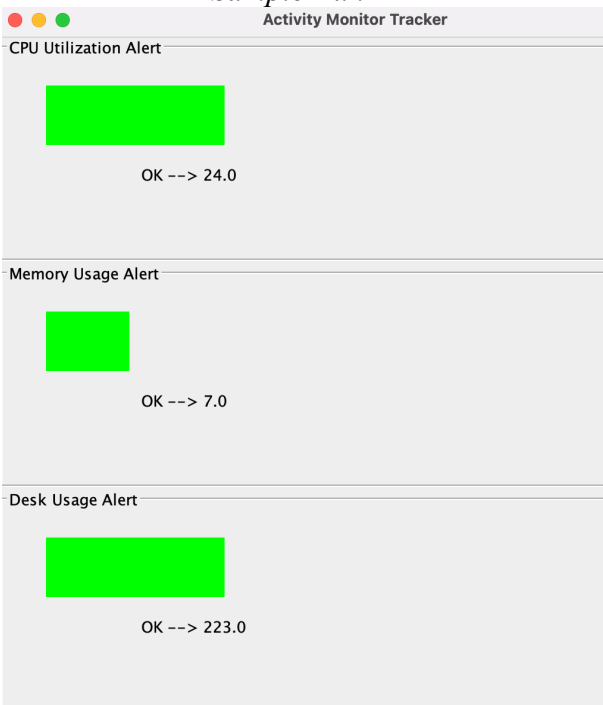
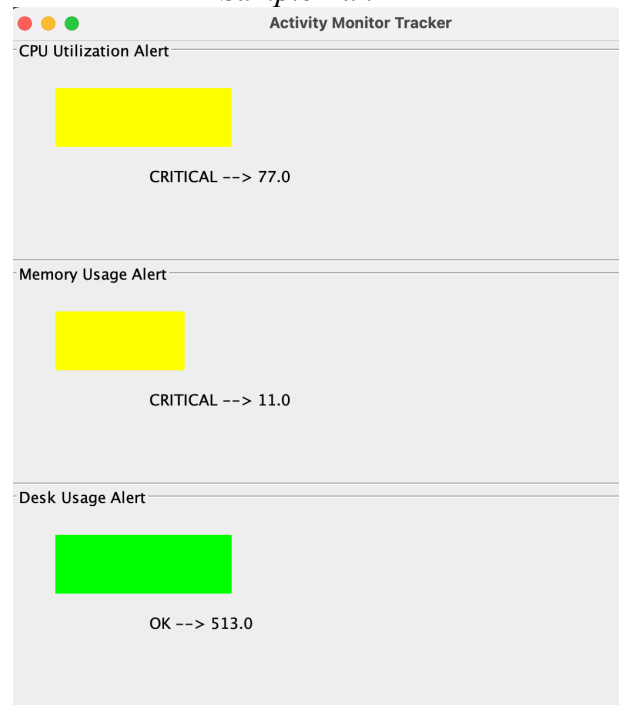


**Sample runs are provided below:**

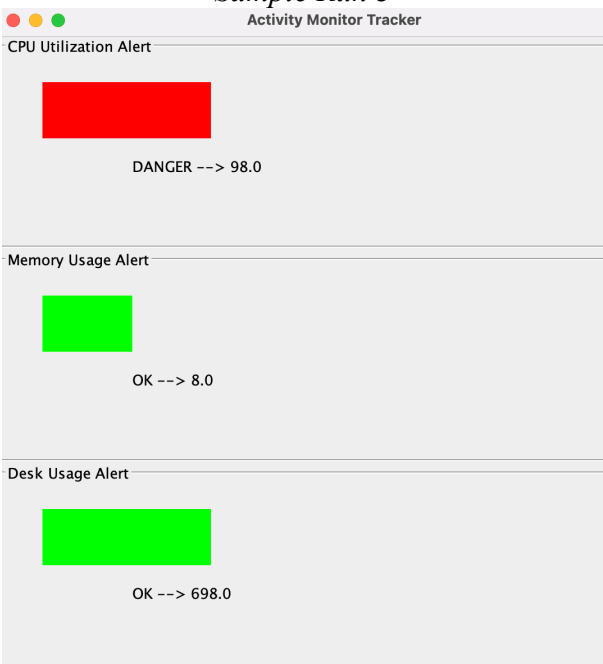
*Sample Run 1*



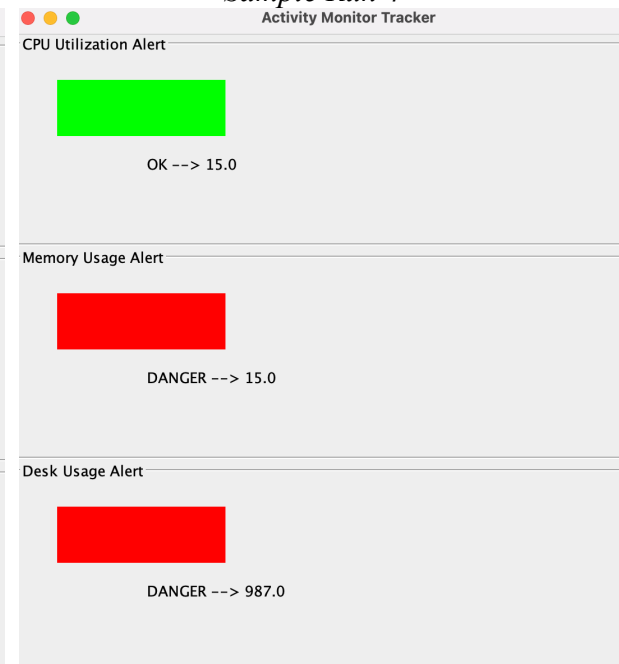
*Sample Run 2*



*Sample Run 3*



*Sample Run 4*



## The API for each sensor

<b>Package</b>	libs
<b>Class</b>	CpuUtilizationSensor
<b>Field Summary</b>	
<b>private double</b>	utilizationValue
<b>Method Summary</b>	
<b>public double</b>	readValue ()
	returns the CPU utilization value
<b>public String</b>	getReport ()
	returns the status indicating whether it is "OK", "Danger" or "Critical".
<b>public String</b>	getAlertName ()
	returns the name of the sensor such as "CPU Utilization Alert"

<b>Package</b>	libs
<b>Class</b>	DeskUsageSensor
<b>Field Summary</b>	
<b>private double</b>	usedDeskSpaceInGb
<b>Method Summary</b>	
<b>public double</b>	readValue ()
	returns the pressure value
<b>public String</b>	getReport ()
	returns the status indicating whether it is "OK", "Danger" or "Critical".
<b>public String</b>	getAlertName ()
	returns the name of the sensor such as "Desk Usage Alert"

<b>Package</b>	libs
<b>Class</b>	MemoryUsageSensor
<b>Field Summary</b>	
<b>private double</b>	usedMemoryInGb
<b>Method Summary</b>	
<b>public double</b>	readValue ()
	returns the radiation level
<b>public String</b>	getRepor ()
	returns the status indicating whether it is "OK", "Danger" or "Critical".
<b>public String</b>	getAlertName ()
	returns the name of the sensor such as "Memory Usage Alert"

Each sensor decided the status according to following table.

	<b>CPU</b>	<b>Desk</b>	<b>Memory</b>
<b>OK</b>	<75	<700	<10
<b>Critical</b>	75-90	700-900	10-13
<b>Danger</b>	>90	>900	>13