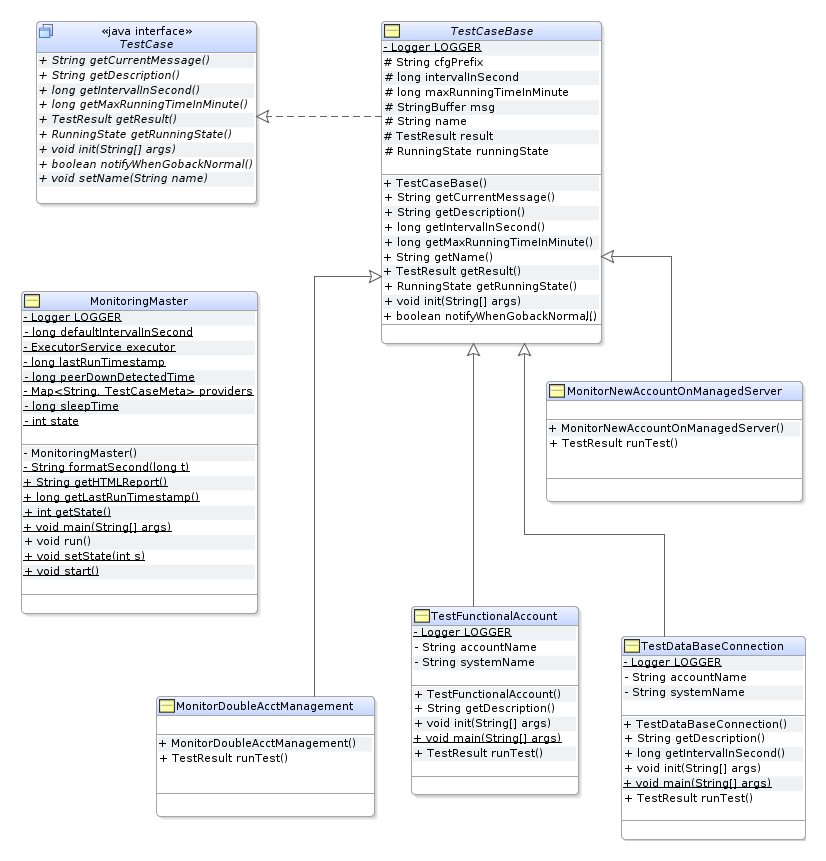
****

PAM Monitoring System

PAM monitoring system is an automation framework that runs test cases to detect system errors before end users experience issue.

1. **Key Class Design**

(Only limited test case classes are displayed in the diagram below)



* MonitoringMaster: The master thread class, it schedules test case when in active mode, and pings primary site when in standby mode.
* TestCase: Defines the API method for test cases.
* TestCaseBase: The default configuration driven implementation for most of the TestCase interface methods. All test cases should extend from this class. The only mandatory method a test case class needs to implement is runTest().
* TestDataBaseConnection: A test case that checks database connection.
* MonitorDoubleAcctManagement: A test case that detects any account is managed by two PAM system (shore + ship).
* …

The identified test cases so for:

* Basic check-out check-in
* Functional account test for all platforms (Linux, AIX, Windows, Oracle, MS SQL, make sure password rotates)
* Monitoring Mash account, it generates low severity alert 3 days before next key rotation time if still not fully synced, medium alert 2 days before, and high severity 1 day before.
* Monitoring double managed account, it make sure shore and ship PAM servers don’t managed the same account.
* Detect new account on managed system.

As we move on, we’ll add more test cases.

1. **Test Case Scheduling and Execution**

MonitoringMaster is the scheduling thread, it never blocks and submit the test case to the thread pool executor at the execution time.

The framework provides interval based scheduling, the interval for each test cases can be specified in the configuration file, e.g.

*TestDataBaseConnection.intervalInSecond = 300*

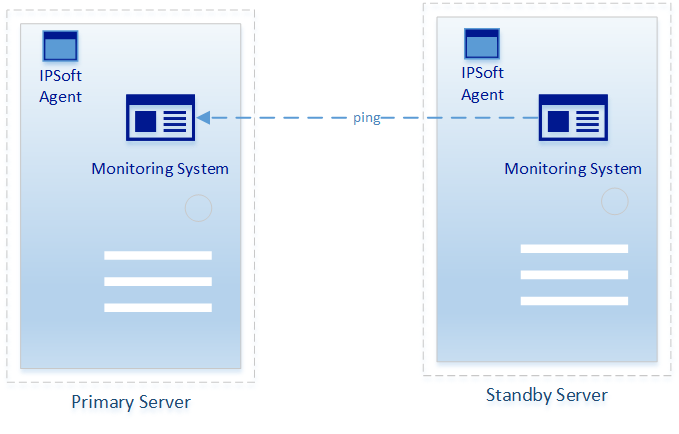
*TesCheckinCheckout.intervalInSecond = 100*

If not specified, the system default value is used:

*monitoring.defaultIntervalInSecond=600*

If a test has to be executed at certain time, e.g. 8am every day, the test case class has to overwrite the getIntervalInSecond() method. The scheduler calls this method at the startup and after each execution, returning dynamic value will allow scheduler execute the test at desired time.

1. **High Availability and IPSoft Integration**



Monitoring system is deployed to two servers, one is active and the other one is standby. The one starts first becomes the active one, the other one becomes standby server and keeps on pining the primary, it promotes self to primary 5 min after active server becomes unreachable.

In case two servers start at the same time, the configuration value determines which one is primary.

*monitoring.isPrimary=Y (or N)*

*monitoring.HA.peerServer = otherserverIP:8080/monitoring/admin*

IPSoft agent is also deployed on the monitoring server, it pings monitoring system to make sure it’s up and running.

1. **Alert Reporting**

IPSoft invokes generateReport.sh command to trigger report generation, the command use a curl command to send notification to monitoring system.

*curl http://localhost/monitoring/admin?command=generatereport*

Two types of reports are generated: Summary and Detail

Summary contains only the status GOOD or BAD for each test cases, e.g.

Test Functional Account For Oracle:GOOD

Test Functional Account For MS SQL:BAD

Test Functional Account For Windows:GOOD

Test Basic Check-out Check-in:GOOD

RegisterNewServer:GOOD

Detailed report contains the detailed message in name=value format, || is the delimiter.

TaskName=Test Functional Account For Oracle||TaskType=Monitoring||Severity=OK||TestOutcome= Successful

TaskName=Test Functional Account For MS SQL||TaskType=Monitoring||Severity=MEDIUM||TestOutcome=Failure||code=1||msg=Try get managed account winsqlddb07 (Db Instance: TESTSQLSERVER)/PAM\_SA\_User#Error:basicpbpsdemo.Client$APIException: Not Found 404. "Managed Account not found"#Test completed with ERROR!!!!!!!!!!!!!!!!!!!!

TaskName=Test Functional Account For Windows||TaskType=Monitoring||Severity=OK||TestOutcome= Successful

TaskName=Test Basic Check-out Check-in||TaskType=Monitoring||Severity=OK||TestOutcome= Successful

TaskName=Monitor MSH Account||TaskType=Monitoring||Severity=HIGH||TestOutcome=Failure||code=-1||msg=java.lang.RuntimeException: java.lang.NullPointerException

TaskName=Monitor Double Managed Account||TaskType=Monitoring||Severity=MEDIUM||TestOutcome=Failure||code=1||msg=Pull all managed system from primary ...#Pull all platforms ...#Pull all accounts from primary ...#No ship PAM server setup yet!#Test completed with ERROR!!!!!!!!!!!!!!!!!!!!

IPSoft picks up the report files in the agreed folder with the following file name format:

Summary\_PAM\_{SERVER\_NAME}\_{yy-MM-dd\_HH-mm-ss}.text

Detail\_PAM\_{SERVER\_NAME}\_{yy-MM-dd\_HH-mm-ss}.text

E.g.

Detail\_PAM\_rhlsaiampam602.na.rccl.com\_19-10-22\_14-33-44.txt

Summary\_PAM\_rhlsaiampam602.na.rccl.com\_19-10-22\_14-33-44.txt

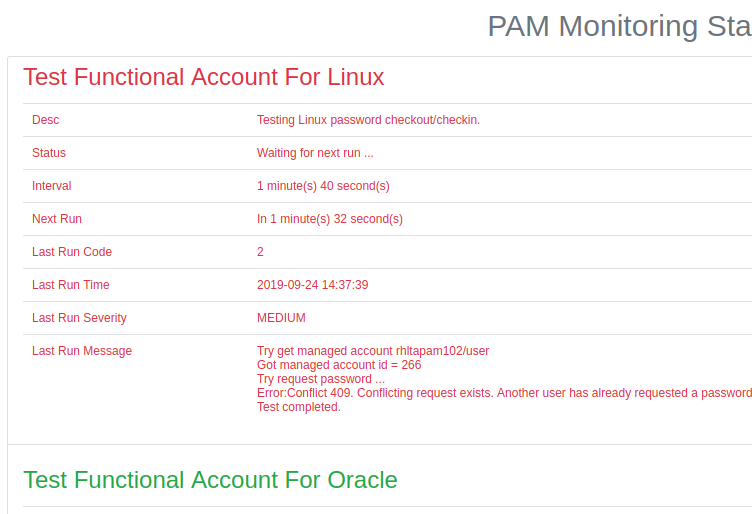
1. **Dashboard**

The dashboard url:

http://{server}/monitoring/report

E.g.

<http://rhlpiamms610.na.rccl.com/monitoring/report>



The status of test cases are display in the order of severity, red color for high/medium severity, yellow for low, and green if no issue is detected.

On standby server, you will get this message: Server is in standby mode!