Test Strategy

Table of Contents

The overall system represents applications with frontend (UI) parts and API realization. Different applications will be tested using different approaches corresponding to their specifics.

1. Test Types

1.1. API Testing

The goal of API Testing is to verify that Application Programming Interface (API) operates according to technical specification, properly handles normal and invalid input, and delivers meaningful responses in time.

Test Objective:	Ensure API operates according to technical specification.
Technique:	Execute each use case, use-case flow or function, using valid and invalid data, to verify the following: ● The expected results occur when valid data is used. ● The appropriate error or warning messages are displayed when invalid data is used. ● Each business rule is properly applied. Consider the following while Test Cases/Checklists creation: ● Priorities are set for each test case/checklist item of the User Story test document. ● Bug Priorities definition together with Business value are assessed to identify the most critical functions, key areas and flows.
Completion Criteria:	● All planned tests have been executed. ● All identified defects have been addressed.
Special Consideration:	All API tests passed
Responsible:	QA team, manual and automation

Priority	Must be executed

1.2. Regression Testing

The goal of Regression Testing is to make sure that previously fixed bugs do not appear again and that added new features and bug fixes do not cause regress of the product quality. This type of testing is based on verifying the application and its internal processes by interacting with the application via the User Interface (UI) and analyzing the output or results. Testing will be performed automatically.

The application will be tested using a “black-box” either functional testing method without reference to the internal structure of the component or system

Test Objective:	Ensure proper feature functionality
Technique:	Execute each use case, use-case flow or function, using valid and invalid data, to verify the following: ● The expected results occur when valid data is used. ● The appropriate error or warning messages are displayed when invalid data is used. ● Each business rule is properly applied. Consider the following while Test Cases/Checklists creation: ● Priorities are set for each test case/checklist item of the User Story test document. ● Bug Priorities definition together with Business value are assessed to identify the most critical functions, key areas and flows.
Completion Criteria:	● All planned tests have been executed.

	• All identified defects have been addressed.
Special Considerations:	Primary browser considering Compatibility matrix
Responsible:	QA team, manual and automation
Priority:	Must be executed

1.3. Compatibility Testing

Browser Compatibility testing verifies the operation of the product on different combinations of Browsers, Operating systems and Mobile devices to make sure:

- In terms of functionality, the application must behave and respond the same way across different browsers.

Test Objective:	Ensure product functions and looks properly on the required browsers
Technique:	Testing will be executed using Functional test cases. UE testing will be executed by manual tests.
Completion Criteria:	For each covered combination all operations are successfully completed. The UI is not corrupted.
Special Considerations:	All browsers considering Compatibility matrix
Responsible:	QA team, manual and automation

Priority:	Will be executed if time allows

Compatibility matrix

The matrix was created based on Site Usage statistics (Oct 2021 - Aug 2022) for browsers and devices.

Browser version	**Desktop**	
	Windows	**Mac OS X**
	7/10/11	12.4
Firefox (2 previous versions)	⊕	⊕
Chrome (2 previous versions)	⊕	⊕
Firefox (the latest version until UAT)	⊕ (*primary*)	⊕
Chrome (the latest version until UAT)	⊕ (*primary*)	⊕

1.4. Performance Testing

The goal of performance testing is to determine and ensure that the system functions properly beyond the expected maximum workload. Additionally, it evaluates the performance characteristics, such as response times, transaction rates, and other time sensitive issues.

This type of testing will be conducted and supported by a third part.

1.5. Security Testing

System-level security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate

gateways. Access Control Testing focuses on application-level access to the business functions. It ensures that actors are restricted to specific functions or workflows, or are limited in the data that is available to them, in accordance to their roles and permissions.

Security testing will be out of scope for the project.

1.6. Unit Testing

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.

This type of testing will be conducted and supported by the developers team.

2. Test Levels

2.1. Smoke Tests

Smoke Test is performed to quickly assess the readiness of the product for further deeper and more thorough testing. It includes testing of major functions on the one most important and often used configuration.

Separate Smoke Test document is created to list the functionality that falls under the build acceptance testing. In Test Cases area/feature Smoke Test level is marked as High Priority.

If Smoke Test fails, Test Team sends notification and suspends testing until a corrected version of the product is available.

2.2. Critical Path Tests

The goal of the Critical Path Test is to find bugs that could affect the major functionality of the application that is most important for the product users. Critical Path Test will be performed automatically according to Test Cases documents on defined platforms.

In Test Cases Critical Path Test level is marked as Medium Priority.

2.3. Extended Tests

The Extended Test's goal is to find bugs related to the non-typical but still possible and likely usage scenarios (e.g. entering incorrect data into the fields, boundary testing and so on). Extended will be performed both according to test cases and using ad hoc testing scenarios.

In Test Cases Extended Test level is marked as Low Priority. This level of testing will be performed by manual QA team

3. Bug and Documentation Tracking

- Tools described in the section Test Tools will be used for bug reporting and documentation tracking.

3.1. Bug Priority Definitions

Priority	Guideline
Blocker	The defect blocks ~75%-100% of UAT scenarios (Story Acceptance Criteria) from particular sections and/or related to failure of order flow functionality, general crash of the application which leads to inability to use application. There is no workaround found. Bug blocks following testing Application, component or module crash or are not accessible
Critical	The defect blocks ~50-75% of UAT scenarios (Story Acceptance Criteria) from a particular section and/or leads to serious breaks of order flow functionality or crash of separate application parts like product/cs cockpits , hmc, etc. Any kind of workaround is found but difficult. Data corruption/loss, a problem in major functionality, no workaround is known
Major	The defect does not result in a failure of complete order flow but causes failure of full scenario. A problem with existing workaround, secondary features do not work properly

Minor	Any Defects which don't fall into previous categories should be considered as minor or trivial. Cosmetic flaw
Trivial	Very minor UI defects

4. Test Tools

Tool	Definition	Scope of application
Jira	Project and Process Tracking System. Bug Tracking System	Will be used for issue tracking and project management
JUnit	A framework for unit testing software in the Java language	Will be used to implement tests
RestAssured	Java library for REST API testing	Will be used to write API tests
Selenium WebDriver	Browser automation tool	Will be used to write UI tests
Jenkins	Build Management System CI/CD Tool	Will be used to automate the parts of software development related to building, testing, and deploying, facilitating CI and CD
Maven	Project build automation tool	Will be used to build project
Gradle	Project build automation	Can be used to build project

	tool	
Java 11	Programming language	Will be used to write a test framework
Cucumber	Tool for implementing the Behavior Driven Development (BDD) approach in Java	Will be used to implement BDD tests
Report Portal	ReportPortal is a service that provides increased capabilities to speed up results analysis and reporting through the use of built-in analytic features.	For representing a statistics and a test results visualization of already launched builds on the Jenkins job page

5. Test Environments

- Development environment. Used for implementation and debugging code, unit testing. Does not contain any user data. Supported by the Development team.
- Test environment. Used for regression (UI) and API testing by QA team. Contains test user data. 3rd party services are stubbed. Supported by the Test environment support team.
- Production environment. Environment where the software is deployed and the product is available to users. Contains real user data. All 3rd party services are integrated. Supported by the Production support team.

6. Test Phases

Phase	Activity	Work Products	Responsible
Test Planning	Review the business & software requirements and identify any defects in those specifications. After Requirements Analysis you have gathered a general idea of what needs to be tested, create a plan and scope for the tests.	● Test Plan ● Test Estimation ● Test Schedule	Business Analyst, QA Team, Scrum Master
Test Designing	Design / detail tests on the basis of detailed requirements / design of the software.	● Test Cases / Test Scripts / Test Data ● Requirements Traceability Matrix	Dev Team, QA Team, Scrum Master
Test Environment Setup	Setup the test environment (server / client / network, etc) with the goal of replicating the end-users' environment.	● Test Environment	DevOps

Test Execution	Execute Test Cases / Scripts in the Test Environment to see whether they pass and prepare various reports for various stakeholders.	• Test Results • Defect Reports • Test Closure Report	QA Team

7. Entry Criteria

- Requirements have been prepared and clarified
- Code compiles and unit tests pass.
- Unit tests have a minimum 75% code coverage.
- Regression test cases and API test cases were written and approved from Manual QA and developers sides

8. Acceptance Criteria

- Automated tests are written and passed.
- Tests have been merged to Main.
- Reviewed by Product owner.
- Story in Jira closed