The testIsValid() function is used to verify that a given URL is indeed a valid URL input. It does this by first calling a testIsValid() function that takes in two parameters. One is an array of objects and the second is a long data type value for options. It uses the parameters to check the validity of the URL given. The function iterates through each part of the URL (which is split into 5 sections: scheme, authority, port, path, and query). If any part of the URL is false, then the value of the boolean “expected” is false and the function will print out an “X”, indicating all instances where a URL fails. The function will continue this pattern until each part of each URL has been checked. After the testIsValid() function with parameters completes, a setUp() function is called that resets the testPartsIndex[] values to be 0.

URLs are built by systematically combining all members of five different sets which represent different fields or parts of a URL. The fields are scheme, authority, port, path, and query. There are a total of 538,650 URLs that are generated and tested for using this methodology. Both true and false URLs are evaluated against expected results.

Example valid URL: http://www.google.com:80/test1/test1?action=view

Example invalid URL: http:1.2.3.4.5:65a/../#?action=view

Real world tests would be both similar and different from what we wrote. The underlying concepts and applications are the same for real world tests and those we did in class, but the scale and complexity of systems being tested in the real world are significantly higher than anything that we worked with. The real world tests seem significantly more thorough, for example - systematically generating thousands of possible test inputs to cover a huge variety of potential errors.