# Gas Station Finder - Implementation Manual

Overview
The Gas Station Finder application is a JavaFX-based desktop application that provides users with the ability to search for gas stations by zip code and submit reviews for them. The application uses the Google Places API to retrieve real-time data about gas stations.

Architecture
The application is structured into several classes, each serving a distinct role within the MVC (Model-View-Controller) design pattern:

- `Main.java`: This is the entry point of the application that launches the JavaFX user interface. It sets up the primary stage and scene, and initialises other components of the UI.
- `Controller.java`: This class acts as the intermediary between the view (UI) and the model (data). It processes user inputs from the UI, interacts with the `ApiHandler` to fetch data, and updates the view accordingly.
- `ApiHandler.java`: Responsible for making API requests to the Google Places API, parsing the JSON responses, and returning a list of `GasStation` objects to the `Controller`.
- `GasStation.java`: Represents the model for a gas station, containing properties such as name, address, and reviews. It also contains methods for adding reviews to a gas station.
- `Review.java`: Represents the model for a review, containing properties such as username, rating, and comment.

UML Class Diagram
The provided UML class diagram shows the relationships between the classes. `Main` uses `Controller`, which in turn uses `ApiHandler` to fetch gas station data. Both `ApiHandler` and `Main` use the `GasStation` class, which contains a list of `Review` objects.

API Integration
The application uses the Google Places API to retrieve information about gas stations. The API key is managed within the `ApiHandler` class.

User Interface
The UI is designed with JavaFX. It includes a text field for zip code input, a button to trigger the search, and a list view to display gas stations and select them for review submission. The `Main` class uses JavaFX layouts to arrange these components in a user-friendly manner.

Data Handling
The application does not store data persistently. Reviews and gas station data are only held in memory during runtime and are lost once the application is closed.

Building and Running

To build and run the application, you need a Java IDE that supports JavaFX. Compile the `Main.java` class and ensure that JavaFX libraries are included in the classpath.