# Solution Document – Application Pokedex

# High Level Requirement

Prepare an application which acts as a wrapper around the existing Poke API and Fun transaction APIs and provides two different endpoints.

1. An endpoint to fetch the Pokemon information by Name.
2. An endpoint to fetch the Pokemon information along with the translation of Description in either Yoda or Shakespeare.

# High Level Solution

We have created a Spring Boot Application, with a Rest Controller which can accommodate the endpoints above.

Business logic has been kept in the Service.

Spring Boot is an ideal choice as it is clean and quick and manages most of the non-business requirements along with a huge support community.

Spring Boot will allow us monitor and manage the application in production using the actuator provided by the Spring Boot.

Note: Before moving the application to production it would be wise to add OAuth for the authentication.

# Documentation

Along with adding the **Readme.md** file, we have added proper comments to the Java program which can be used to generate the Javadoc.

# Unit test cases

Cover at least 85% of the code using the Unit test cases. JUnits in this case.

Graphical user interface, application

Description automatically generated

# Test Results

Following are the test results of the Application.

1. Calling the API to get the Pokemon information for the name **mewtwo** along with the translation. As this Pokemon is legendary, descripted will be translated using Yoda translation.

Graphical user interface, text, application, email

Description automatically generated

1. Fetch Pokemon information with the name **mewtwo** without the translation.

Graphical user interface, text, application, email

Description automatically generated

1. Fetch Information of the Pokemon **pikachu** along with translation, as this pokemon’s habitat is forest and not legendary. Description will be translated using the Shakespeare translation.

# Graphical user interface, text, application, email, website Description automatically generated

1. Tried calling the API with an Invalid resource name **Nawazish**, resource returned has null values and http status code is **404 Not found**.

# Graphical user interface, text, application, email Description automatically generated