**Application**

This application uses RESTful APIs in order to provide the user with weather information for a city, and one other interesting fact. For this application it will provide the current forecast, the maximum and minimum temperature for the day, the wind speed, and what it currently feels like. Additionally the interesting fact it provides is the 3 closest Starbucks to the heart of the city so that if the user felt like they needed to get a hot drink on a cold day, or a nice and refreshing drink on a hot day they may locate their nearest Starbucks.

**Design**

This application uses OpenWeatherMap’s API to get the information on the weather, and it uses the GooglePlaces API to locate the Starbucks. The application first takes in a user input and does its API call to the OpenWeatherMap API, and if it is successful in connecting to the OpenWeatherMap API it will print the weather information and then it will take the latitude and longitude of the city.

Using the latitude and longitude of the city the program will then call GooglePlaces API to locate up to 20 Starbucks sorted by distance, but it does not specify the exact distance between the starbucks and the city. The program then will print the information of the 3 closest Starbucks and will use the haversine formula to calculate the exact distance in miles for the user.

In the event that the user is not able to connect to either API this program has retry logic implemented with exponential backoff. The retry logic first tries to connect to the API, and if it fails it will wait 1 second + randomNumberOfMilliseconds, then it will try again with the previous wait time doubled, up until the wait time is over 8 seconds for 1 wait. This means that users will have to wait a maximum of 7 to 9 seconds total. randomNumberOfMilliseconds is a value between 0 and 1000 (1 second). By adding a random number of milliseconds to wait along with the wait time it helps ensure if the service goes down it helps reduce the possibility of our application trying to connect at the exact same time as potential other applications which would cause the applications to cause a denial of service.

**Specifications**

* The application only takes in a city without a country name
* If the city is made up of two words it will concatenate the two words to make the city into one string
* If the city shares the name as another city the program will use the first city that is given by OpenWeatherMap’s API
* If the city is in the United States of America it will print the temperature in Fahrenheit otherwise the program will print out the temperature in Celsius.
* If we were not able to print out weather information for the city the program will not print out any nearby Starbucks
* The program supports characters from the roman alphabet and has not been tested on foreign characters.
  + If trying to find Starbucks in a foreign country such as Japan there is a possibility that non roman characters will be printed out as “??????”
* If there are not 3 Starbucks nearby it will print the most amount of starbucks it can, with some scenarios the application will not be able to find a Starbucks at all.