

## **CS 362 - 400 Final Project - Part B**

Group Members: Andrius Kelly, Charlotte Murphy, Mark Ruoff

GitHub Repository:

[https://github.com/kellandrosu/CS362-004-F2017/blob/final\\_project/kellandr/FinalProject/URLValidatorInCorrect/src/UrlValidatorTest.java](https://github.com/kellandrosu/CS362-004-F2017/blob/final_project/kellandr/FinalProject/URLValidatorInCorrect/src/UrlValidatorTest.java)

### **Methodology:**

Manual testing and unit testing are the two methodologies we employed.

We each chose manual test inputs to cover a variety of URL addresses. Once we completed manual testing, we more closely examined the range of valid inputs for each part of a URL address to determine input partitions. The input partition data and the errors discovered during manual testing then provided a guide for designing effective unit tests to methodically test the full range of valid inputs, as well as invalid inputs.

Sample of manual test urls:

```
http://amazon.com?action=view
http://0.0.0.0?action=edit&mode=up
Amazon.com
http://256.256.256.256
ftp://0.0.0.0:80/test1?action=view
http://go.au:0/test1
http://go.au:65a/test1
http://go.au/./file
```

Rather than write three separate unit tests that could potentially duplicate testing efforts and results, we opted to design a single unit test to evaluate every possible combination of URL test parts. The partition data below served as a guide in determining a set of valid and invalid inputs for each URL part.

### **Partitioning:**

The partitioning is based on the separate parts that together compose a URL address: scheme, two slashes, hostname, port, path, query, fragment identifier. We selected unit tests near partition boundaries as much as possible, as well as some close to the middle, if possible. For example, we manually tested IP address 255.255.255.255, IP address 256.256.256.256 and 192.168.0.1 with an expected true.

We researched valid and invalid inputs for the various URL components and collected our findings below:

- Scheme / Protocol
  - Examples: http(s), ftp, mailto, file, data, irc

- Two slashes (//): This is required by some schemes and not required by some others. When the authority component (explained below) is absent, the path component cannot begin with two slashes.
- Hostname
  - Domain name: ([www.example.com](http://www.example.com))
    - Each label must be from 1 to 63 characters long,[2] and the entire hostname (including the delimiting dots but not a trailing dot) has a maximum of 253 ASCII characters
    - may contain only the ASCII letters 'a' through 'z' (in a case-insensitive manner), the digits '0' through '9', and the minus sign ('-').
  - IPv4 - 0.0.0.0 to 255.255.255.255
  - IPv6 - **Note:** We search but couldn't find a working example to concretely test a working IPv6 address, so we chose not to test for it.
- Port numbers
  - Port numbers range from 0 to 65535, but only port numbers 0 to 1023 are reserved for privileged services and designated as well-known ports
- Path / Filename
  - must either be empty or begin with a slash ("/") character.
  - [Valid Characters](#)
- ? query
  - Delimiters either & or ;
  - [Valid Characters](#)
- [Fragment Identifier](#)
  - <http://www.example.org/foo.html#bar>
    - the fragment refers to the element with id="bar".

## **Bug Report:**

### **Manual Test Cases:**

True test cases that should return false

- Bug in IP address
  - <http://256.256.256.256>

False test cases that should return true

- Port number returns false
  - <http://www.home.com:1234>
- Bug in query
  - <http://amazon.com?action=view>
  - <http://amazon.com/?action=view>
  - <http://0.0.0.0?action=edit&mode=up>
- Bug in scheme
  - Amazon.com

## **Debugging:**

### Bug #1

Test case: <http://256.256.256.256>

Expected result: False      Actual result: True

URL Validator Bug with IP addresses is located in InetAddressValidator at line 96. The statement (if(ilPsegment > 255) goes to a return true statement. Per the function description true indicates valid. This should return false instead. To locate the bug placed a breakpoint, using the Eclipse IDE debugging tools, at the function call with the invalid ip address url and followed it through function calls by stepping into each function.

### Bug #2

Test Case: <http://amazon.com?action=view>

Expected result: True      Actual result: False

URLValidator Bug with query string appears to be located in URLValidator.java at lines 314 and 315. When sent a valid URL query, it is returning false in this statement, instead of the intended True Statement. Followed similar steps to locate the bug as before using breakpoints and the IDE debugging tools.

### Bug #3

Test Case: <http://www.home.com:1234>

Expected result: True      Actual result: False

URLValidator Bug for port numbers is located in lines 393 and 394 of the URLValidator.java code. When reaching this point in the code with a URL Port of 1000, the code is returning a false statement instead of the intended true statement in this area.

## Bug Reports:

Title: URLValidator incorrectly returns false for port numbers above 999

Class: Serious Bug

Date: November 21, 2017

Reported By: Andrius Kelly, Charlotte Murphy, Mark Ruoff

Email: [kellandr@oregonstate.edu](mailto:kellandr@oregonstate.edu)

Is it reproducible: Yes

Description: URLValidator incorrectly returns false for port numbers above 999

Steps to Produce:

```
UrlValidator urlVal = new UrlValidator(null, null, UrlValidator.ALLOW_ALL_SCHEMES);  
System.out.println (urlVal.isValid(http://www.home.com:1000) );
```

Expected Results: Prints true

Actual Results: Prints false

---

Title: URLValidator incorrectly returns true for IP addresses with units above 255

Class: Serious Bug

Date: November 28, 2017

Reported By: Andrius Kelly, Charlotte Murphy, Mark Ruoff

Email: [ruoffm@oregonstate.edu](mailto:ruoffm@oregonstate.edu)

Description: URLValidator incorrectly returns true for IP addresses with units above 255

Steps to Produce:

```
System.out.println(urlVal.isValid("http://256.256.256.256"));
```

Expected Results: Print False

Actual Results: Print True

---

Title: URLValidator does not return valid response for valid URL query Statements

Class: Serious Bug

Data: November 28, 2017

Reported By: Andrius Kelly, Charlotte Murphy, Mark Ruoff

Email: [ruoffm@oregonstate.edu](mailto:ruoffm@oregonstate.edu)

Description: When a URL query is added to a url the URLvalidator incorrectly returns false

Steps to Produce:

```
System.out.println(urlVal.isValid("http://amazon.com?action=view"));
```

Expected Results: Print True

Actual Results: Print False

Sources:

[Wikipedia](#)

<https://tools.ietf.org/html/rfc3986>

[https://en.wikipedia.org/wiki/List\\_of\\_Internet\\_top-level\\_domains](https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains)

[https://www.webopedia.com/quick\\_ref/portnumbers.asp](https://www.webopedia.com/quick_ref/portnumbers.asp)

[https://en.wikipedia.org/wiki/Query\\_string](https://en.wikipedia.org/wiki/Query_string)

[http://www.gestioip.net/docu/ipv6\\_address\\_examples.html](http://www.gestioip.net/docu/ipv6_address_examples.html)

### **Teamwork**

Work was completed by meeting via google hangouts and collaborate using google drive. Up until the unit tests all portions were worked on in a group setting. For the unit tests, Andrius and Mark split up the partitioning and Charlotte worked on the loop that brought them together. After the individual portions were worked the group meet again via google hangouts and combined the work and did final debugging.