360T Assesment WalkThrough

Automating Test Scenarios for Telephone Dail Pad

By: Megha Vernekar

Agenda

- Introduction & Problem Statement
- Approach and Thought Process
- Tools and Framework
- Test Scenarios and Implementation
- Results
- Missing Test Scenarios

Introduction & Problem Statement

1. **Objective**: Test and automate scenarios for the **Telephone Dail Pad**.

2. The class file focuses on printing possible combination of alphabets for the numeric that you see in a telephone dial pad.
3. We must focus on writing test cases and automating them for the telephone dail pad class.

Approach and Thought Process

1. Brainstorming Test Scenarios:
    - Checked the given source code and went through it
    - Positive Scenarios (e.g., valid inputs with expected responses)
    - Negative Scenarios (e.g., special characters, empty inputs)
    - Edge cases (e.g., Floating point numbers, non-limit)

2. Divided the implementation task into multiple sub-tasks and addressed them accordingly
    - Designed test cases accordingly and went through them
    - Prepared test data
    - Decided on which tools and framework to use
    - Addressed the GitHub account and planned further course of action
    - Started implementation

Tools and Framework

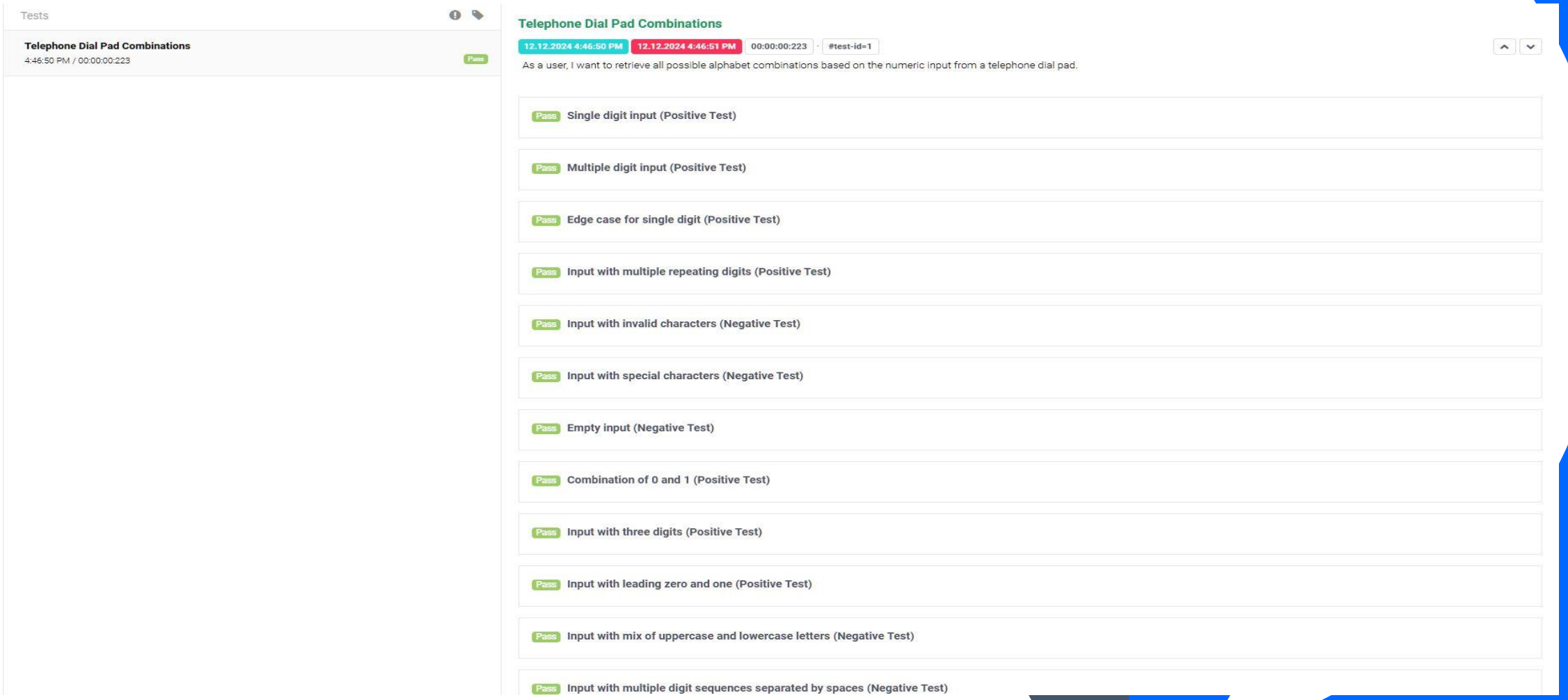
1. Programming Language: Java
2. Framework: BDD(Cucumber)
3. Build Tool: Maven
4. Version Control : GitHub
5. IDE: VSCODE
6. Additional Open Source Libraries: ExtentReport

Test Scenarios and Implementation

1. Test Scenarios:
    - Positive Testcase: 8
    - Negative Testcase: 10
    - Implementation Details: Testcases.xlsx

2. Implementation:
    - All the relevant details can be found in the below repo
    - Source Code: GitHub Repository

Result


Tests
Telephone Dial Pad Combinations
4:46:50 PM / 00:00:00:223
Pass
Telephone Dial Pad Combinations
12.12.2024 4:46:50 PM
12.12.2024 4:46:51 PM
00:00:00:223
#test-id=1
As a user, I want to retrieve all possible alphabet combinations based on the numeric input from a telephone dial pad.
Pass Single digit input (Positive Test)
Pass Multiple digit input (Positive Test)
Pass Edge case for single digit (Positive Test)
Pass Input with multiple repeating digits (Positive Test)
Pass Input with invalid characters (Negative Test)
Pass Input with special characters (Negative Test)
Pass Empty input (Negative Test)
Pass Combination of 0 and 1 (Positive Test)
Pass Input with three digits (Positive Test)
Pass Input with leading zero and one (Positive Test)
Pass Input with mix of uppercase and lowercase letters (Negative Test)
Pass Input with multiple digit sequences separated by spaces (Negative Test)

Result

Started
Dec 12, 2024 04:46:50 PM

Ended
Dec 12, 2024 04:46:51 PM

Features Passed
1

Features Failed
0

Features

1 features passed
0 features failed, **0** skipped, **0** others

Scenarios

32 scenarios passed
0 scenarios failed, **0** skipped, **0** others

Steps

96 steps passed
0 steps failed, **0** skipped, **0** others

Timeline

Tags

NAME	PASSED	FAILED	SKIPPED	OTHERS	PASSED %
@Medium	12	0	0	0	100%
@Low	2	0	0	0	100%
@High	18	0	0	0	100%

Missing Test Scenarios

1. Some of the test scenarios are considered but have not been implemented. Below are some of them for your reference.

    - Input with excessively large digits (e.g., "99999999999999999")
        - No limit was present in the code, hence this could be an important scenario but has not been implemented as there was less knowledge on the limit functionality of the source code.

Result

1. ExtentReport file with all the test case details are generated
2. This file should contain all the details of relevant and executed test scenarios

Thank You

Megha Vernekar