**Test Plan for Technical Challenge - Oracle**

**1. Test Plan Overview**

**Objective**: To ensure the correctness and reliability of the data processing application by testing various input sources (Excel, database, PDF) and report generation functionalities.

**Test Types**:

* Unit Tests
* Integration Tests
* Edge Cases

**2. Test Scenarios**

**2.1 Unit Tests**

**1. Parsing Data Entries**

**Objective**: Verify that the data parsing from different sources correctly converts input data into DataEntry objects.

**Test Cases**:

* **Excel Input**:
  + **Valid Input**: Test with a properly formatted Excel file.
  + **Invalid Input**: Test with an Excel file missing some columns or having incorrect data types.
* **Database Input**:
  + **Valid Input**: Test with a properly configured database and valid query.
  + **Invalid Input**: Test with missing database connection details or incorrect query.
* **Manual Input**:
  + **Valid Input**: Test with a long formatted String.
  + **Invalid Input**: Test with a String unexpected format or missing data.

**Edge Cases**:

* Empty file/database.
* File with only headers.
* Corrupt or unreadable file.

**Expected Results**:

* Correctly parsed DataEntry objects.
* Handling of invalid data with appropriate error messages.

**2. Report Generation**

**Objective**: Validate the report generation logic for correct results based on the parsed data.

**Test Cases**:

* **Unique Customer Count by Contract**:
  + **Multiple Contracts**: Check unique customer counts across various contracts.
  + **Single Contract**: Validate when all data is for a single contract.
* **Unique Customer Count by Geozone**:
  + **Multiple Geozones**: Verify unique customer counts for different geozones.
  + **Single Geozone**: Verify unique customer counts when all data is for one geozone.
* **Average Build Duration by Geozone**:
  + **Valid Durations**: Test with normal build durations.
  + **Mixed Valid and Invalid Durations**: Test with some invalid durations.
* **Unique Customers by Geozone**:
  + **Multiple Geozones**: Check unique customer IDs for each geozone.
  + **Single Geozone**: Validate with all data in one geozone.

**Edge Cases**:

* Build durations with non-numeric values.
* Duplicate customer IDs in the same geozone or contract.
* Formatting in build durations.

**2.2 Integration Tests**

**Objective**: Ensure that the system components (input readers, data processing, and report generation) work together as expected.

**Test Cases**:

* **Integration of Excel Reader and Report Generator**:
  + **End-to-End Test**: Use a complete Excel file to verify the integration of reading data and generating reports.
* **Integration of Database Reader and Report Generator**:
  + **End-to-End Test**: Use data from the database and validate the report generation.

**Edge Cases**:

* Integration with a corrupted or partially readable file/database.
* Large volume of data to test performance and scalability.

**3. Test Data**

\*\*1. **Excel Data**:

* Valid: valid\_data.xlsx with correctly formatted columns.
* Invalid: invalid\_data.xlsx with missing columns or incorrect data.

\*\*2. **Database Data**:

* Valid: Ensure the database has a table data\_table with correct schema and data.
* Invalid: Test with incorrect credentials or an empty table.

**4. Test Execution**

**1. Unit Tests**:

* **Tools**: TestNG
* **Commands**: mvn test for Maven users

**2. Integration Tests**:

* **Tools**: TestNG or manual testing
* **Commands**: mvn test or manual verification of integration

**5. Expected Results**

* **Unit Tests**:
  + The application should correctly parse and handle data from various sources.
  + Reports should be accurate and correctly reflect the input data.
* **Integration Tests**:
  + The application should handle data end-to-end from input through to reporting.
  + The system should gracefully handle errors and edge cases.

**6. Handling Failures**

\*\*1. **Invalid Input Handling**:

* Log appropriate error messages.
* Ensure the application does not crash.

\*\*2. **Edge Cases**:

* Provide default values or error handling for invalid durations or missing data.

\*\*3. **Integration Failures**:

* Ensure the application can recover from issues like file corruption or database unavailability.

\*\*4. **Performance Issues**:

* Test with large datasets to ensure performance and optimize if necessary.