TestManager

API test management and automation solution

Table of Contents

I. Introduction

Part I Background

1. Challenge
2. Goal
3. Benefits

Increase of API Defect Costs

- ✓ Increased use of Open APIs increases demand for **increased demand of open services**
- ✓ **Impact of API defect is expanded f rom inside the enterprise to outsi de**

Expansion of cloud service costs

- ✓ **The transformation of your existing On-premise envir onment to the cloud** is getting bigger.
- ✓ **Unoptimized API performance drives excessive cloud costs**

Inadequate Readiness of MSA architecture

- ✓ MSA architecture implementation **need API test management framework.**
- ✓ **Expansion of CI/CD leads to the need for automation of service t ests/performance tests**

“ Need to build up an integrated management system for API Testing”

Enhancing corporate competitiveness through improvement of API management

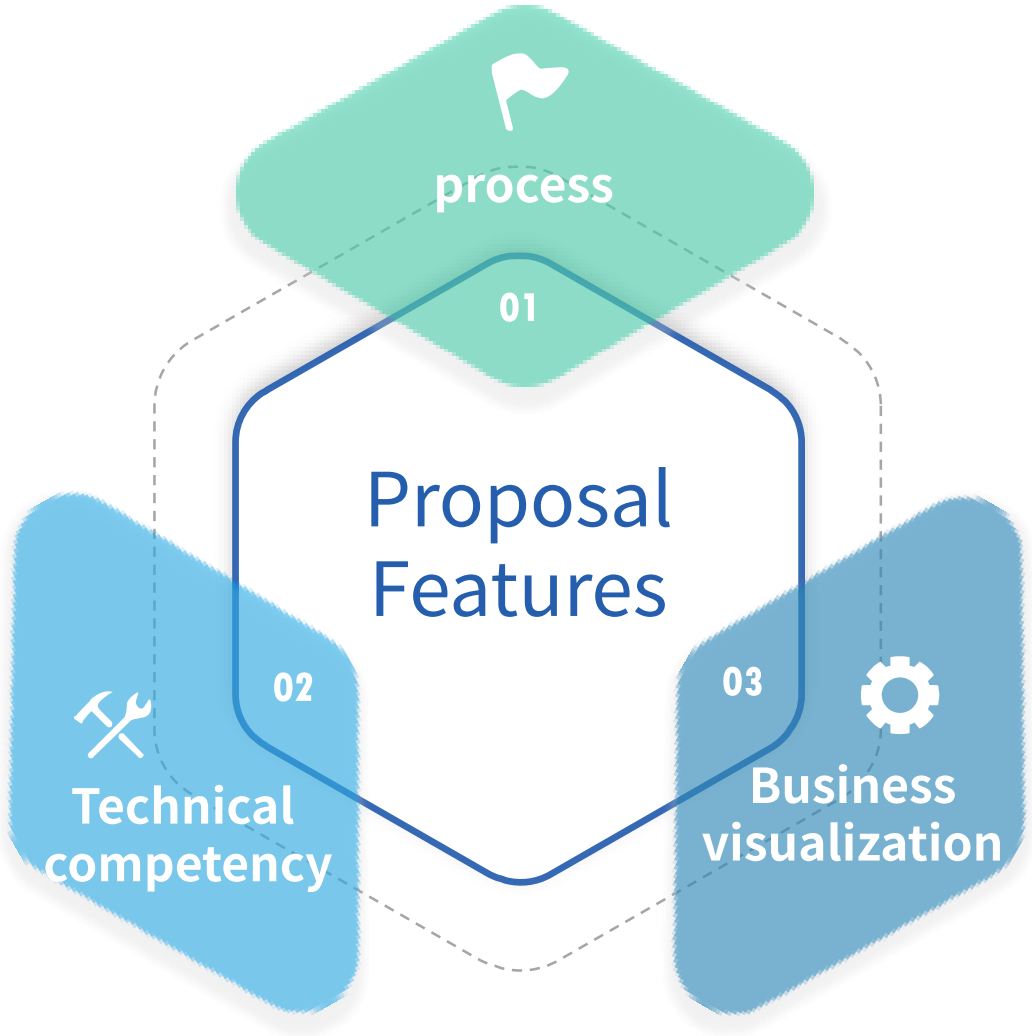
Increase API Quality Effectiveness

Expand your business visibility

API Test Management	API Test Automation	API monitoring
Establish a continuous test Management system ✓ Implement test management strategy ✓ Implement test management system	Establish a continuous functional/performance validation system ✓ Automate test script ceation (RESTful based) ✓ Automate tests execution ✓ Automate test results collection	Establish API monitoring system for key service ✓ Impletement Business enabled API monitoring system

Test management framework

Benefits



Process

- ✓ **Improvement of development productivity** by establishing API test management system
- ✓ **Automate 24x7 functional/performance validation**

Technical competency

- ✓ **Provide test results of the same quality** regardless of test personnel level
- ✓ **Assetization of testing know-how**

Business visualization

- ✓ **Implement business-linked API monitoring**

II. Proposed Solution

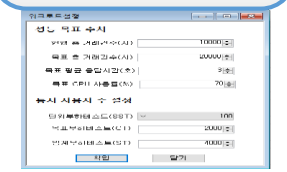
Part II

Proposed Solution

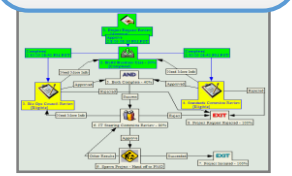
1. Solution Overview
2. Main Features
3. To-Be Image

TestManager is a solution that provides test management and test automation across areas from API test planning to end of test.

Test planning



Process Automation



Test Status Report

test plan | test design | test run | defect mgmt | End of test

Dashboard/Reports/API Monitoring

- ✓ test / defect related KPI
- ✓ test / defect related reports
- ✓ API performance/availability status

Functional validation

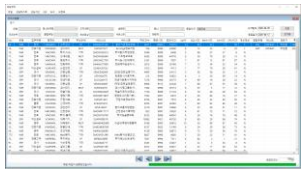
- •API spec auto-registration
- •Generate test script
- •Run Test
- •Validate test results
- •Manage defect

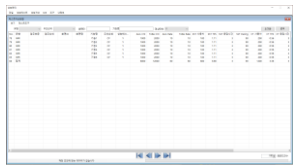
Perf. validation

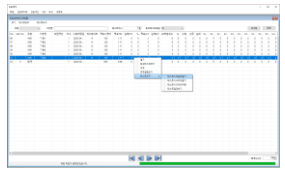
- •Generate test scripts automatically
- •Run performance tests
- •Validate test results
- •Manage defect

Solution Integration

- •Test tool (LR, JMeter)
- •Issue Mgmt(JIRA)
- •Notification(SMTP, SLACK, Line Messenger,etc)
- •Swagger API specification (JSON)







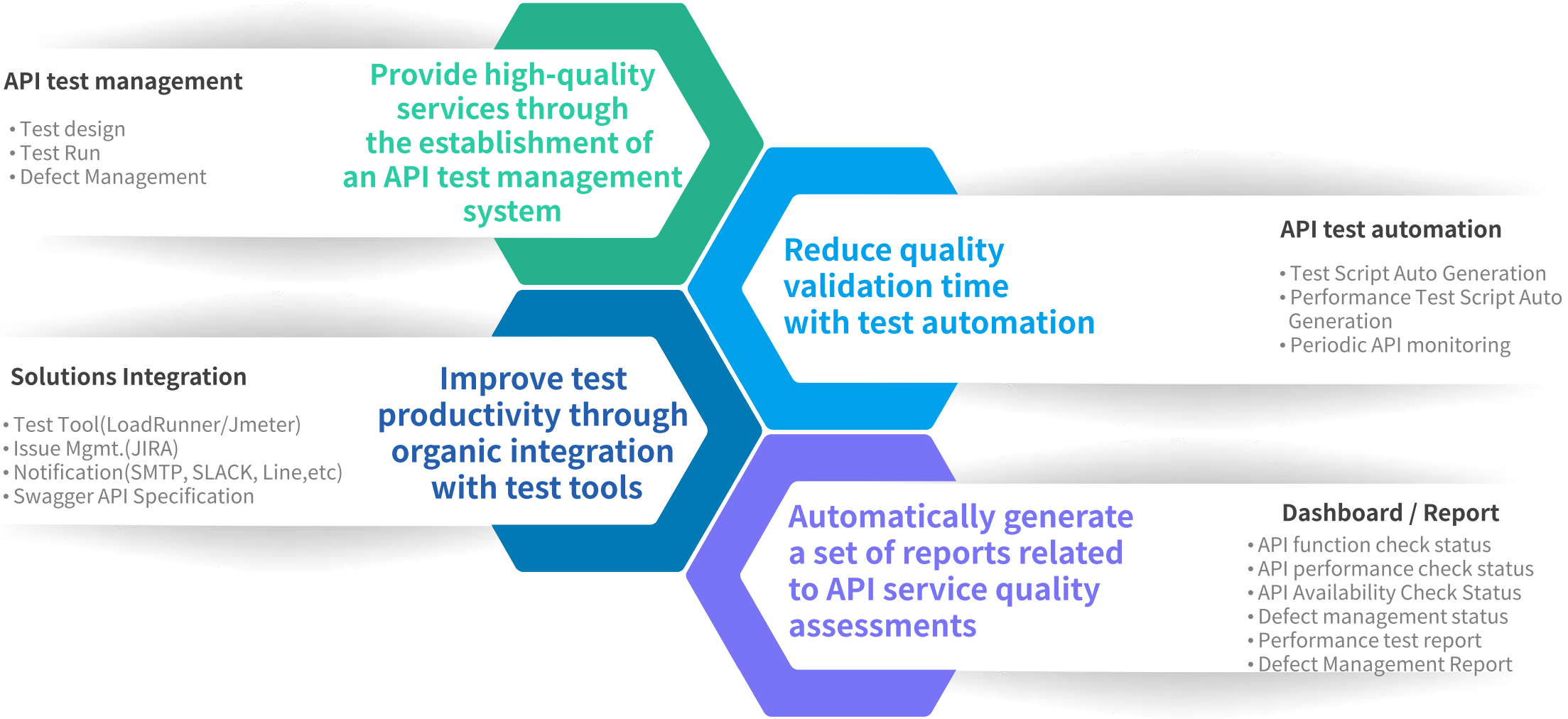
① enforce efficiency — **API Test Management System**

② enforce productivity — **API Test Automation**

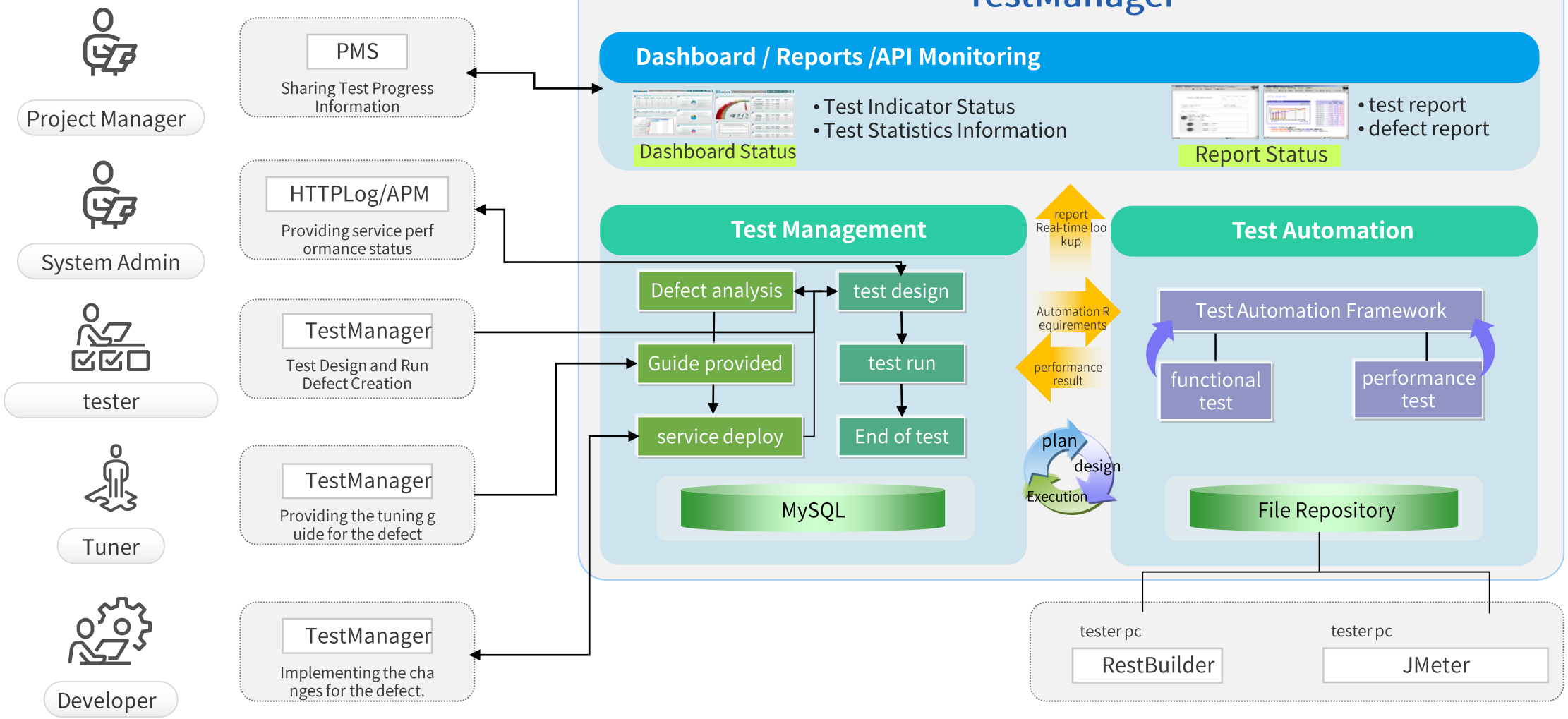
③ enforce scalability — **Solution Integration**

④ enforce visibility — **Dashboard / Report**

TestManager provides various features for key areas of API testing and can be customized to m eet customer requirements .



Maximize API quality visibility and business enabled monitoring with the introduction of TestManager.

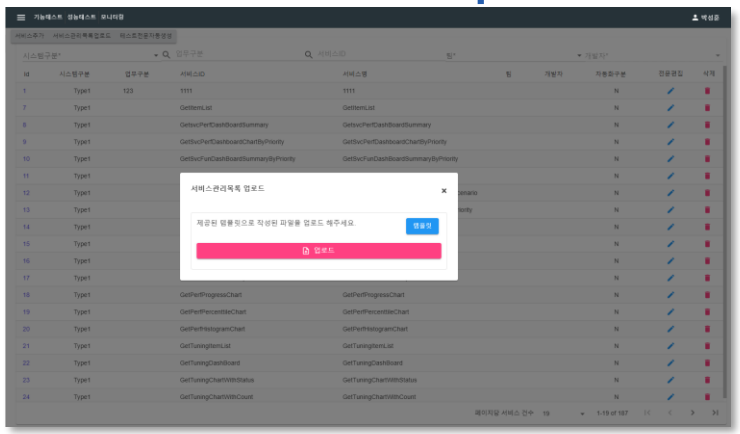


III. Features

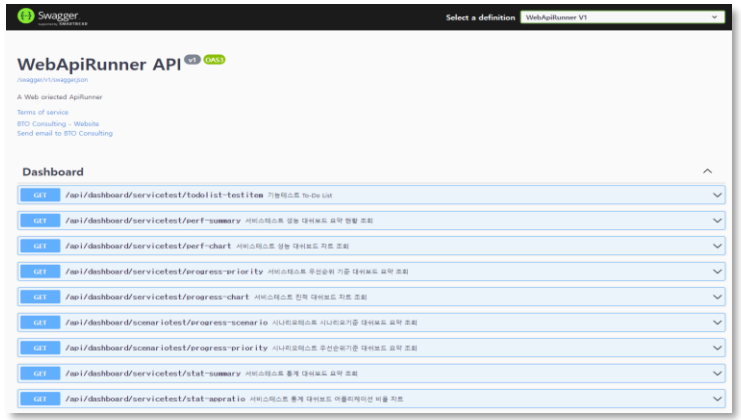
Part III Solution Features

1. Build API ServiceTest management system
2. Build API ScenarioTest management system
3. Build API PerformanceTest manage ment system

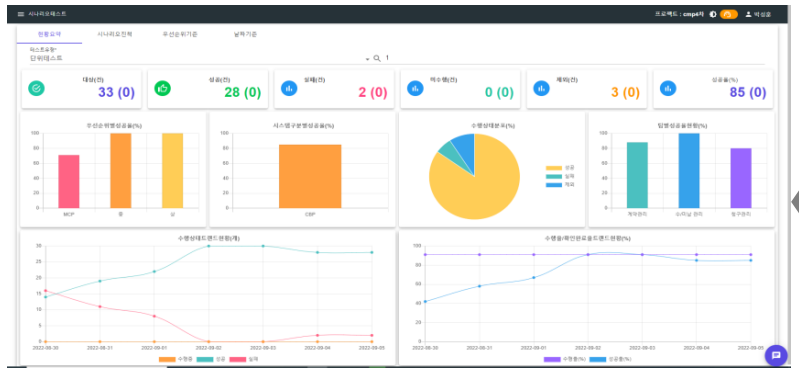
In addition to API management information, API specification-based test case management and automatic switching of performance test scripts supports overall testing activities, from functional validation to performance validation.



Upload API management sheet file (Excel)



Automatic creation of test script based on Swagger



Run tests and validate the results

Edit test script and pre-validation

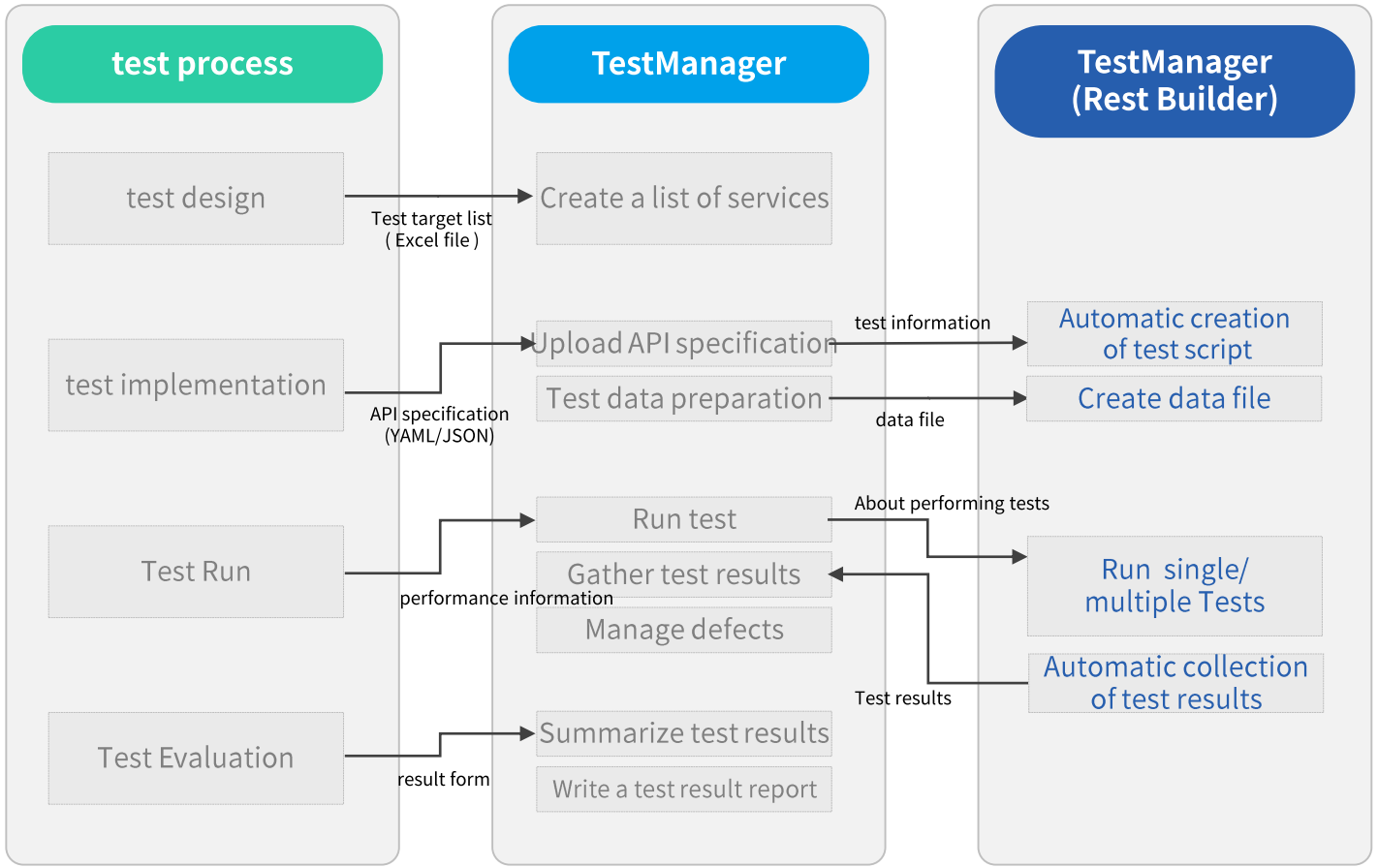
❖ Main Features

- ✓ API Management
- ✓ Automatic creation of spec-based test text
- ✓ Write test cases and perform tests
- ✓ Automatic tracking of test results and History management

❖ Benefit

- ✓ understand API test status
- ✓ Build up API test management system

The test script is automatically generated through the Swagger-based API specification, and functional validations are automated through the built-in RestBuilder.



❖ Main Features

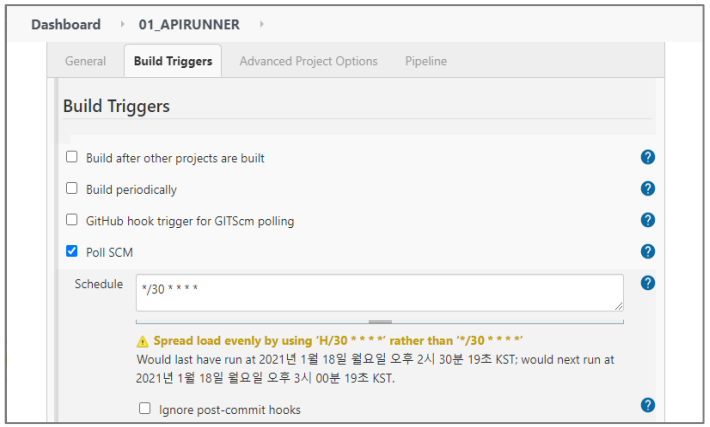
- ✓ API Management Document-Based Service Definitions
- ✓ Ceate test script based on API specifications
- ✓ Automating test execution and collection of test results

❖ Benefit

✓Establish an efficient API test management scheme

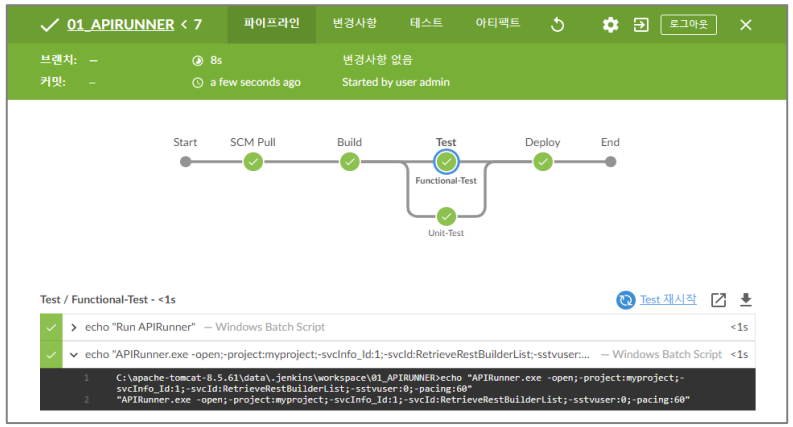
Auto-generated test script can also be used for CI/CD integration through Jenkins or its own scheduler.

✓ TestManager / Jenkins Pipeline Integration configuration

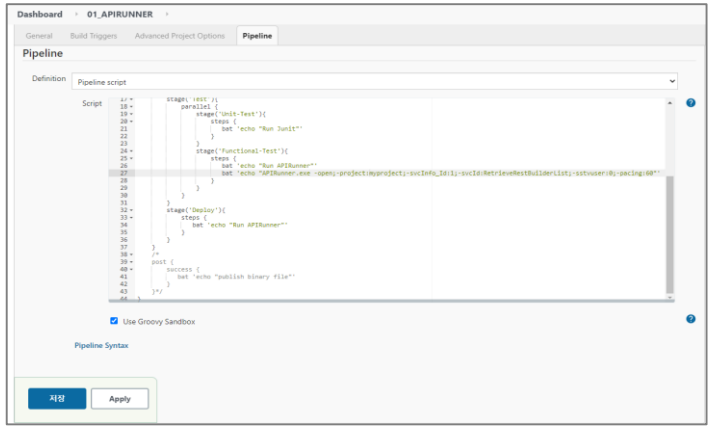


Jenkins existing job configuration

CI/CD Task Pipeline execution

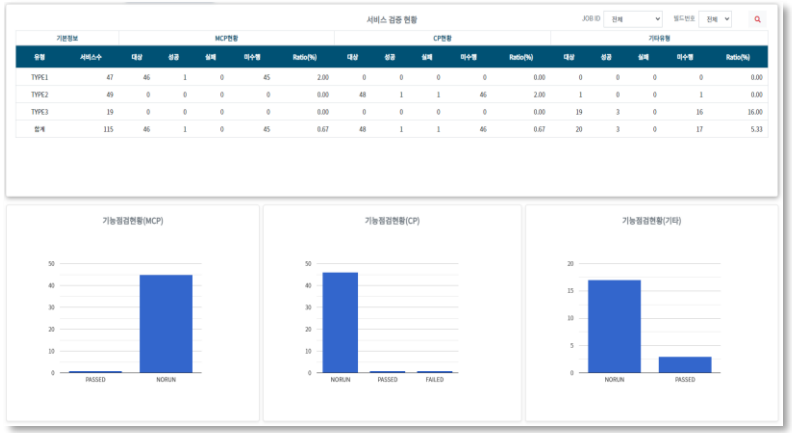


Pipeline Execution Result



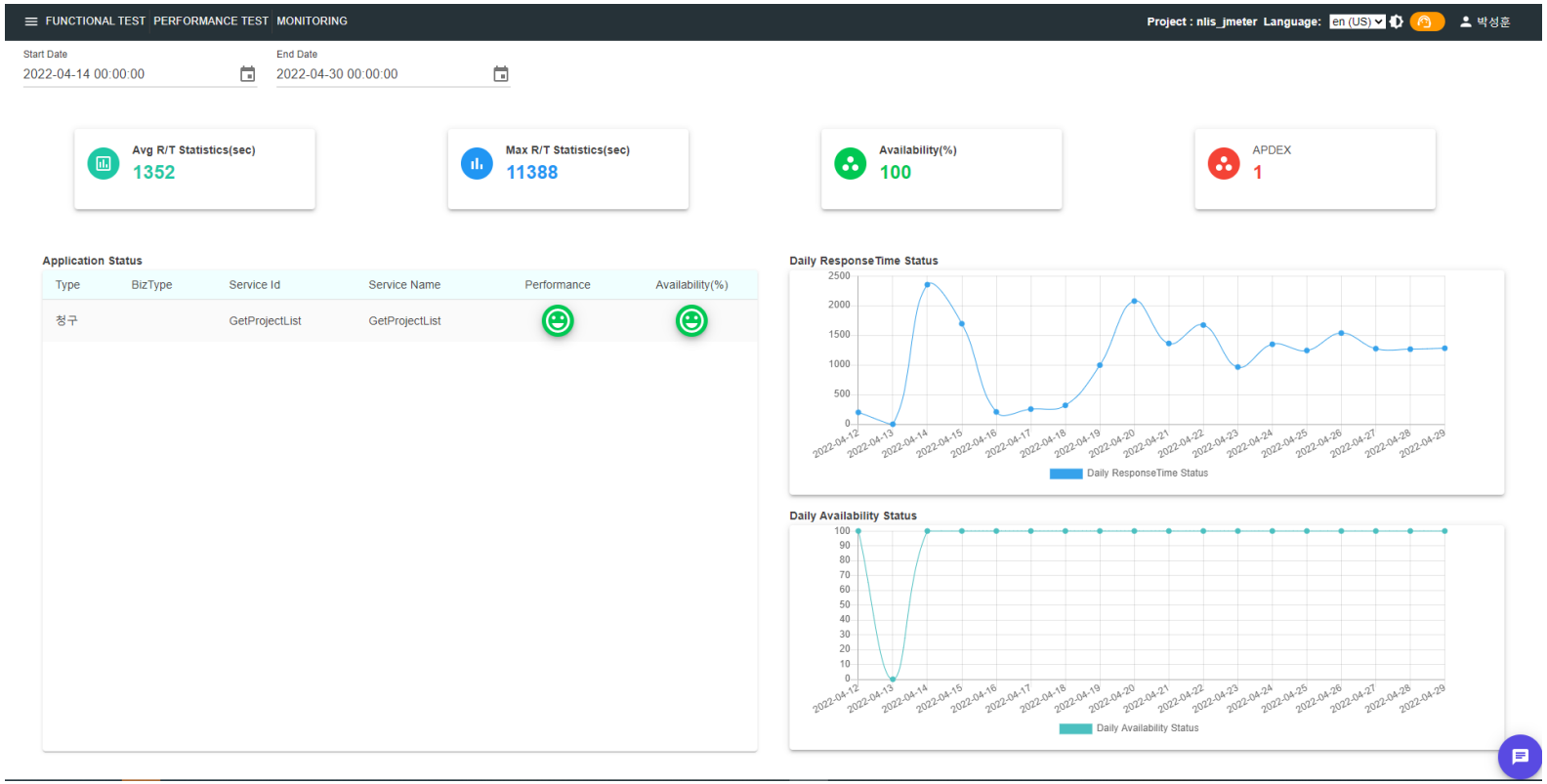
TestManager configuration to Test stage in Pipeline

(API target list and test environment setting)



test status based on build information

Key APIs are performed periodically through their own scheduler to provide business visibility information (performance/availability).



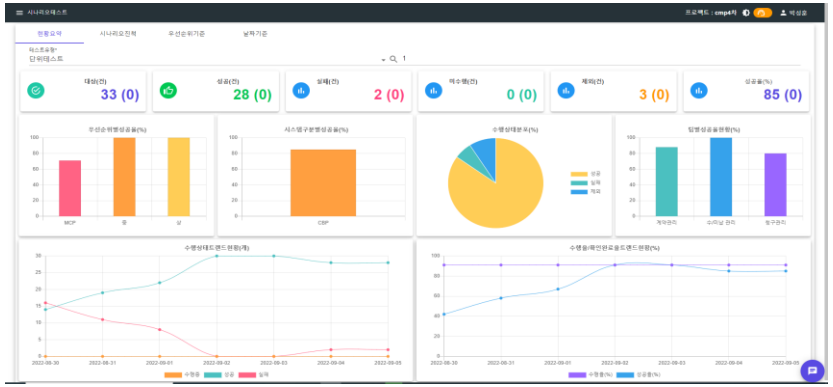
❖ Main Features

✓ weekly / daily status of

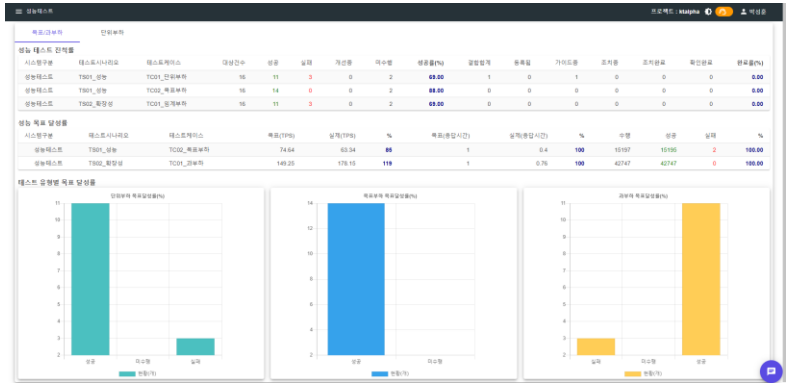
❖ Benefit

✓Provide business availability

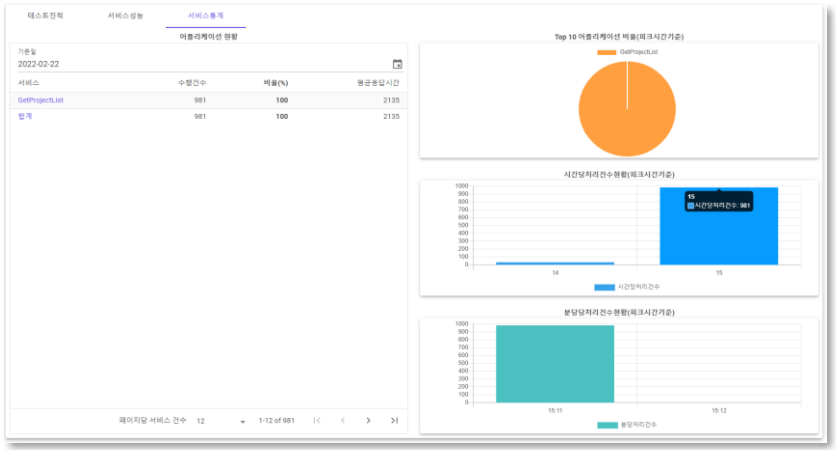
Provides real-time dashboards based on key statuses related to testing.



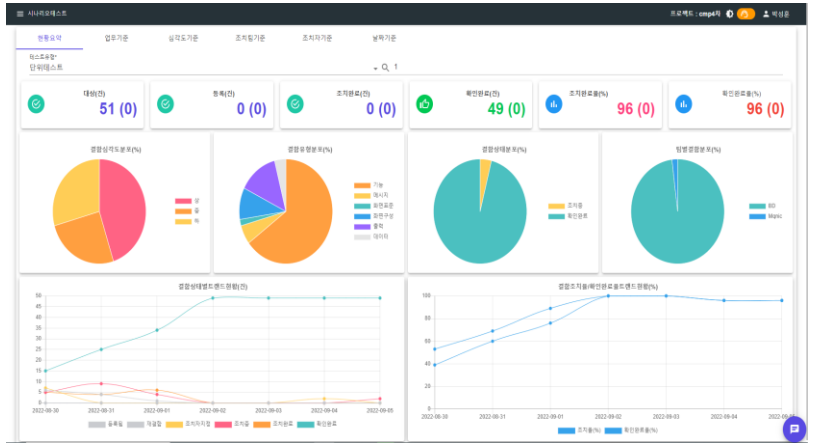
Functional Test Status



Performance Test Status



Service Status



Defect Status

❖ Main Features

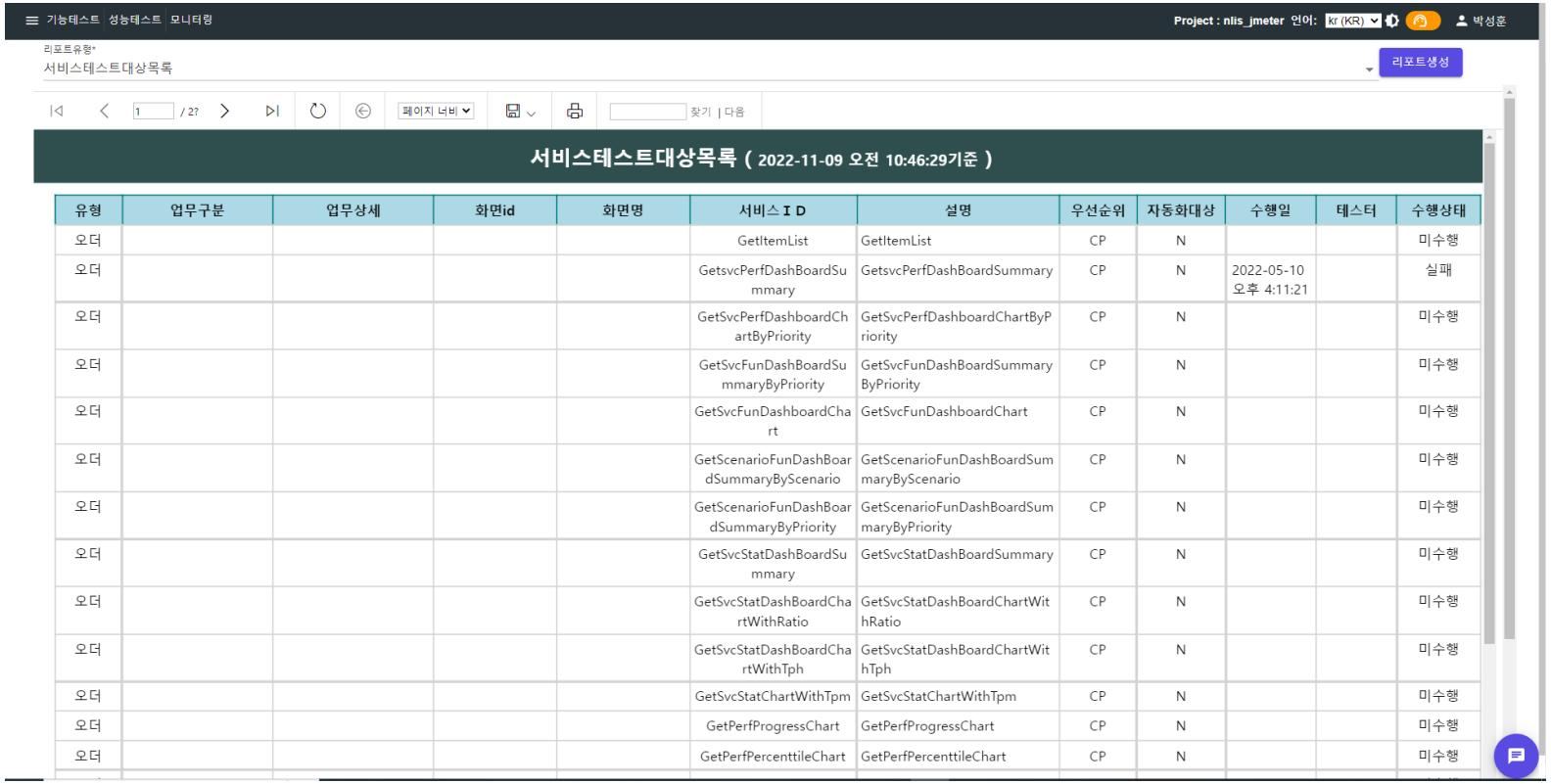
✓ Provide information on the status of various tests

❖ Benefit

✓Establish a real-time test evaluation system

Provides real-time reports based on key statuses related to testing.

✓ The report can be customized to meet the customer's requirements .



서비스테스트대상목록 (2022-11-09 오전 10:46:29기준)

유형	업무구분	업무상세	화면id	화면명	서비스ＩＤ	설명	우선순위	자동화대상	수행일	테스터	수행상태
오더					GetItemList	GetItemList	CP	N			미수행
오더					GetsvcPerfDashBoardSummary	GetsvcPerfDashBoardSummary	CP	N	2022-05-10 오후 4:11:21		실패
오더					GetSvcPerfDashboardChartByPriority	GetSvcPerfDashboardChartByPriority	CP	N			미수행
오더					GetSvcFunDashBoardSummaryByPriority	GetSvcFunDashBoardSummaryByPriority	CP	N			미수행
오더					GetSvcFunDashboardChart	GetSvcFunDashboardChart	CP	N			미수행
오더					GetScenarioFunDashBoardSummaryByScenario	GetScenarioFunDashBoardSummaryByScenario	CP	N			미수행
오더					GetScenarioFunDashBoardSummaryByPriority	GetScenarioFunDashBoardSummaryByPriority	CP	N			미수행
오더					GetSvcStatDashBoardSummary	GetSvcStatDashBoardSummary	CP	N			미수행
오더					GetSvcStatDashBoardChartWithRatio	GetSvcStatDashBoardChartWithRatio	CP	N			미수행
오더					GetSvcStatDashBoardChartWithTph	GetSvcStatDashBoardChartWithTph	CP	N			미수행
오더					GetSvcStatChartWithTpm	GetSvcStatChartWithTpm	CP	N			미수행
오더					GetPerfProgressChart	GetPerfProgressChart	CP	N			미수행
오더					GetPerfPercenttileChart	GetPerfPercenttileChart	CP	N			미수행

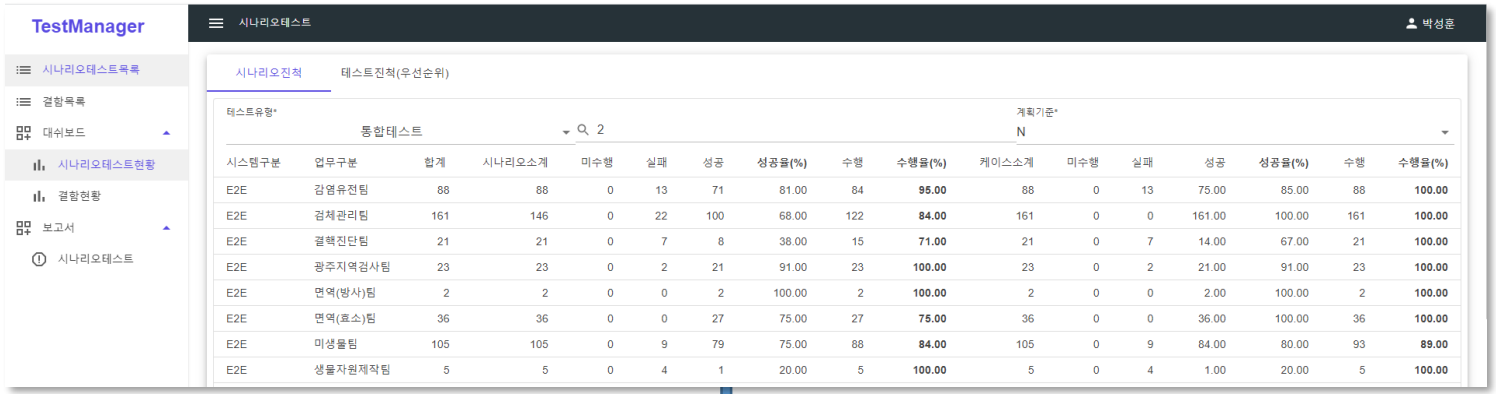
❖ Main Features

✓ Provide information on the status of various tests

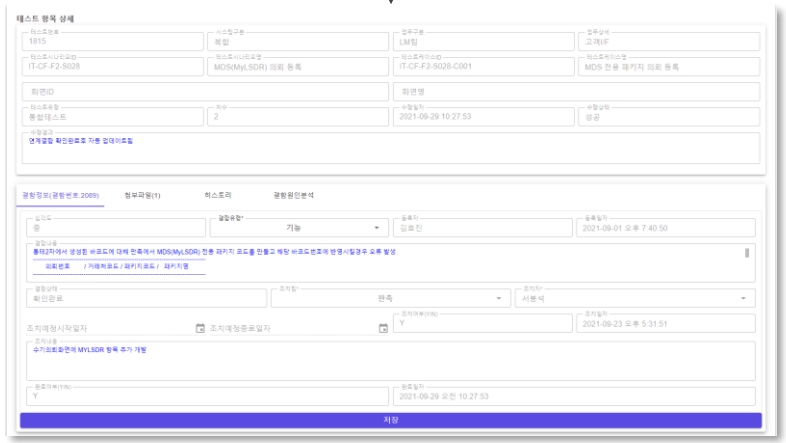
❖ Benefit

✓Establish a real-time test evaluation system

Enables systematic management of scenario-based unit/integration/system/acquisition tes ting activities after the service test phase.



Test case management



Defect management

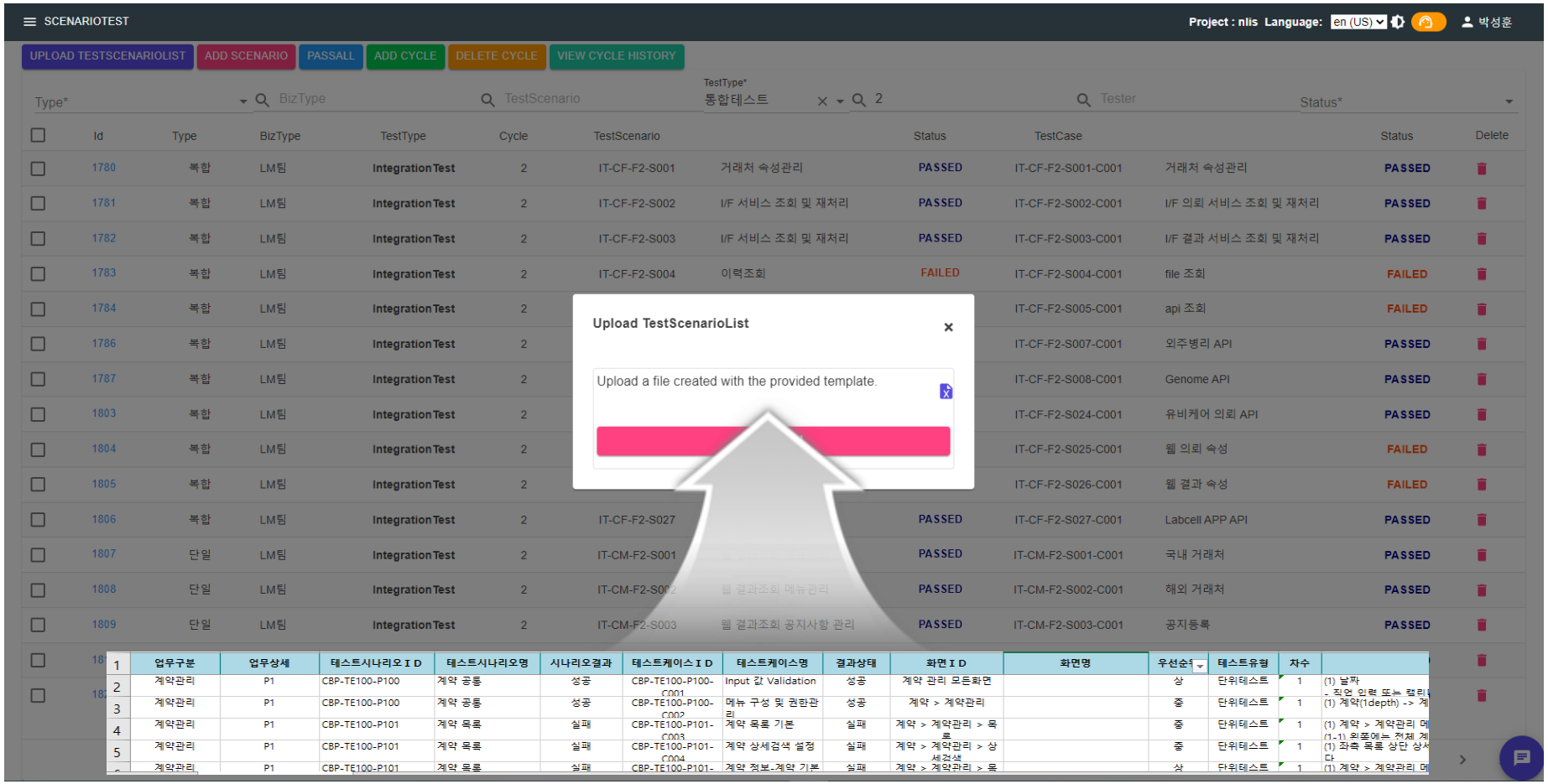
❖ Main Feat

- ✓ Scenario-based test case (unit / integration / system / acquisi tion) management
- ✓ Defect management
- ✓ Provide test status dashboard
- ✓ Provide test status report

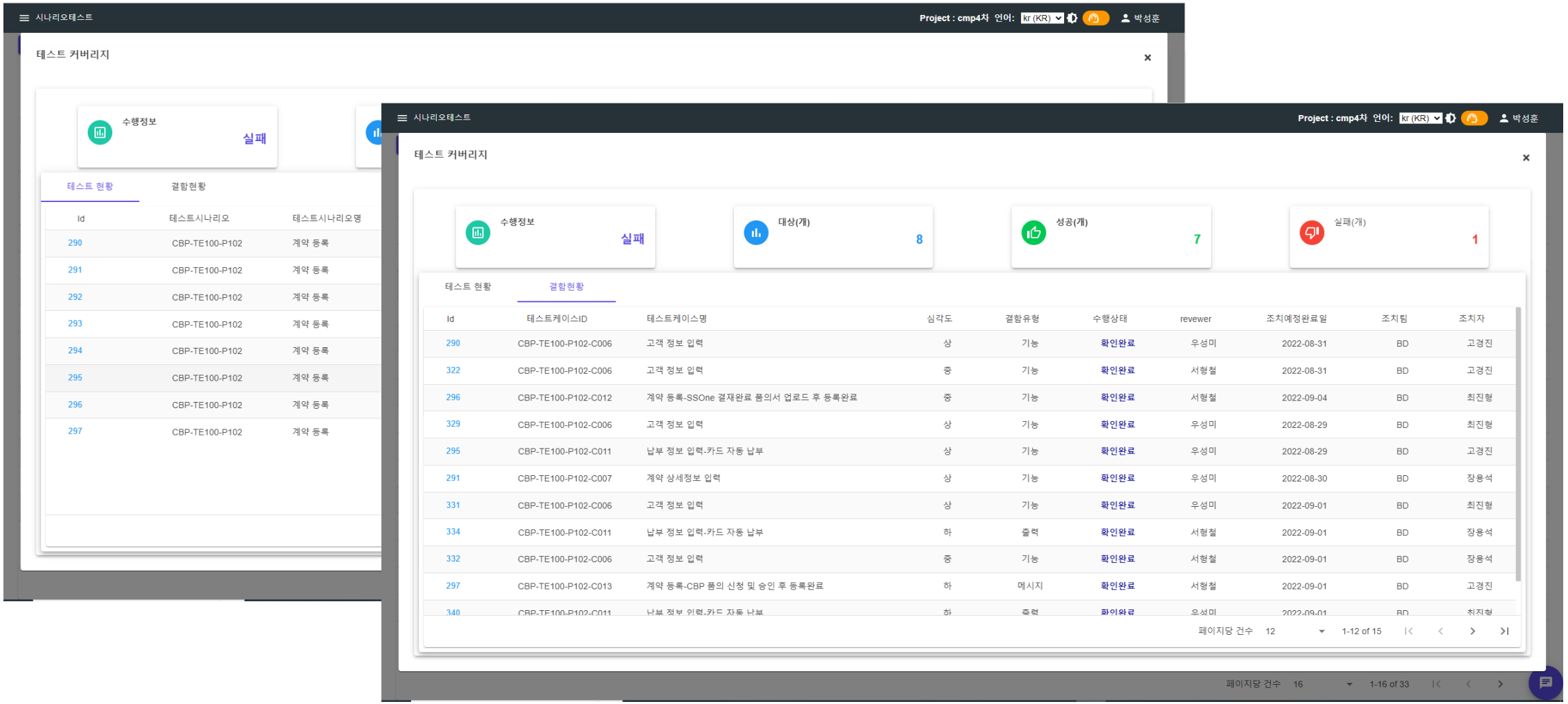
❖ Benefit

✓Improved feature quality and visibility

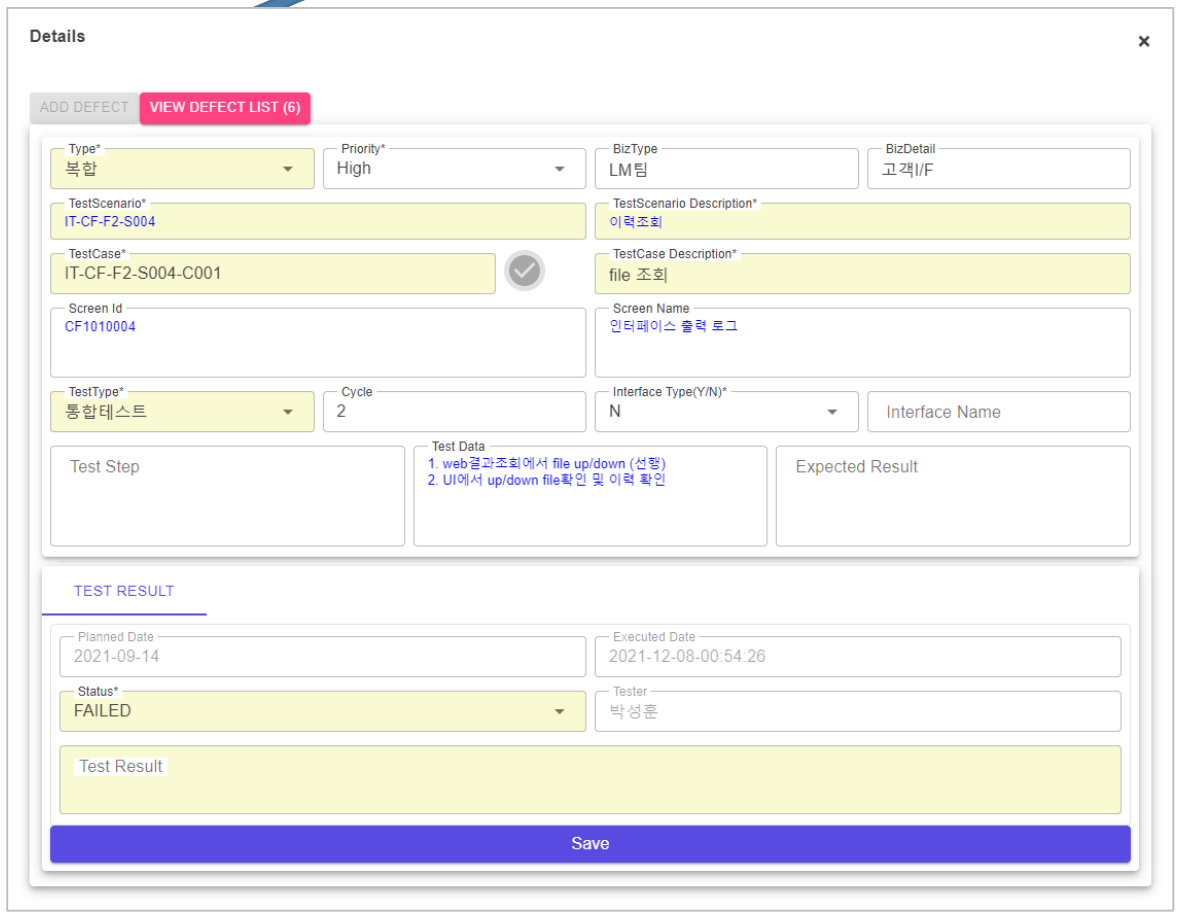
You can configure test scenarios by uploading a list of test scenarios created based on a pre-prepared Excel template..

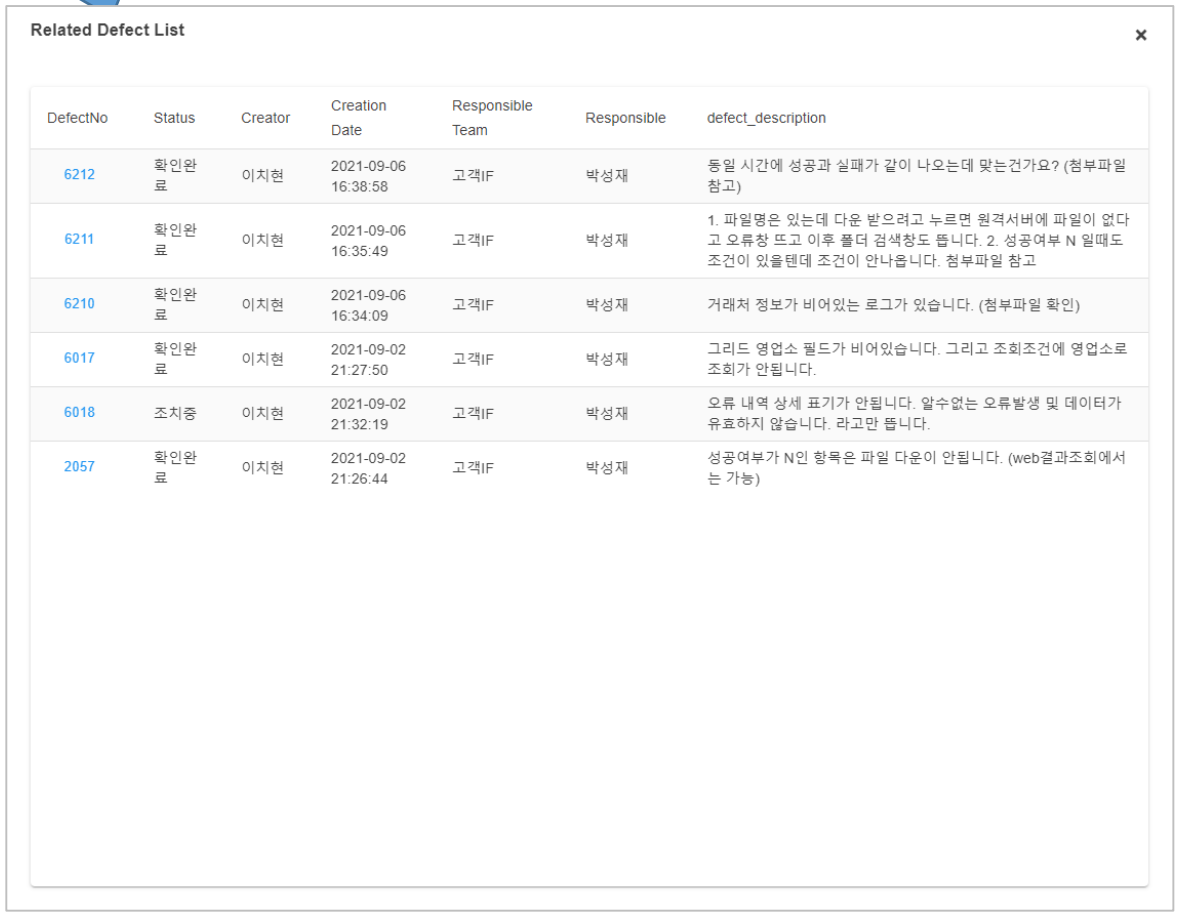


The test scenario coverage analysis provides an intuitive view of the associated test case status and fault status.

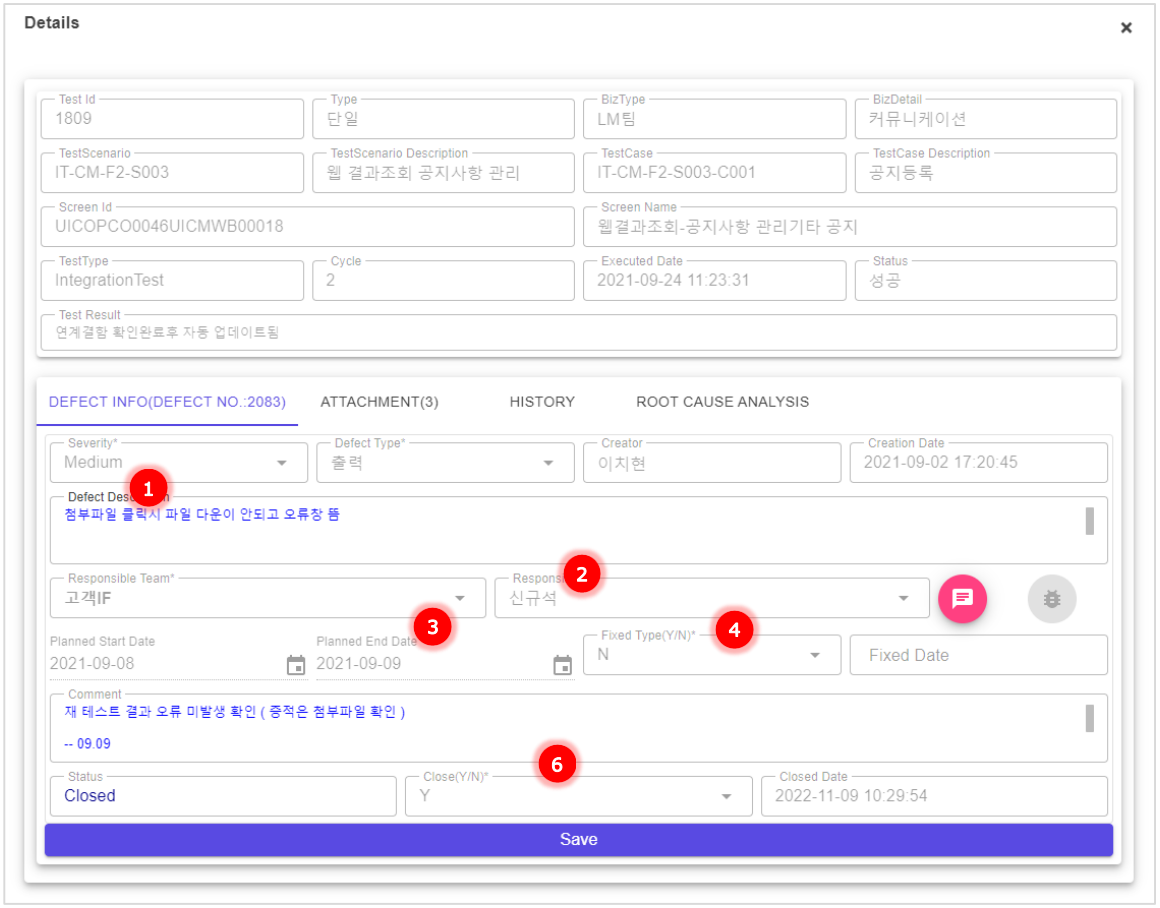


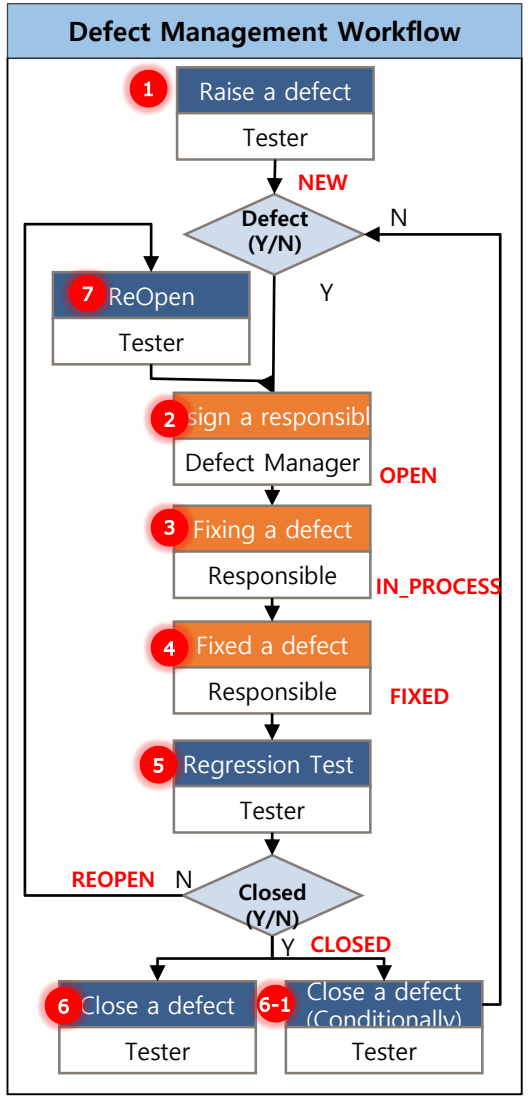
Defects related to the test case can be checked on one screen, and you can move it directly through the link.



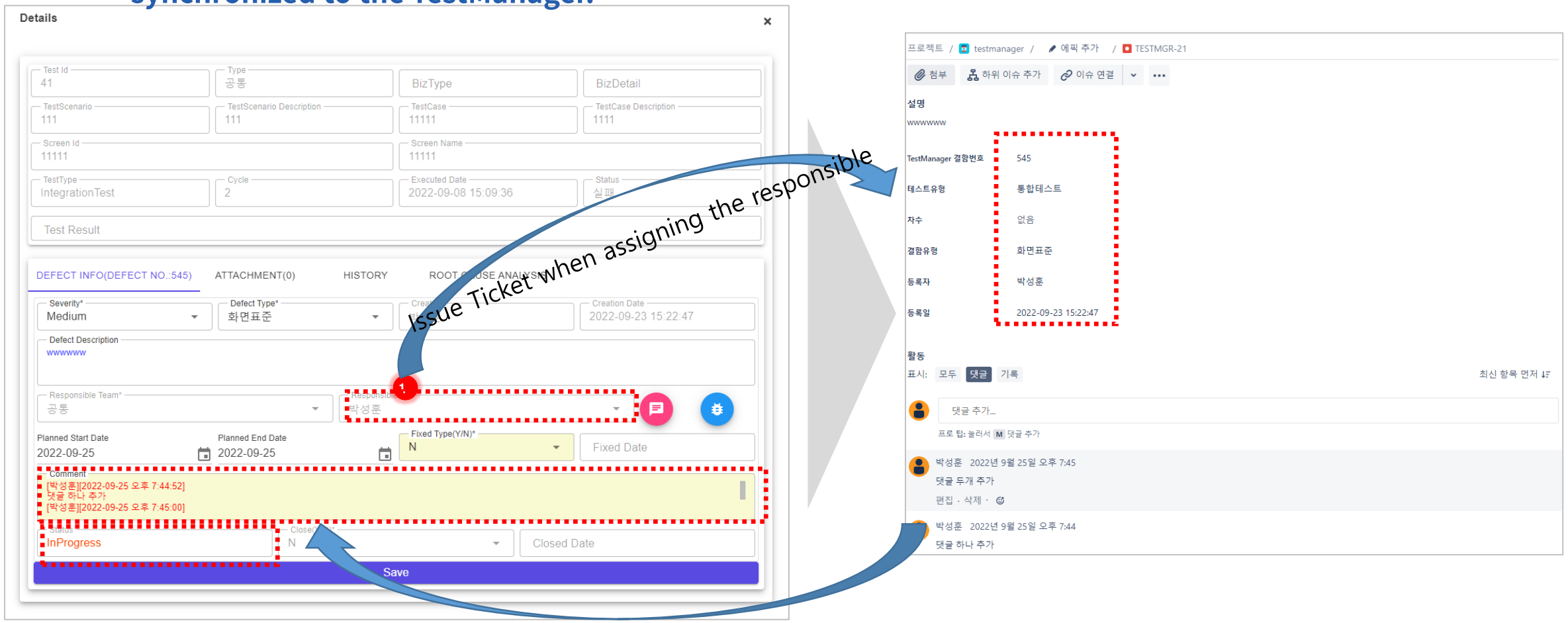


Built-in best practice-based defect management workflows enable efficient defect management with little customization.



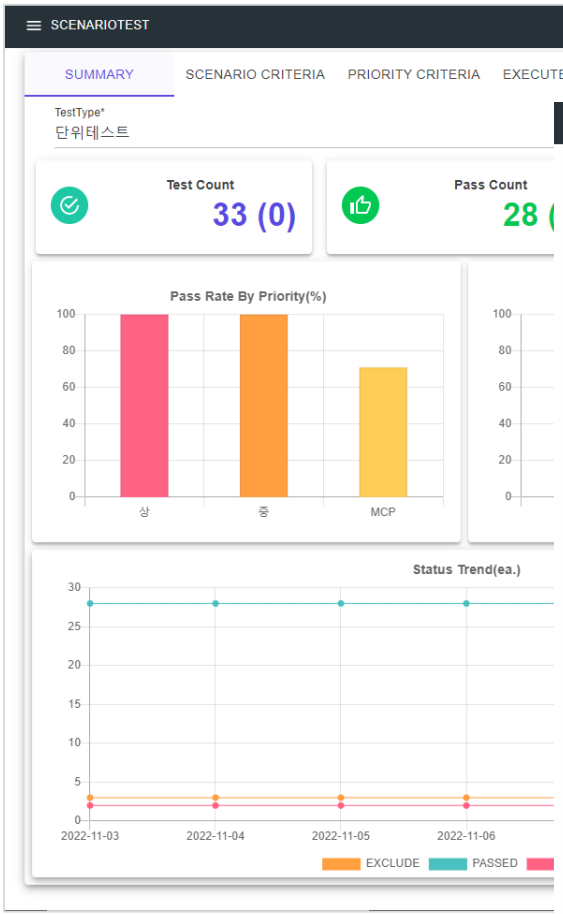


If integrated with the Jira system, the Jira ticket is automatically issued based on the defect details when assigning the defect responsible, and the changes made on the Jira system are automatically synchronized to the TestManager.

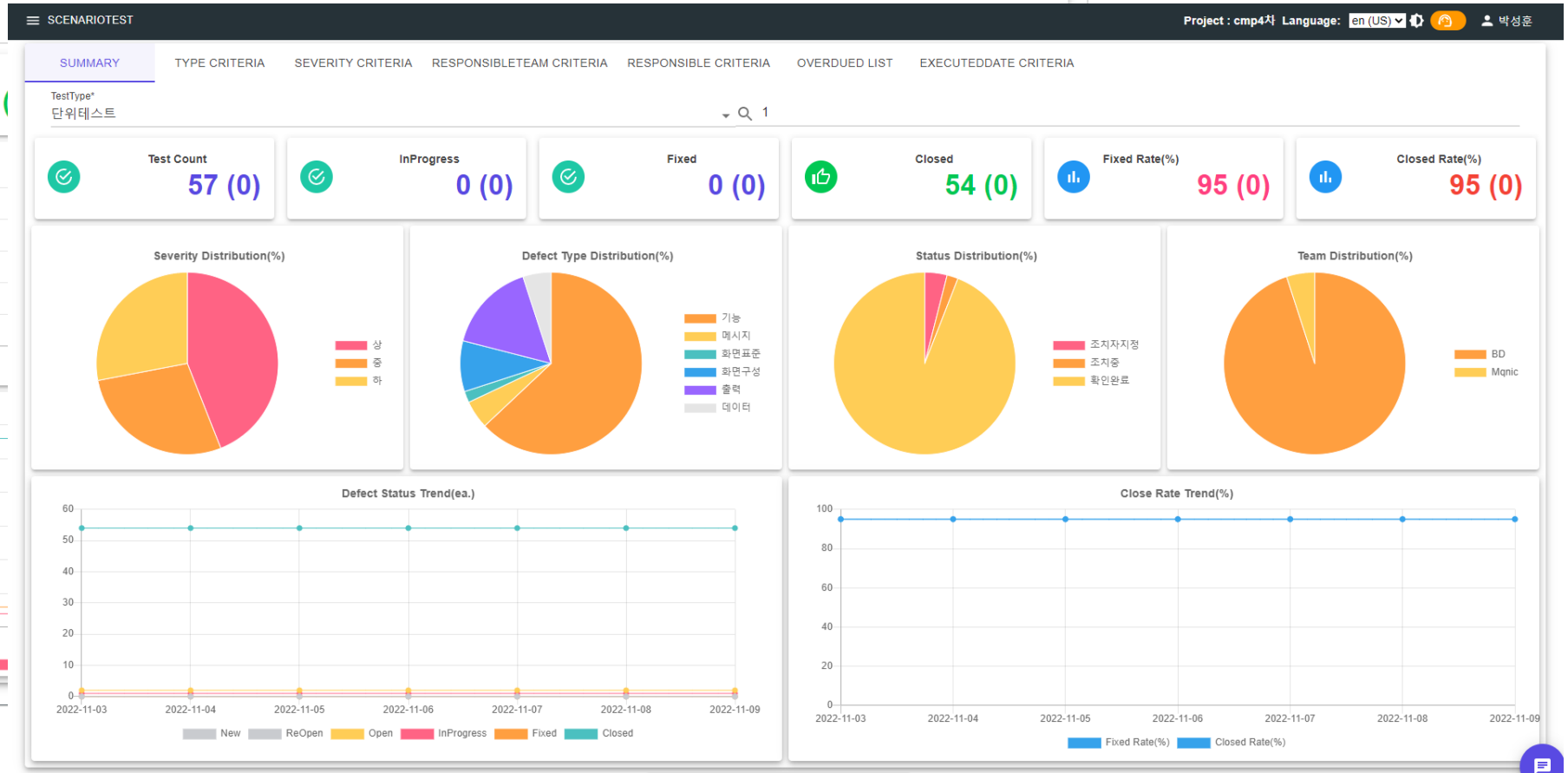


On defect fixed, the changes are synchronized with the TestManager

Provides real-time dashboards based on key statuses related to testing.



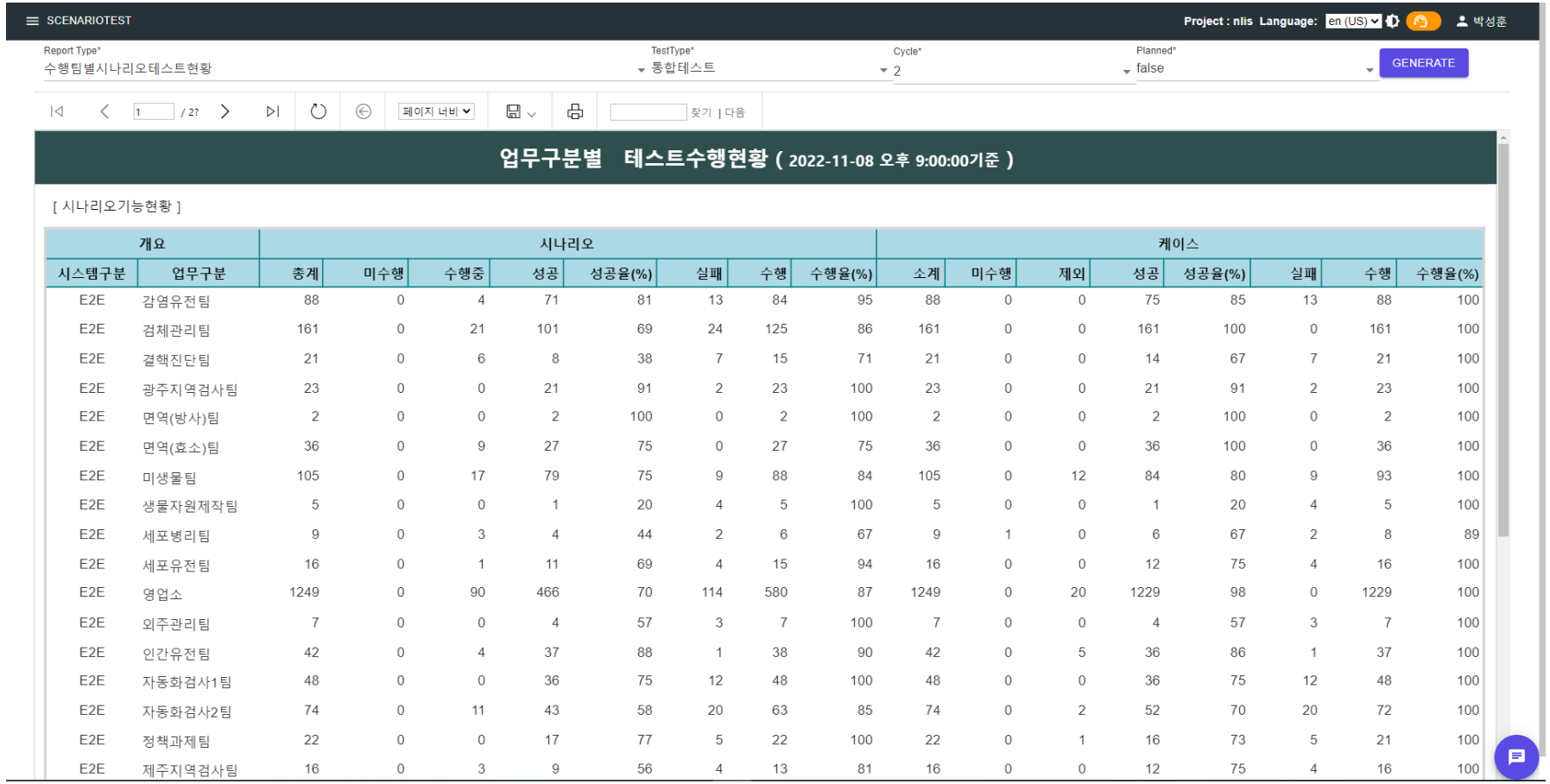
Test Status



Defect Status

Provides real-time reports based on key statuses related to testing.

✓ The report can be customized to meet the customer's requirements .



개요		시나리오								케이스							
시스템구분	업무구분	총계	미수행	수행중	성공	성공율(%)	실패	수행	수행율(%)	소계	미수행	제외	성공	성공율(%)	실패	수행	수행율(%)
E2E	감염유전팀	88	0	4	71	81	13	84	95	88	0	0	75	85	13	88	100
E2E	검체관리팀	161	0	21	101	69	24	125	86	161	0	0	161	100	0	161	100
E2E	결핵진단팀	21	0	6	8	38	7	15	71	21	0	0	14	67	7	21	100
E2E	광주지역검사팀	23	0	0	21	91	2	23	100	23	0	0	21	91	2	23	100
E2E	면역(방사)팀	2	0	0	2	100	0	2	100	2	0	0	2	100	0	2	100
E2E	면역(효소)팀	36	0	9	27	75	0	27	75	36	0	0	36	100	0	36	100
E2E	미생물팀	105	0	17	79	75	9	88	84	105	0	12	84	80	9	93	100
E2E	생물자원제작팀	5	0	0	1	20	4	5	100	5	0	0	1	20	4	5	100
E2E	세포병리팀	9	0	3	4	44	2	6	67	9	1	0	6	67	2	8	89
E2E	세포유전팀	16	0	1	11	69	4	15	94	16	0	0	12	75	4	16	100
E2E	영업소	1249	0	90	466	70	114	580	87	1249	0	20	1229	98	0	1229	100
E2E	외주관리팀	7	0	0	4	57	3	7	100	7	0	0	4	57	3	7	100
E2E	인간유전팀	42	0	4	37	88	1	38	90	42	0	5	36	86	1	37	100
E2E	자동화검사1팀	48	0	0	36	75	12	48	100	48	0	0	36	75	12	48	100
E2E	자동화검사2팀	74	0	11	43	58	20	63	85	74	0	2	52	70	20	72	100
E2E	정책과제팀	22	0	0	17	77	5	22	100	22	0	1	16	73	5	21	100
E2E	제주지역검사팀	16	0	3	9	56	4	13	81	16	0	0	12	75	4	16	100

Incremental validation strategy of performance test enables individual API verification, business verification, and system architecture and capacity acceptance in order.

성능 테스트 진척률

시스템구분	테스트시나리오	테스트케이스	대상건수	성공	실패	개선중	미수행	성공률(%)	결함합계	등록됨	가이드중	조치중	조치완료	확인완료	완료률(%)
성능테스트	TS01_성능	TC01_단위부하	30	28	1	0	1	**93.00**	12	0	0	1	0	11	**92.00**
성능테스트	TS01_성능	TC02_목표부하	25	19	2	0	3	**76.00**	2	0	0	0	0	2	**100.00**
성능테스트	TS02_확장성	TC01_임계부하	25	0	0	0	25	**0.00**	0	0	0	0	0	0	**0.00**

성능 목표 달성률

시스템구분	테스트시나리오	테스트케이스	목표(TPS)	실제(TPS)	%	목표(응답시간)	실제(응답시간)	%	수행	성공	실패	%
성능테스트	TS01_성능	TC02_목표부하	86.27	268.03	**310.69**	2.8	2.49	**112.61**	52744	52743	1	**100.00**
성능테스트	TS02_확장성	TC01_과부하	173.5	0	**0**	2.8	0	**280**	0	0	0	**0.00**

Test scenario	Test case	Test Purpose	metrics
Performance	Single Scenario Test	Check the maximum performance of each individual API	• Maximum TPS • Average response time at maximum TPS state • Average CPU utilization at maximum TPS state
	Compound Test	Check the performance of overall APIs under the peak-time load	• Rmaximum TPS • Average response time at traget TPS state • Average CPU utilization at target TPS state
Scalability	Scalability Test	Check maximum performance Under the maximum CPU	• Critical TPS versus

Performance Test Strategy

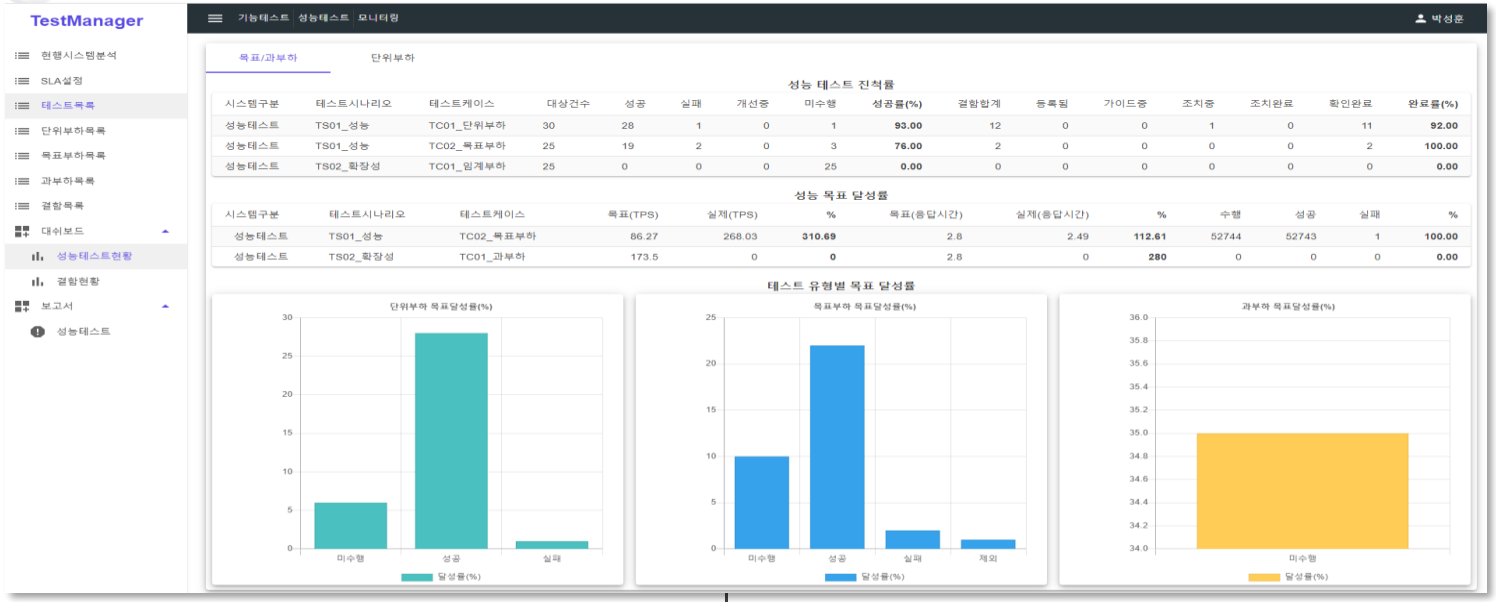
❖ Main Features

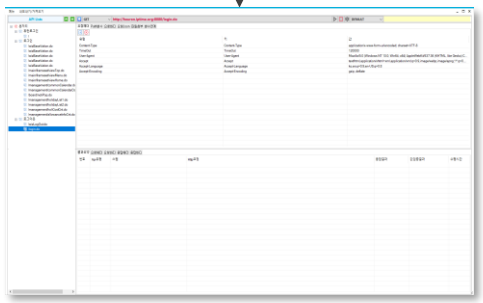
- ✓ Systematize a strategy for performing performance tests
- ✓ individual API performance validation (Developer)
- ✓ Peak load / overload validation (performance test expert)
- ✓ Real-time identification of performance test status

❖ Benefit

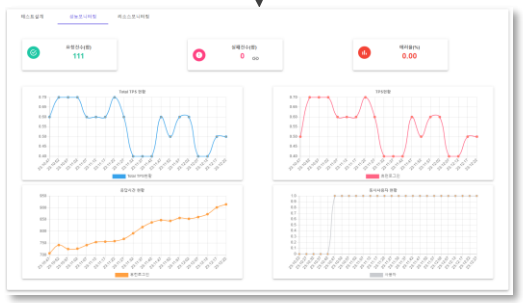
- ✓ Individual API performance validation
- ✓ System Architecture Verification
- ✓ Verification of system capacity adequacy

Integration with the performance test tool (Jmeter/LoadRunner) enables systematic management of various activities related to performance test activities.





Test script Integration



Real-time monitoring

Test results

❖ Main Features

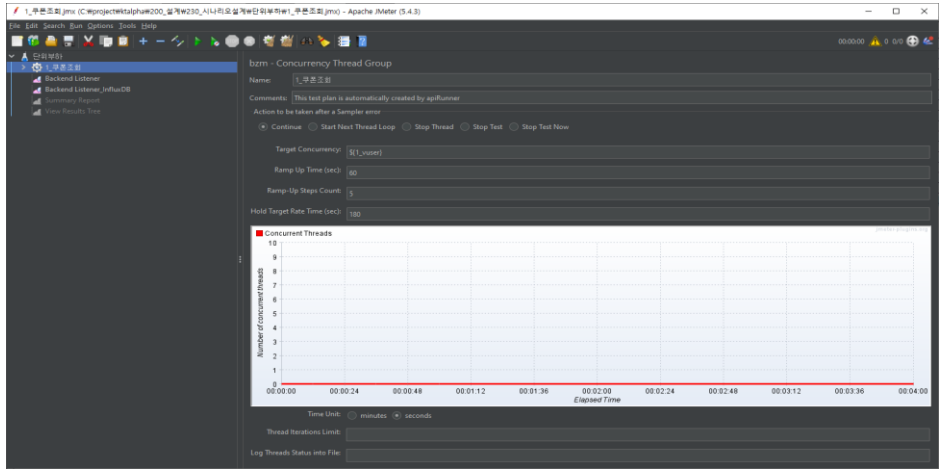
- ✓ Provide test script template
- ✓ Automatically create test scripts (with data)
- ✓ Real-time Test Satus Monitoring (Back Listener)
- ✓ Automating test execution and saving the execution result DB

❖ Benefit

✓Improving performance test quality

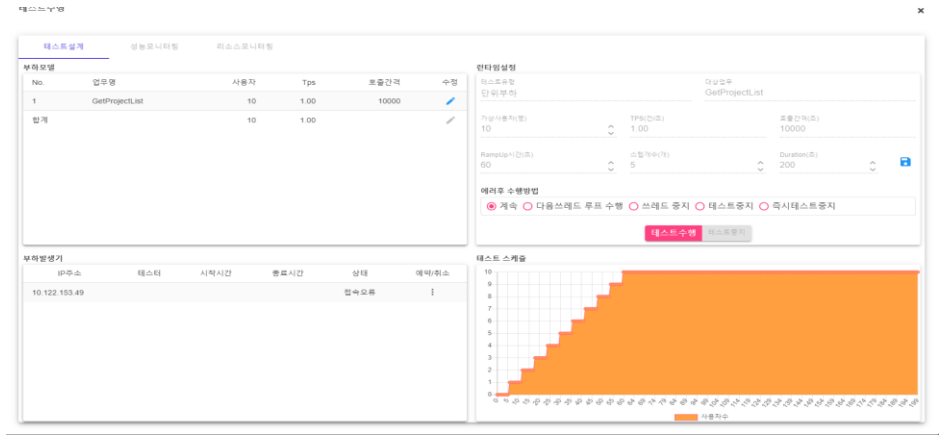
After functional test execution, test script can be used for performance validation purposes by conver ting into performance test scripts and linking them with performance test tools.

functional testing



performance test script

test result



test run

For API tests, the test script is written using the RestBuilder and can also be written directly from Jmeter.

PERFORMANCE TEST

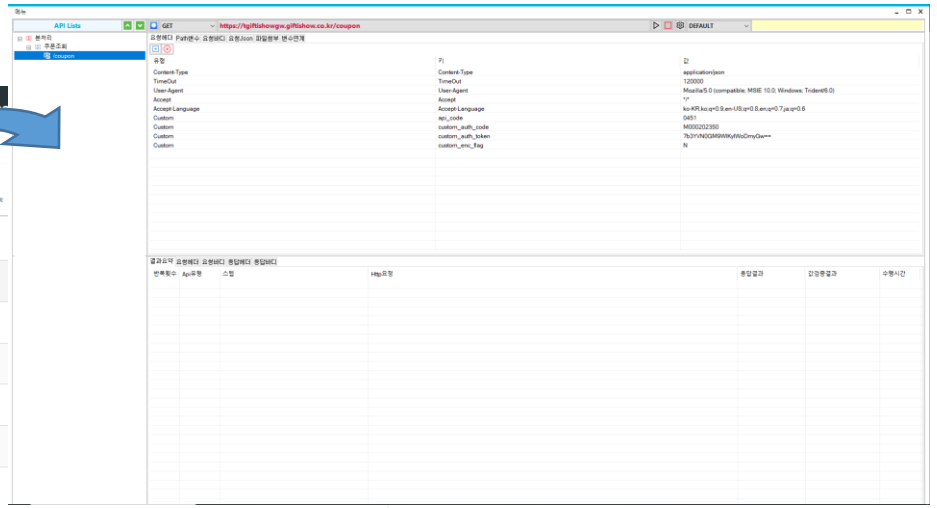
ADD TEST | UPLOAD TESTLIST

Type* | Automation Type* | UseType* Y | SstOnly*

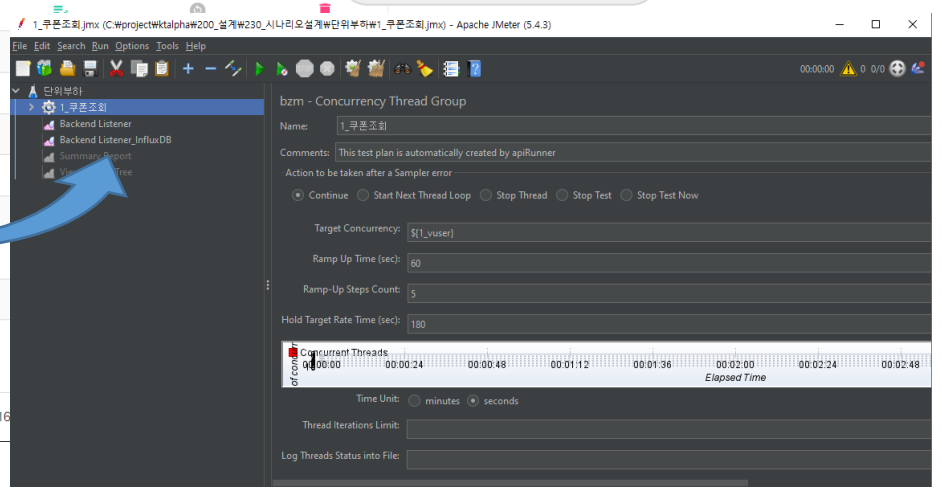
Id	Type	TestCase	Single Scenario Test TPS	Compound Test TPS	Scalability Test TPS	Automation Type
1	MGW	쿠폰조회	43.58	21.79	43.58	Y
2	MGW	원픽쿠폰조회	38.46	12.82	25.63	N
3	MGW	쿠폰상세조회	17.58	5.86	11.72	Y
5	MGW	상품리리스조회	8.49	2.83	5.65	Y
7	MGW	쿠폰발급	2.85	0.95	1.9	Y
9	MGW	쿠폰교환	0.88	0.44	0.88	Y
10	MGW	쿠폰상태조회	2.37	0.79	1.57	Y
11	MGW	쿠폰상품정보조회	3.42		2.29	Y
13	POS	쿠폰인증	4.06	03	4.06	Y
14	POS	쿠폰교환_POS	4.06		4.06	Y
15	POS	쿠폰교환취소	4.06	2.03	4.06	Y
16	BIZ	쿠폰조회_BIZ	60.9	20.3	40.59	Y
17	BIZ	비즈머니이력조회	1.56	0.52	1.04	Y
18	BIZ	쿠폰발급_BIZ	1.53	0.51	1.01	Y
19	MGW	쿠폰발급1	1.71	0.57	1.14	Y
20	TOTAL		195.51	74.61	149.18	

Rows per Page 16

RESTBUILDER | JMETER | CLOSE

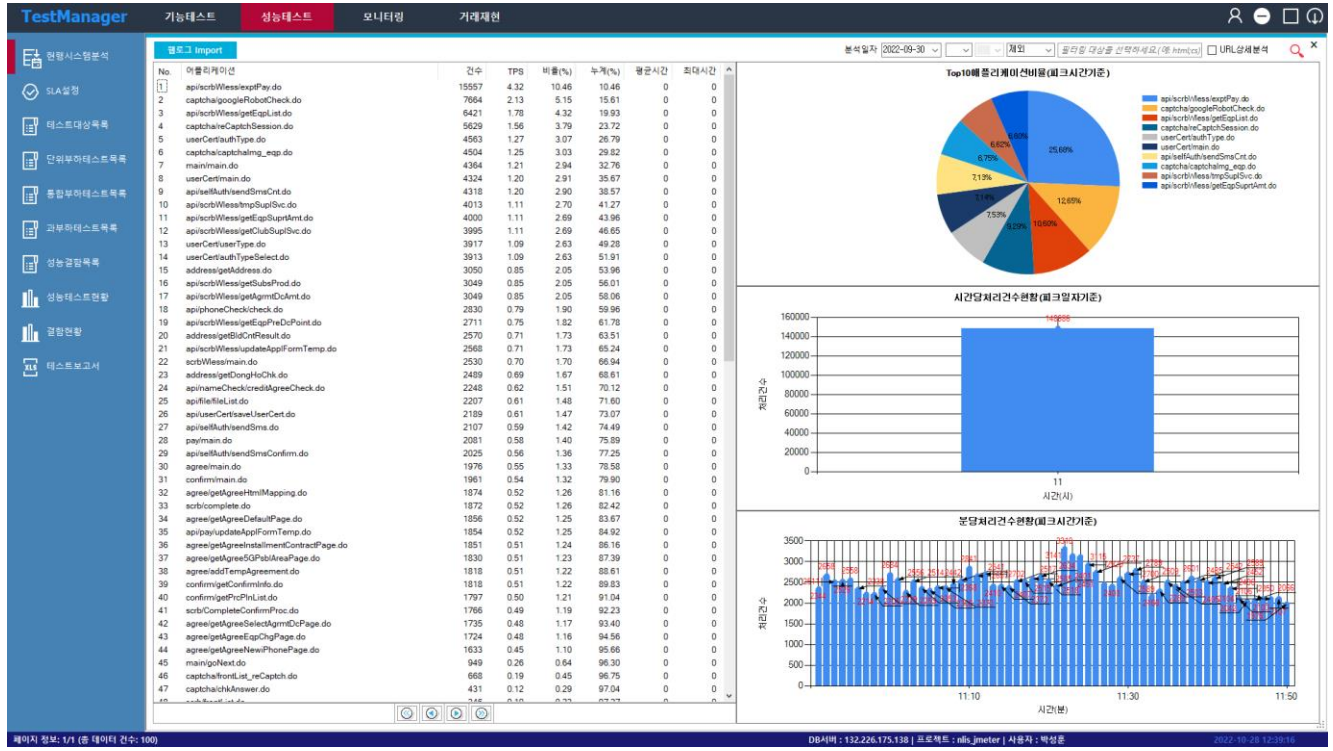


RestBuilder



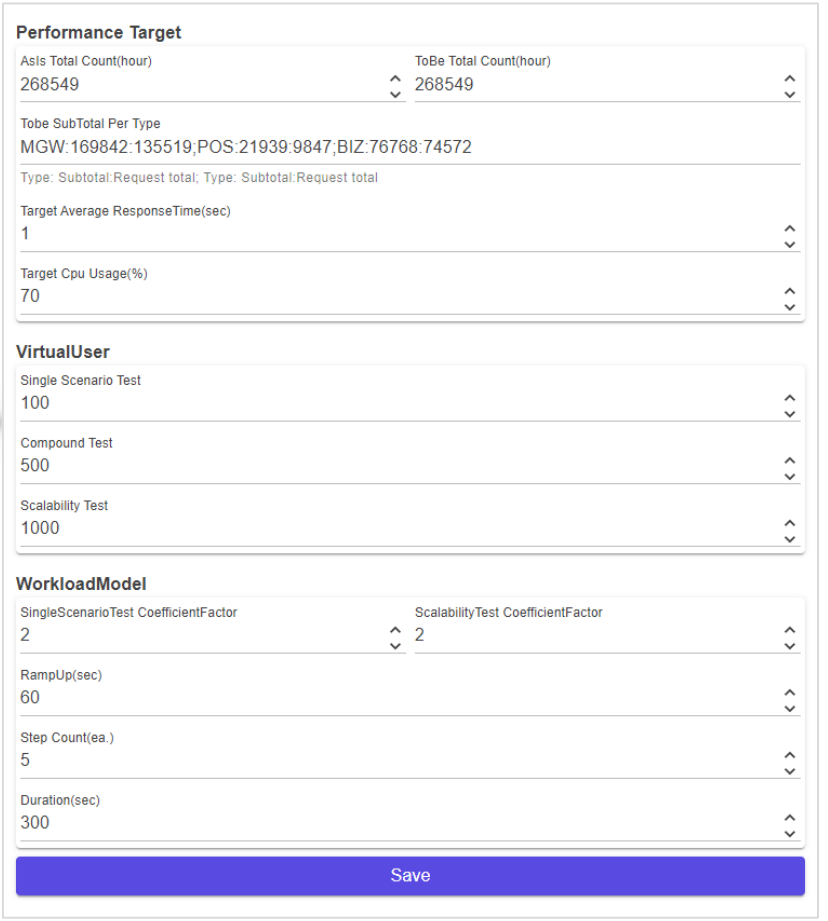
Jmter UI

It provides the As-Is system workload analysis based on the embedded weblog analysis tool to create a realistic load close to the actual situation.



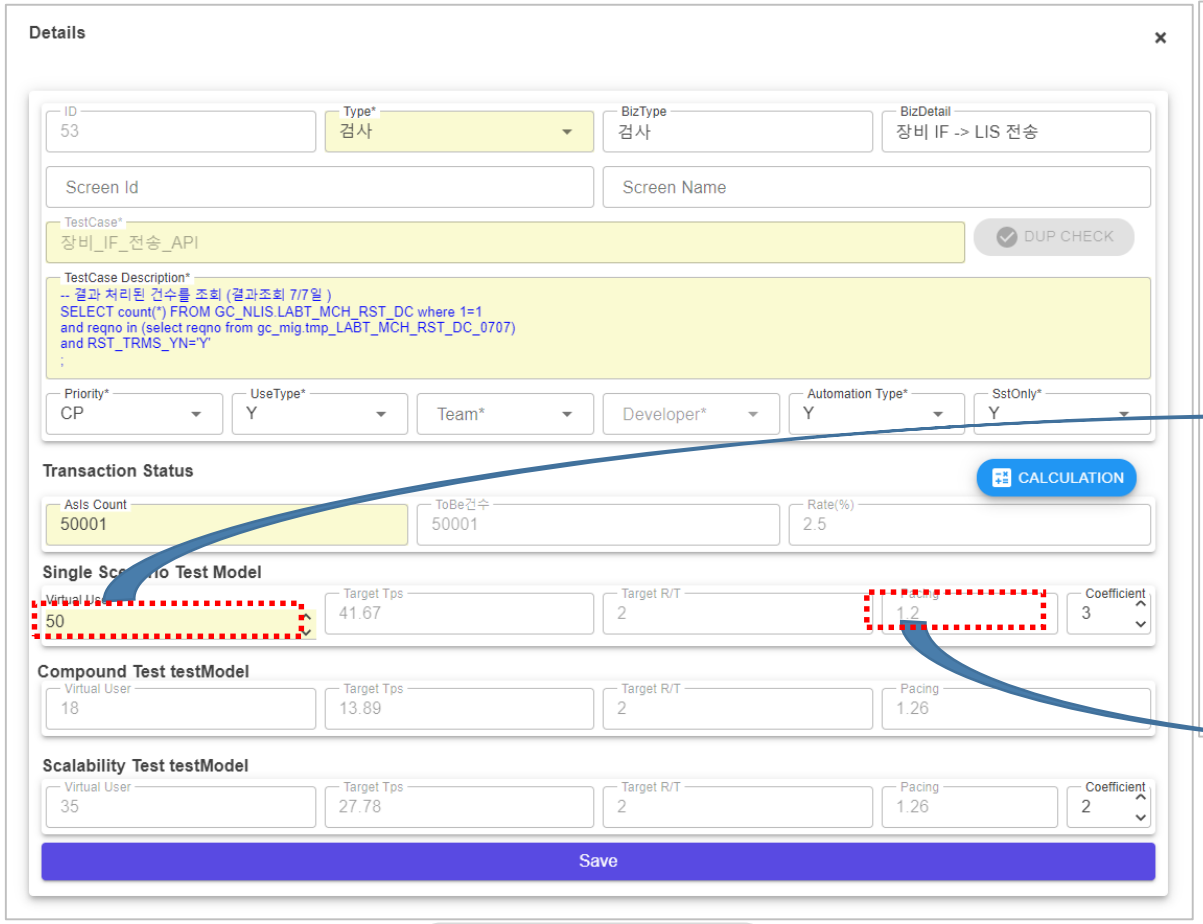
As-Is system workload analysis

Provide the basic data for the performance target



SLA Management

The load model (virtual user, objective TPS, Pacing Time) required for each test case can be calculated automatically based on the performance target and apply it in the test script.

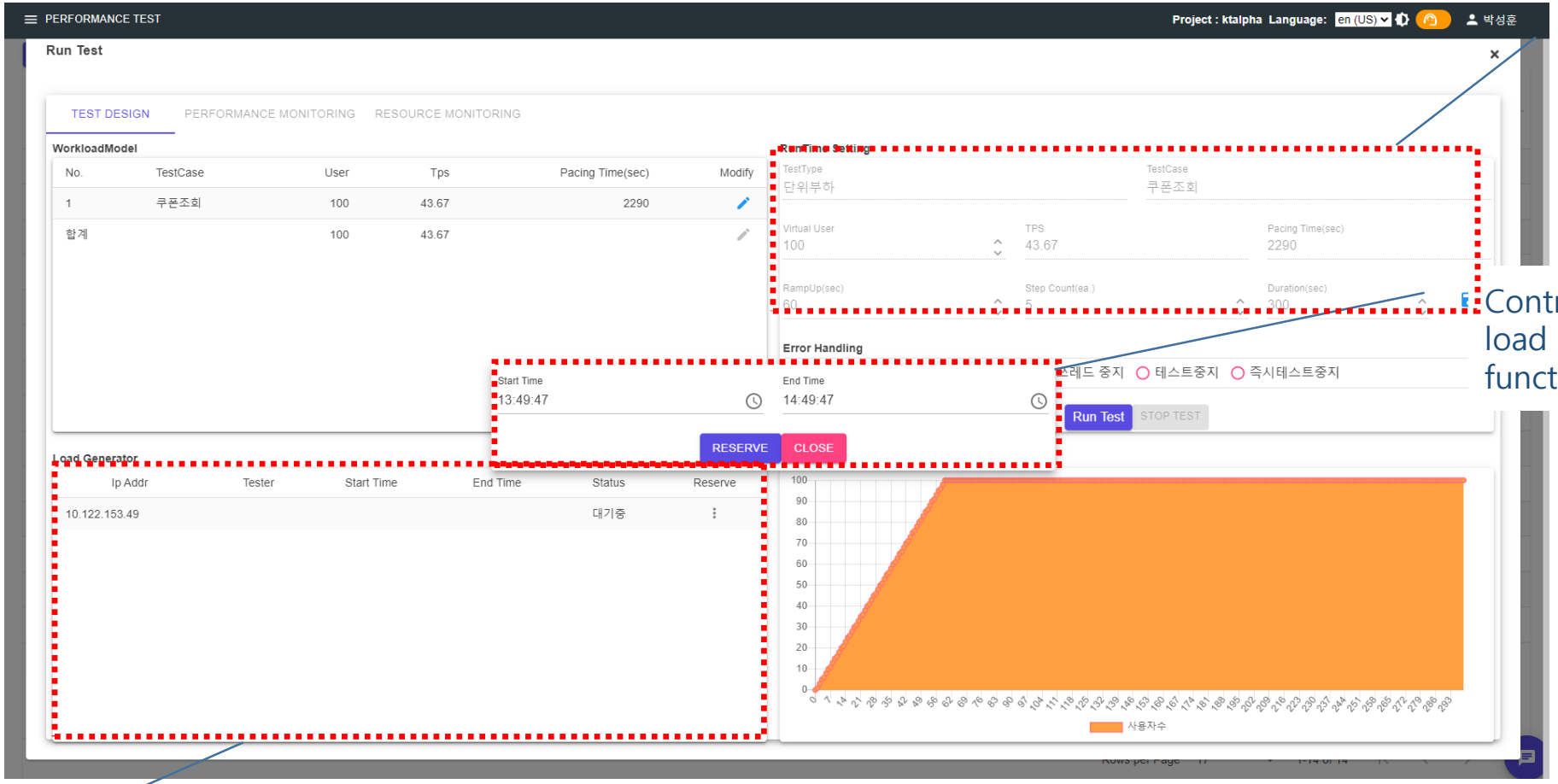


TestCase details

```
<com.blazemeter.jmeter.threads.concurrency.ConcurrencyThreadGroup guiclass="com.blazem
"com.blazemeter.jmeter.threads.concurrency.ConcurrencyThreadGroup" testname="1_쿠폰조회"
  <elementProp name="ThreadGroup.main_controller" elementType="com.blazemeter.jmeter.c
  <stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
  <stringProp name="TargetLevel">${1_vuser}</stringProp>
  <stringProp name="RampUp">60</stringProp>
  <stringProp name="Steps">5</stringProp>
  <stringProp name="Hold">180</stringProp>
  <stringProp name="LogFilename"></stringProp>
  <stringProp name="Iterations"></stringProp>
  <stringProp name="Unit">S</stringProp>
  <stringProp name="TestPlan.comments">This test plan is automatically created by apiR
</com.blazemeter.jmeter.threads.concurrency.ConcurrencyThreadGroup>
<hashTree>
  <Arguments guiclass="ArgumentsPanel" testclass="Arguments" testname="User Defined Va
    <collectionProp name="Arguments.arguments">
      <elementProp name="1_vuser" elementType="Argument">
        <stringProp name="Argument.name">1_vuser</stringProp>
        <stringProp name="Argument.value">100</stringProp>
        <stringProp name="Argument.metadata">=</stringProp>
      </elementProp>
      <elementProp name="1_pacingTime" elementType="Argument">
        <stringProp name="Argument.name">1_pacingTime</stringProp>
        <stringProp name="Argument.value">2290</stringProp>
        <stringProp name="Argument.metadata">=</stringProp>
      </elementProp>
    </collectionProp>
```

Jmeter Script

When performing a test, the pre-registered load generator is available after reservation and automatically reflected in the test script when the load model is modified.



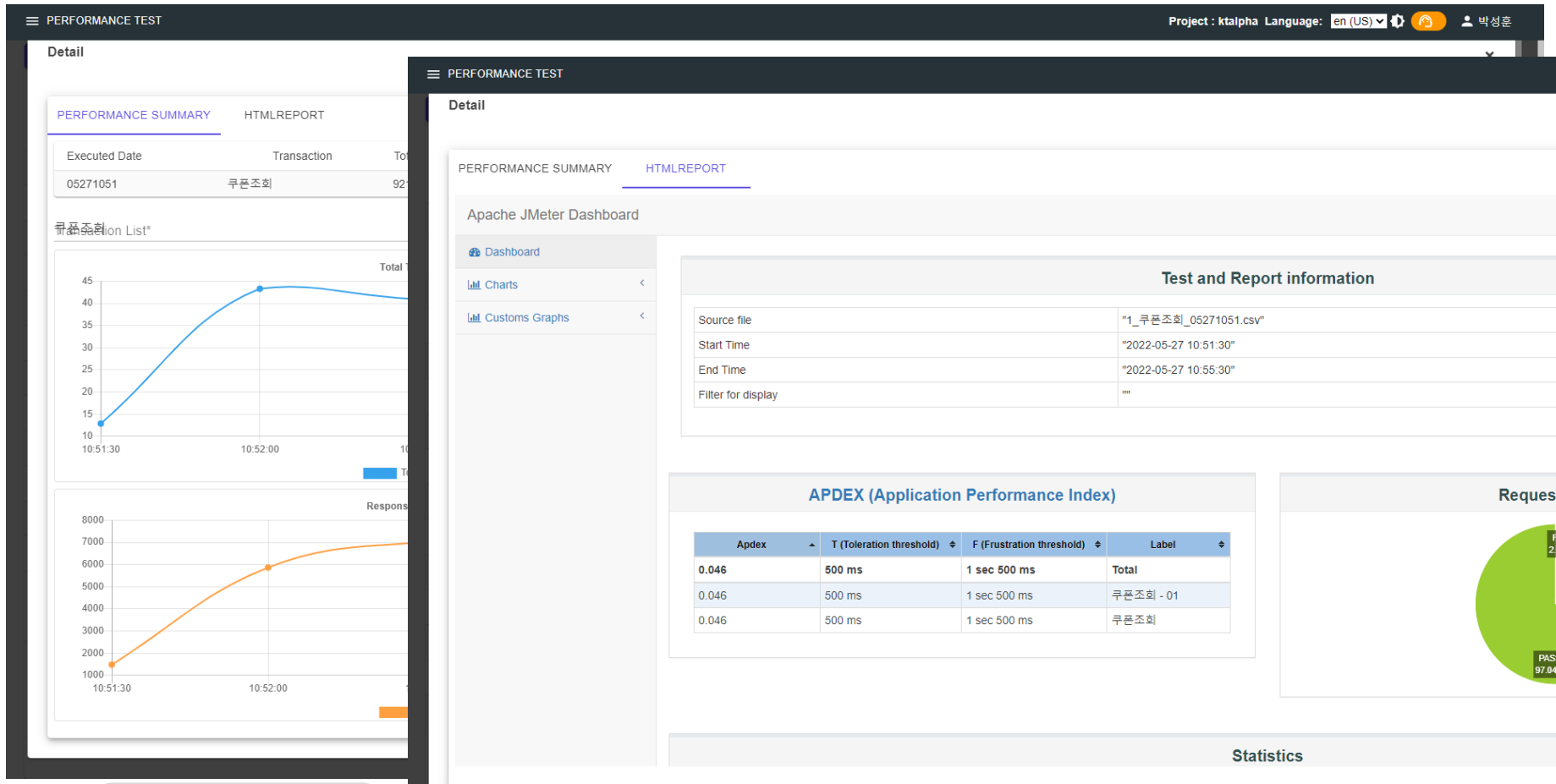
Modify the load model

Control test concurrency with load generators with scheduling functionality

Pre-registered load generator list

After performing the test, the test results are provided in historical form and provide trend analysis information for performance metrics.

Test results provide key performance metrics (TPS, response time, concurrent users) chart s, as well as HTML report provided by Jmeter.



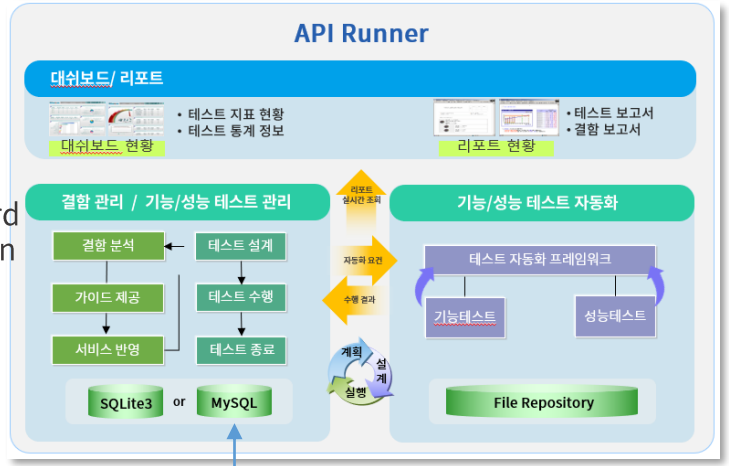
Test Result

Jmeter HTML Report

With the integration of visualization tool(Grafana), it provide real-time monitoring/test result analysis data for infrastructure systems associated with performance testing.

System Monitoring Dashboard

dashboard expansion



JDBC

monitoring collection
Java daemon process

Administrator registerable monitoring data

Cloud provider-provided data

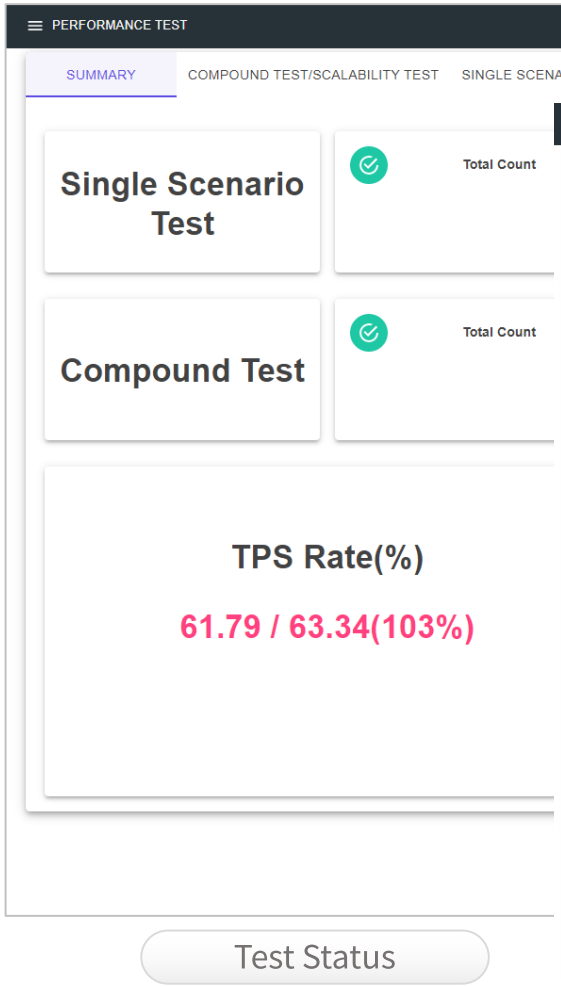
❖ Main Feat

- ✓ Provides system monitoring data
- ✓ Available on various types of data sources

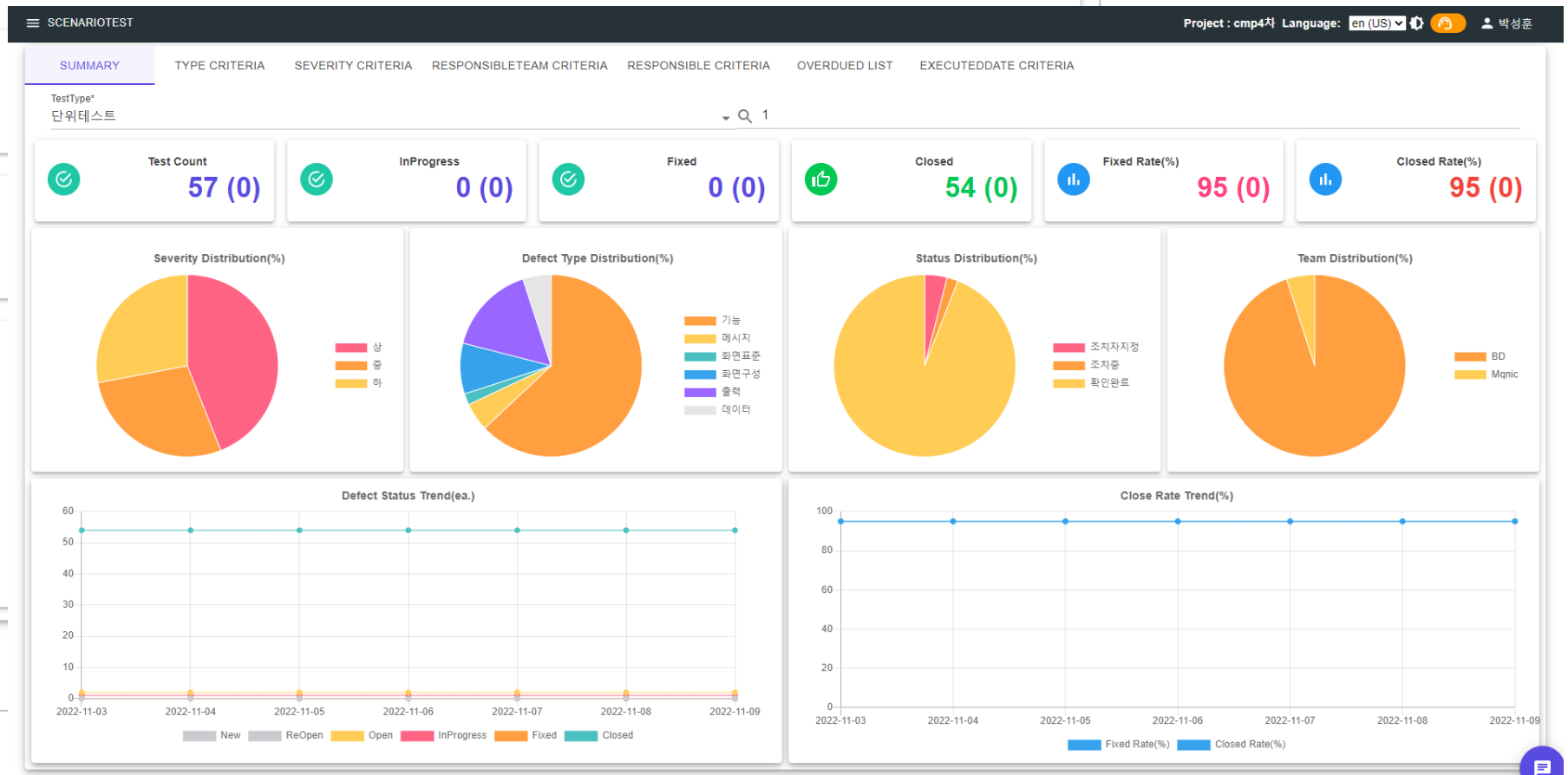
❖ Benefit

✓Improving performance test quality

Provides real-time dashboards based on key statuses related to testing.



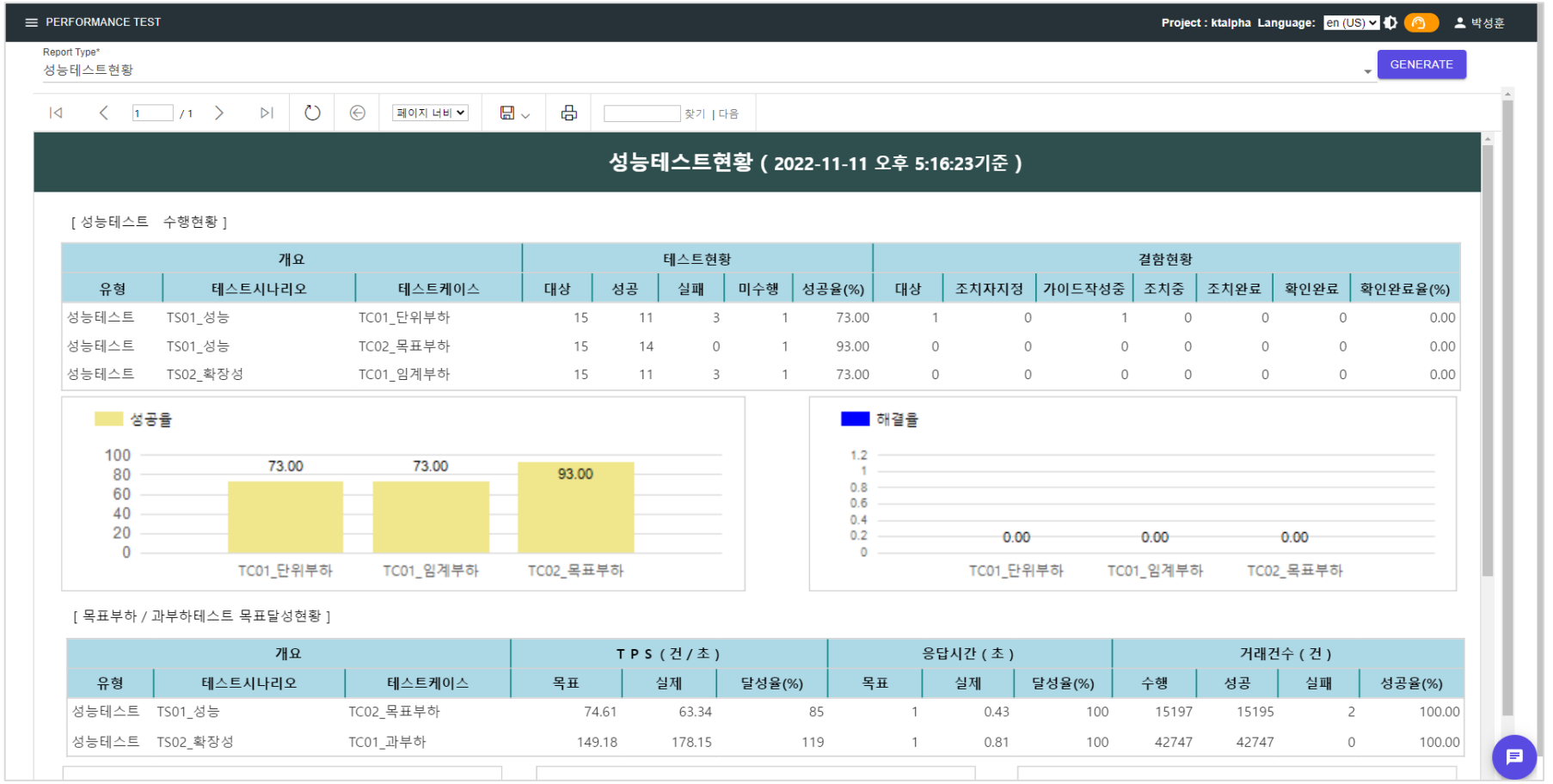
Test Status



Defect Status

Provides real-time reports based on key statuses related to testing.

✓ The report can be customized to meet the customer's requirements .



개요			테스트현황					결함현황						
유형	테스트시나리오	테스트케이스	대상	성공	실패	미수행	성공율(%)	대상	조치자지정	가이드작성중	조치중	조치완료	확인완료	확인완료율(%)
성능테스트	TS01_성능	TC01_단위부하	15	11	3	1	73.00	1	0	1	0	0	0	0.00
성능테스트	TS01_성능	TC02_목표부하	15	14	0	1	93.00	0	0	0	0	0	0	0.00
성능테스트	TS02_확장성	TC01_임계부하	15	11	3	1	73.00	0	0	0	0	0	0	0.00

개요			TPS (건 / 초)			응답시간 (초)			거래건수 (건)			
유형	테스트시나리오	테스트케이스	목표	실제	달성율(%)	목표	실제	달성율(%)	수행	성공	실패	성공율(%)
성능테스트	TS01_성능	TC02_목표부하	74.61	63.34	85	1	0.43	100	15197	15195	2	100.00
성능테스트	TS02_확장성	TC01_과부하	149.18	178.15	119	1	0.81	100	42747	42747	0	100.00

Thank you

wizbase