TEAM PROJECT SmartHome FORD CS

SW-T

SmartHome FORD CS Test Specification

version v2022.1

By: N.N.  
 N.N.

N.N.

Supervisor: Ömer Karacan

Istanbul, 12.02.2022

Table of Contents

[1. Introduction 3](#_Toc108209619)

[2. Test planning 4](#_Toc108209620)

[3. Test monitoring and control 5](#_Toc108209621)

[4. Test analysis 6](#_Toc108209622)

[5. Test design 7](#_Toc108209623)

[5.1. Test Case C\_TC\_FSM\_#1 7](#_Toc108209624)

[5.2. Test Case C\_TC\_FSM\_TC\_#2 7](#_Toc108209625)

[6. Test implementation 8](#_Toc108209626)

[7. Test execution 9](#_Toc108209627)

[8. Test completion 10](#_Toc108209628)

# Introduction

# Test planning

**Determining the scope, objectives, and risks of testing**

**Defining the overall approach of testing**

**Integrating and coordinating the test activities into the software lifecycle activities**

**Making decisions about what to test, the people and other resources required to perform the various test activities, and how test activities will be carried out**

**Scheduling of test analysis, design, implementation, execution, and evaluation activities, either on particular dates (e.g., in sequential development) or in the context of each iteration (e.g., in iterative development)**

**Selecting metrics for test monitoring and control**

**Budgeting for the test activities**

**Determining the level of detail and structure for test documentation (e.g., by providing templates or example documents)**

# Test monitoring and control

**Produce comparison report of actual progress against planned progress**

**Test control involves taking actions necessary to meet the objectives of the test plan**

**Define and set Exit-criteria**

# Test analysis

**“What to test”**

**Analyzing the test basis appropriate to the test level being considered**

**Evaluating the test basis and test items to identify defects of various types**

**Identifying features and sets of features to be tested**

**Defining and prioritizing test conditions for each feature based on analysis of the test basis, and considering functional, non-functional, and structural characteristics, other business and technical factors, and levels of risks**

# Test design

**“How to test?”**

**Designing and prioritizing test cases and sets of test cases**

**Identifying necessary test data to support test conditions and test cases**

**Designing the test environment and identifying any required infrastructure and tools**

**Capturing bi-directional traceability between the test basis, test conditions, and test cases**

# Test Case C\_TC\_FSM\_#1

<short description if needed>

|  |  |
| --- | --- |
| Component  Test Case ID Name | <ID, e.g., C\_TC\_FSM\_#1> <name, e.g., “Test Transition “Ready to Charge” to “Vehicle connected””> |
| Preconditions | <preconditions (operational conditions) that should prevail at the beginning of test case execution> |
| Test Steps | 1. …. |
| Post-Condition | <postconditions (operational conditions) that should prevail after test case execution. Description of this condition is mandatory if it is, at the same time, the precondition of the next test case.> |
| Test Data | <input test data that should be provided to test execution> |
| Expected Result | <prognosed data according to specification> |
| Actual Result | <parsed data after test case execution> |
| Verdict (Pass/Fail) | <result of comparison between “Expected Result” and “Actual Result”> |
| Code Coverage | <image> |

# Test Case C\_TC\_FSM\_TC\_#2

# Test implementation

**“do we now have everything in place to run the tests?”**

**Developing and prioritizing test procedures and, potentially creating automated test scripts**

**Creating test suites from the test procedures and (if any) automated test scripts**

**Arranging the test suites within a test execution schedule in a way that results in efficient test execution**

**Building the test environment (including, potentially, test harnesses, service virtualization, simulators, and other infrastructure items) and verifying that everything needed has been set up correctly**

**Preparing test data and ensuring it is properly loaded in the test environment**

**Verifying and updating bi-directional traceability between the test basis, test conditions, test cases, test procedures, and test suites**

# Test execution

**Recording the IDs and versions of the test item(s) or test object, test tool(s), and testware**

**Executing tests either manually or by using test execution tools**

**Comparing actual results with expected results**

**Analyzing anomalies to establish their likely causes (e.g., failures may occur due to defects in the code, but false positives also may occur**

**Reporting defects based on the failures observed**

**Logging the outcome of test executions (based on the failures observed), e.g., pass, fail, blocked.**

**Repeating test activities either as a result of action taken for an anomaly, or as part of the planned testing (e.g., execution of a corrected test, confirmation testing, and/or regression testing)**

**Verifying and updating bi-directional traceability between the test basis, test conditions, test cases, test procedures, and test results.**

# Test completion

**Checking whether all defect reports are closed, entering change requests or product backlog items for any defects that remain unresolved at the end of test execution**

**Creating a test summary report to be communicated to stakeholders**

**Finalizing and archiving the test environment, the test data, the test infrastructure, and other test-ware for later reuse**

**Handing over the test-ware to the maintenance teams, other project teams, and/or other stakeholders who could benefit from its use**

**Analyzing lessons learned from the completed test activities to determine changes needed for future iterations, releases, and projects**

**Using the information gathered to improve test process maturity**