System Architecture Overview

1. Introduction

• Purpose of the Document: This document outlines the architecture and key design decisions for the City Suggestion API. The API Specification can be found at https://github.com/renaldiaddison/citysuggestion.

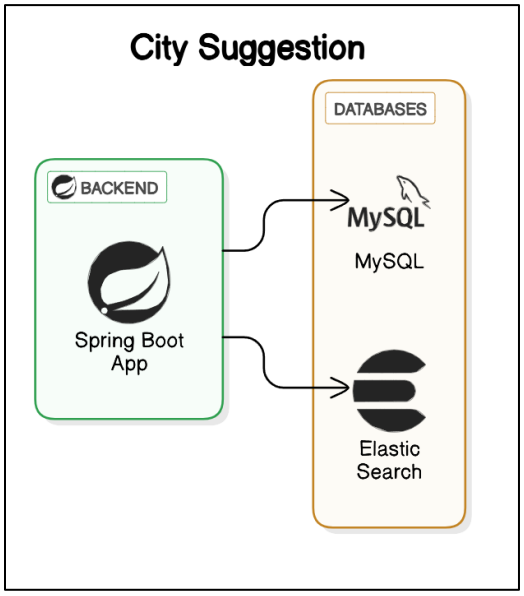
• Spring Boot API Link: https://citysuggestion-production.up.railway.app

• Elastic Search Link: https://8a2f0674a575467681c1c8df633dc2de.ap-southeast-1.aws.found.io/

• Scope: Covers the back-end, database, and search engine components.

2. System Architecture

• Architecture Diagram:



• Description: The architecture consists of a back-end Spring Boot RESTful API, a MySQL database, and an Elastic Search service. The back-end communicates with the MySQL database and Elastic Search service using its perspective driver. The Spring Boot API and MySQL database is hosted using Railway and for the Elastic Search service is hosted using elastic.co.

3. Key Components

Back-End API

• Description: A Spring Boot application that handles business logic, data processing, and API endpoints.

• Technologies Used: Spring Boot.

Database

• Description: MySQL is a RDBMS that is used to store all data.

• Technologies Used: MySQL.

Search Engine

• Description: Elastic Search is used to store frequently searched data and is used because of it's capability of more faster search.

• Technologies Used: Elastic Search.

4. Search Score Calculation

• In this project, a scoring mechanism is implemented to evaluate the similarity between the retrieved data and the searched data. The score is calculated using a combination of two algorithms: the Jaro-Winkler string similarity algorithm, which is assigned a weight of 0.7, and the Haversine algorithm, used for calculating distance, which is assigned a weight of 0.3.