**Program Design:**

The program prompts a user for a city name, or city zip code. The user is then prompted to enter a country where that city or zip code resides. The user is then prompted to choose whether to get current weather for that location or a 5 day forecast. If the input is valid, the program consumes a RESTful service and obtains weather data about the chosen city.

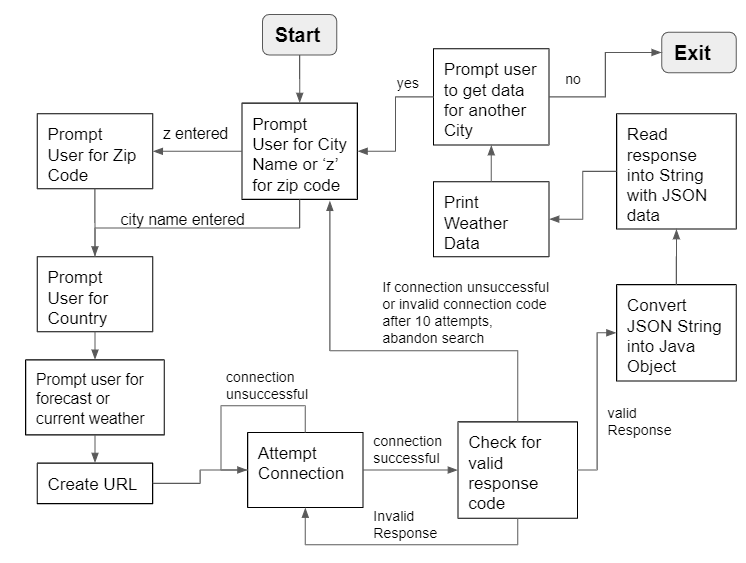
The program assembles a URL using the user input, then uses an HTTP request to obtain the weather data. The data is received in JSON format, but is converted to Java Objects using the GSON library.

If the weather data for the given city is successfully acquired, some of the data is displayed to the user. The data selected to be displayed is information that would be considered useful to someone that was interested in general weather data. To this end, some of the information received is not displayed to the user to help maintain relevance and clarity in the displayed output.

The program implements a robust system for handling valid http connections and responses that are not fulfilled by the web service. Connections and response code verification is implements in a loop. If a connection fails or a response code is not valid, the program attempts another connection. After each attempt, the program waits to allow server issues to resolve before attempting again. As the number of failed attempts increases, the wait time between reattempts also increases. This method allows the user to reasonably avoid interruptions due to a short term inability for a server to fulfill a valid request.

A limitation observed in the system is that zip (postal) codes for some countries that have more than one part and contain numbers and letters will only work by inputting the first half of the zip code. For instance, a Canada zip code ‘A1A 1A1’ will only work if ‘A1A’ is inputted for the zip code.

**Program Model:**

****