Contents

[A Brief Explanation of Solution for a Non-Technical Manager/Stakeholder 1](#_Toc191329413)

[Executive Summary 1](#_Toc191329414)

[What Does This Automation Do? 1](#_Toc191329415)

[Test Scenario 1: Duplicate Column Validation 1](#_Toc191329416)

[Test Scenario 2: Yes/No Value Validation 2](#_Toc191329417)

[How the Results Are Provided 2](#_Toc191329418)

[Business Impact & Value 2](#_Toc191329419)

[Potential Bugs/Issues Found 2](#_Toc191329420)

[Issue 1: Duplicate Column Found (Test Data) 2](#_Toc191329421)

[Issue 2: Invalid Value in Yes/No Column 3](#_Toc191329422)

[**Issue:** 3](#_Toc191329423)

[**Details:** 3](#_Toc191329424)

[**Why This is Important?** 3](#_Toc191329425)

[**Suggested Fix:** 3](#_Toc191329426)

[Summary of Findings 4](#_Toc191329427)

[Conclusion 4](#_Toc191329428)

[Project Structure 4](#_Toc191329429)

# A Brief Explanation of Solution for a Non-Technical Manager/Stakeholder

## Executive Summary

This solution is a test automation framework designed to validate regulatory report files and ensure data consistency and compliance with regulatory requirements. It provides a fast, reliable, and repeatable way to verify reports before submission, reducing manual effort and human errors.

## What Does This Automation Do?

I implemented two key test scenarios to verify regulatory data in report files. More scenarios can be added to this solution.

## Test Scenario 1: Duplicate Column Validation

* What it does: Ensures that no column appears twice in the same report file.
* Why it's important: Duplicate columns can cause inconsistent data and reporting errors.
* Outcome: If a duplicate column is found, the test fails and logs the issue for correction. Alerting can be added in future steps.

## Test Scenario 2: Yes/No Value Validation

* What it does: Checks if specific columns (marked as Yes/No fields) contain only "Yes" or "No" values.
* Why it's important: Regulatory data often requires strict value formats, and incorrect entries can cause submission failures.
* Outcome: If an invalid value (e.g., "Maybe") is found, the test fails and highlights the incorrect data.

## How the Results Are Provided

* Clear Pass/Fail Reports – Results are displayed in a detailed HTML test report.
* Automated Execution – The tests can be run on demand, as part of CI/CD pipelines (future state), or integrated into existing workflows.

## Business Impact & Value

✔ Saves Time & Effort – Reduces manual validation from hours to seconds.  
✔ Improves Data Accuracy – Ensures error-free regulatory reports before submission.  
✔ Prevents Compliance Risks – Helps avoid potential fines or penalties due to incorrect data.  
✔ Scales with Business Needs – New validation rules can be added easily, making it future-proof.

# Potential Bugs/Issues Found

Since no duplicate columns were originally found, I manually introduced errors in the data to test the robustness of the automation framework. Below are the results based on the test execution.

## Issue 1: Duplicate Column Found (Test Data)

**Issue:**

A duplicate column was manually added to simulate an error and verify that the test correctly detects it.

**Details:**

| **File Name** | **Duplicate Column Found** |
| --- | --- |
| y\_02.01.csv | "c0040" (appeared twice) |

**Why This is Important?**

* The test correctly failed and flagged the issue.
* Confirms that the framework successfully detects duplicate columns.
* Ensures that real duplicates would be caught in future reports.

A screenshot of a computer

AI-generated content may be incorrect.

**Suggested Fix (If This Were a Real Issue):**

* Ensure each column name appears only once per file.
* Remove or rename duplicate column headers.

## Issue 2: Invalid Value in Yes/No Column

**Issue:**

An incorrect value was found in a column that should only contain "Yes" or "No" in y\_07.01.csv.  
This was not manually added but already existed in the file.

**Details:**

| **File Name** | **Column** | **Invalid Value** |
| --- | --- | --- |
| y\_07.01.csv | "y\_07.01.0080" | " PJKqTqBT:xAg2bs" |

**Why This is Important?**

* The test correctly failed and highlighted the issue.
* Confirms that the framework only allows "Yes" or "No" values.
* Helps prevent submission errors due to incorrect formats.

A screenshot of a computer screen

AI-generated content may be incorrect.

**Suggested Fix:**

* Replace " PJKqTqBT:xAg2bs" with a valid value: "Yes" or "No".
* Check other Yes/No fields in y\_07.01.csv for similar mistakes.

### Summary of Findings

| Issue Type | Affected File(s) | Outcome |
| --- | --- | --- |
| Duplicate Column (Simulated Error) | y\_02.02.csv | ✅ Test correctly detected duplicate " c0040" |
| Invalid Yes/No Value (Existing Data Issue) | y\_07.01.csv | ✅ Test correctly flagged " PJKqTqBT:xAg2bs" as an invalid value |

### Conclusion

* The automation framework is working as expected.
* Duplicate column detection and Yes/No validation successfully identified issues.
* y\_07.01.csv already contained invalid data, and the test correctly reported it.
* The framework can now be used to validate future regulatory reports.

# Project Structure

RegDatCheck/ (Project Root)

├── src/

│ ├── test/

│ │ ├── java/

│ │ │ ├── StepDefinitions / # 📌 Cucumber Step Definitions

│ │ │ │ ├── NoDuplicateColumnsSteps.java # Scenario 1: Duplicate Column Check

│ │ │ │ ├── YesNoValidationSteps.java # Scenario 2: Yes/No Validation

│ │ │ ├── utils/ # 📌 Utility Functions (Reusable Methods)

│ │ │ │ ├── FileValidationUtils.java # Common validation logic

│ │ │ ├── runners/ # 📌 Cucumber Test Runner

│ │ │ │ ├── TestRunner.java # Main Test Runner File

│ ├── resources/

│ │ ├── features/ # 📌 Cucumber Feature Files

│ │ │ ├── NoDuplicateColumnsSteps.feature

│ │ │ ├── Validate\_yes\_no.feature

│ │ ├── files/ # 📌 Test Data (CSV Report Files)

│ │ │ ├── y\_01.01.csv ✅ Sample Report File

│ │ │ ├── y\_02.01.csv ✅ Sample Report File

│ │ │ ├── y\_02.02.csv ✅ Sample Report File

│ │ │ ├── y\_02.03.csv ✅ Sample Report File

│ │ │ ├── y\_07.01.csv ✅ Sample Report File

│ │ │ ├── y\_99.01.csv ✅ Sample Report File

│ │ ├── requirements/

│ │ │ ├── Data Fields\_New.xlsx ✅ Regulatory Definition File

├── target/ # 📌 Test Reports (Generated After Running Tests)

│ ├── cucumber-reports/

│ │ ├── cucumber.html ✅ Test Report

│ │ ├── cucumber.json

│ │ ├── cucumber.xml

├── pom.xml # 📌 Maven Dependencies

├── README.md # 📌 Project Documentation