**Test Report**

*Weather App*

Created: 07/25/21 by Tom Lever

Updated: 07/25/21 by Tom Lever

**Introduction**

In developing Weather App, a database manager, an HTTP-request manager, an input manager, and a parser for a JSON object representing forecasts were developed. Each of these components of the Weather App is associated with a corresponding JUnit test class.

**Testing a Database Manager**

The test method of the test class for a database manager:

* Removes the Weather App Java project’s resources folder if it exists, including any database for a current zip code and saved zip codes.
* Creates a database manager, the resources folder, a database, a table for the current zip code, and a table for saved zip codes.
* Displays the table for the current zip code and the table for the saved zip codes.
* Checks valid and invalid zip codes.
* Displays the title and message of an exception that occurs when trying to delete a zip code from the table for saved zip codes that the table for saved zip codes does not contain.
* Inserts a zip code into the table for saved zip codes.
* Displays the table for saved zip codes and the table for the current zip code.
* Deletes a zip code from the table for saved zip codes.
* Displays the table for saved zip codes and the table for the current zip code.
* Displays the title and message of an exception that occurs when trying to get the current zip code when the current zip code has not been set.
* Displays the title and message of an exception that occurs when trying to update the current zip code with a zip code that is not in the table for saved zip codes.
* Inserts a zip code into the table for saved zip codes.
* Displays the table for saved zip codes and the table for the current zip code.
* Updates the current zip code with a zip code that is in the table for saved zip codes.
* Displays the table for saved zip codes and the table for the current zip code.
* Displays the current zip code when the current zip code has been set.
* Deletes a zip code from the table for saved zip codes, and removes that zip code from the table for the current zip code.
* Displays the table for saved zip codes and the table for the current zip code.

**Testing an HTTP-Request Manager**

The test method of the test class for an HTTP-request manager:

* Creates an HTTP-request manager.
* Displays a geographic position (i.e., longitude and latitude) corresponding to Charlottesville, VA and zip code 22903.
* Displays a geographic position corresponding to Lansdale, PA and zip code 19446.
* Displays the title and message of an exception that occurs when trying to get the geographic position for an invalid zip code.

**Testing an Input Manager**

The test method of the test class for an input manager:

* Removes the Weather App Java project’s resources folder if it exists, including any database for a current zip code and saved zip codes.
* Creates an input manager, a database manager, the resources folder, a database, a table for the current zip code, a table for saved zip codes, and an HTTP-request manager.
* Executes various commands to ensure that user input complies with the command menu, appropriate results of actions are displayed, and appropriate exception titles and messages are displayed.

**Testing a Parser for a JSON Object Representing Forecasts**

The test method of the test class for a parser for a JSON object representing forecasts:

* Creates a parser for a JSON object representing forecasts for Charlottesville, VA starting on the day of execution of the test method.
* Displays the JSON object.
* Displays forecasts derived from the JSON object for comparison.