TEST PLAN FOR NO SQL TO RDBMS CONVERTER

ChangeLog *:*

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
001	30.10.2023	Vageesha Rai	Initial Draft

1 Introduction

NoSQL databases are designed to handle large volumes of unstructured and semi-structured data and can be highly scalable and flexible. They are often used in modern web applications, where data is not necessarily structured in a way that fits well with traditional relational databases. On the other hand, RDBMS systems are designed to store structured data in tables with rows and columns and to enforce relationships between different tables. They have been used for decades and have a proven track record of reliability, consistency, and robustness. They are often used in applications that require transactions and data consistency, such as banking, finance, and healthcare. Our project aims to convert NoSQL data to RDBMS (Relational Database Management System) data so that the specific needs and requirements of a particular application or organization is fulfilled. .

1.1 Scope

1.1.1 In Scope

Scope defines the features, functional or non-functional requirements of the software that **will be** tested.
Features of the Project:

1. File format support: The converter website would need to support various NoSQL file formats, such as JSON, BSON, and XML, and convert them to relational database formats such as MySQL, PostgreSQL, or Oracle.

2. Schema mapping: The converter would need to map the NoSQL data schema to the relational database schema. This may involve detecting data types, relationships, and constraints, and generating the SQL schema definition language (SDL) commands to create the tables, columns, and indexes.

3. Data transformation: The converter would need to transform the NoSQL data to fit the relational database schema. This may involve flattening nested data structures, splitting data across multiple tables, and handling data type conversions.

4. User interface: The converter website would need to provide a user-friendly interface for users to upload their NoSQL files, specify the target RDBMS system, and configure any additional settings, such as data mapping rules.

5. Security: The converter website would need to ensure the security of the user's data and protect against unauthorized access or data breaches.

1.1.2 Out of Scope

Out Of Scope defines the features, functional or non-functional requirements of the software that **will NOT be** tested :

1. Custom Knowledge Database: Ensure the database provides accurate and reliable descriptions and context to users.

2. Scalability:: Load testing to ensure the platform can handle increased user loads

efficiently.

1.2 Quality Objective

Here make a mention of the overall objective that you plan to achieve without your testing Some objectives of your testing project could be
Ensure the Application Under Test conforms to functional and nonfunctional requirements.
Ensure the AUT meets the quality specifications defined by the client.
Bugs/issues are identified and fixed before go live

1.3 Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

- QA Analyst : Vageesha Rai
- Test Manager : Prof. Shreela Pareek
- Configuration Manager: Prof. Neha Shukla
- Developers : Saumya Singh, Vageesha Rai, VIkas Kumar Verma
- Installation Team :Prof. Shreela Pareek, Prof. Neha Shukla, Vageesha Rai, Saumya Singh, Vikas Kumar Verma

2 Test Methodology

2.1 Overview

We are using an iterative testing approach to make sure our project works well. This means we test it in small steps, starting with checking if each part works on its own. Then, we see how different parts work together.
We keep testing as we make changes and add new things. This way, we make sure our project is always working well, even after modification.

2.2 Test Levels

Test Levels define the Types of Testing to be executed on the Application Under Test (AUT).

We aim to test our project at the following levels :

1) Unit Testing: This is the lowest level of testing and focuses on individual components or functions within the software. Developers often perform unit tests to verify that specific parts of the code work correctly.

2) Integration Testing: This level of testing checks how different components or modules of the software work together. It ensures that integrated parts of the software function as intended.

3) System Testing: At this level, the entire system is tested as a whole. It verifies that the software

meets its specified requirements and functions properly in its intended environment.

2.3 Test Completeness

Here you define the criterias that will deem your testing complete.
For instance, a few criteria to check Test Completeness would be

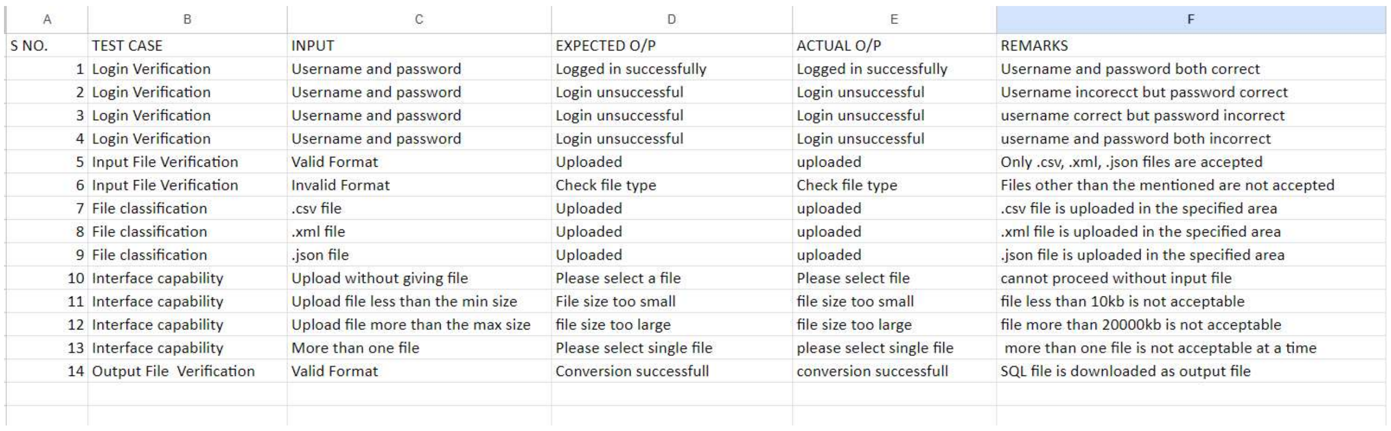
- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release

3 Test Deliverables

Here are the deliverables

- Test Plan
- Test Cases
- Bug Reports
- Test Strategy

4 Test Cases :

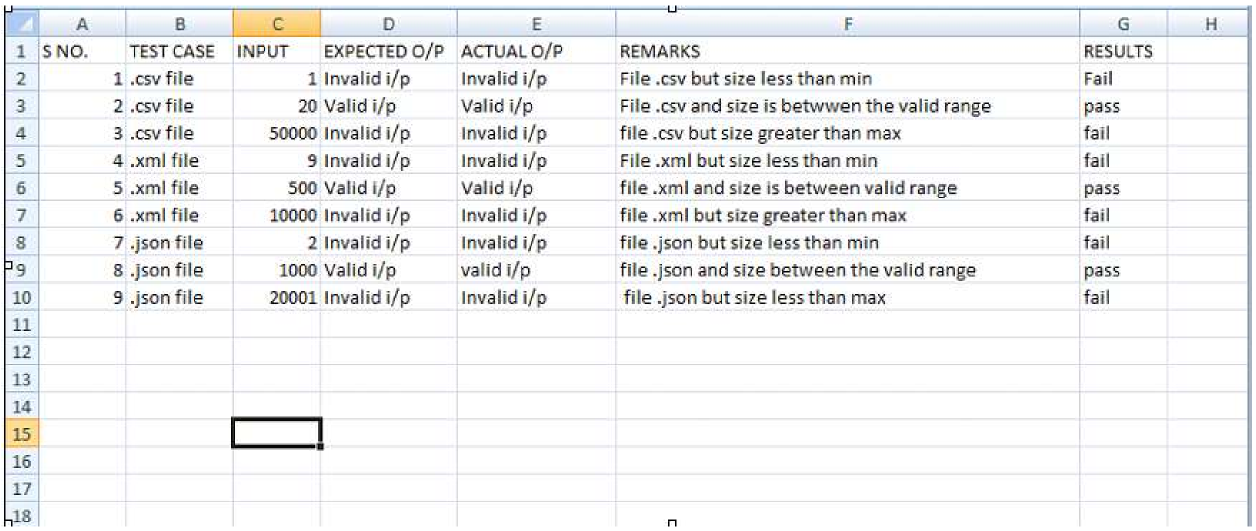


A	B	C	D	E	F
S NO.	TEST CASE	INPUT	EXPECTED O/P	ACTUAL O/P	REMARKS
1	Login Verification	Username and password	Logged in successfully	Logged in successfully	Username and password both correct
2	Login Verification	Username and password	Login unsuccessful	Login unsuccessful	Username incorecct but password correct
3	Login Verification	Username and password	Login unsuccessful	Login unsuccessful	username correct but password incorrect
4	Login Verification	Username and password	Login unsuccessful	Login unsuccessful	username and password both incorrect
5	Input File Verification	Valid Format	Uploaded	uploaded	Only .csv, .xml, .json files are accepted
6	Input File Verification	Invalid Format	Check file type	Check file type	Files other than the mentioned are not accepted
7	File classification	.csv file	Uploaded	uploaded	.csv file is uploaded in the specified area
8	File classification	.xml file	Uploaded	uploaded	.xml file is uploaded in the specified area
9	File classification	.json file	Uploaded	uploaded	.json file is uploaded in the specified area
10	Interface capability	Upload without giving file	Please select a file	Please select file	cannot proceed without input file
11	Interface capability	Upload file less than the min size	File size too small	file size too small	file less than 10kb is not acceptable
12	Interface capability	Upload file more than the max size	file size too large	file size too large	file more than 20000kb is not acceptable
13	Interface capability	More than one file	Please select single file	please select single file	more than one file is not acceptable at a time
14	Output File Verification	Valid Format	Conversion successfull	conversion successfull	SQL file is downloaded as output file

Boundary Value analysis:

Input file size must be between 10-20000 kB:

Invalid (min-1)	Valid (min, min + 1, nominal, max – 1, max)	Invalid (max + 1)
1KB	10KB, 11KB, 10005KB, 19999KB, 20000KB	20001KB

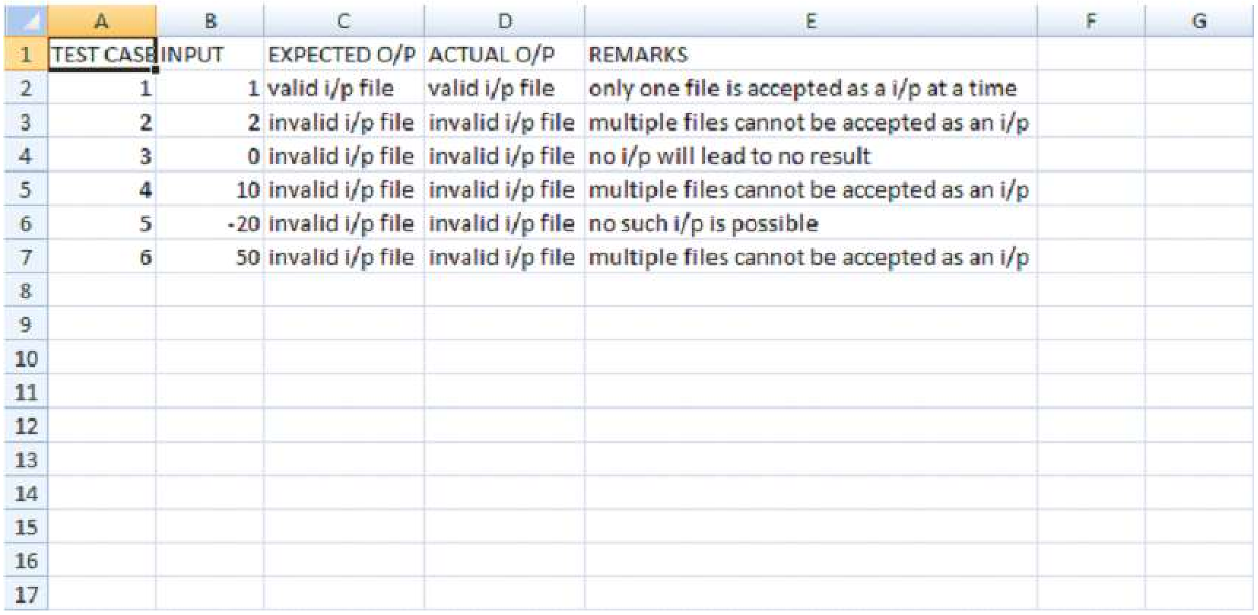


S NO.	TEST CASE	INPUT	EXPECTED O/P	ACTUAL O/P	REMARKS	RESULTS
1	.csv file	1	Invalid i/p	Invalid i/p	File .csv but size less than min	Fail
2	.csv file	20	Valid i/p	Valid i/p	File .csv and size is betwwen the valid range	pass
3	.csv file	50000	Invalid i/p	Invalid i/p	file .csv but size greater than max	fail
4	.xml file	9	Invalid i/p	Invalid i/p	File .xml but size less than min	fail
5	.xml file	500	Valid i/p	Valid i/p	file .xml and size is between valid range	pass
6	.xml file	10000	Invalid i/p	Invalid i/p	file .xml but size greater than max	fail
7	.json file	2	Invalid i/p	Invalid i/p	file .json but size less than min	fail
8	.json file	1000	Valid i/p	valid i/p	file .json and size between the valid range	pass
9	.json file	20001	Invalid i/p	Invalid i/p	file .json but size less than max	fail

Equivalence Class Testing:

No of files acceptable as input = 1

Invalid	Valid	Invalid
0	1	2, 20, 200,...........................



TEST CASE	INPUT	EXPECTED O/P	ACTUAL O/P	REMARKS
1	1	valid i/p file	valid i/p file	only one file is accepted as a i/p at a time
2	2	invalid i/p file	invalid i/p file	multiple files cannot be accepted as an i/p
3	0	invalid i/p file	invalid i/p file	no i/p will lead to no result
4	10	invalid i/p file	invalid i/p file	multiple files cannot be accepted as an i/p
5	-20	invalid i/p file	invalid i/p file	no such i/p is possible
6	50	invalid i/p file	invalid i/p file	multiple files cannot be accepted as an i/p

DECISION TABLE:

Conditions	Input-1	Input-2	Input-3	Input-4
Type	T	T	F	F
Size	T	F	T	F
Result	Accepted	Not Accepted	Not Accepted	Not Accepted

Example:

Conditions	Input-1	Input-2	Input-3	Input-4	Input-5	Input-6
Type	.csv	.csv	.xml	.xml	.json	.json
Size	10kb-20000kb	>=20000kb	10kb-20000kb	>=20000kb	10kb-20000kb	>=20000kb
Result	Accepted	Not Accepted	Accepted	Not Accepted	Accepted	Not Accepted

5 Resource & Environment Needs

5.1 Testing Tools

List of Tools like

- Selenium
- Mentis BT
- Automation BT

5.2 Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

Following **software's** are required in addition to client-specific software.

- Windows 10 and above preferred
- VSCode 2022 or above preferred
- Chrome, Mozilla or Edge Preferred over non-chromium based browsers

6 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION

API	Application Program Interface
AUT	Application Under Test