Methodology Testing:

Manual Testing Functions:

* testManualTest()

Partition Testing Functions:

* testschemePartition()
* testdeliminatorsPartition()
* testportsPartition()
* testpathsPartition()

Programming Testing Functions:

* TBD

For manual testing we wanted to see the implications of a few variables. Firstly we wanted to tested how isValid treated 3 different cases of schemes. Firstly with a custom pasted in list (using otherwise valid schemes), the default (so no parameters passed into the function for scheme), and lastly the allow all flag scheme. All in we only tested 3 different URLs but with different versions of those parameters. Our URLs are http:www.google.com, <https://www.google.com> and <http://kljadklualdfsdlfiadsf> .

For partition testing, initially we tried to use asserts however since that produced errors and halted testing after the first such error in the function we switched to a try/assert/catch methodology to actually see what tests were passing and which were failing. The first function is testschemePartition(). All use the address of [www.google.com](http://www.google.com), but we tested a few known good schemes (also we were using allow all schemes (so any should have been valid) such as http, https, and ftp. We then tested some garbage input sets of non valid schemes. With testdeliminatorsPartition(), we tested various delimitators to <http://www.google.com>. Again we separated them into two arrays, one with valid delimitators and one with garbage inputs. With testportsPartition we tested the values at the bottom/middle/top of the valid port range then tested out of bounds and otherwise invalid ports.

Bug Report:

Bug 1: The case of ALLOW\_ALL and http

1. The failure is that any non http scheme URL and without ALLOW\_ALL being set as a flag will come back as invalid
2. We detected early on during random manual testing and then refined our manual test to isolate it further.
3. On line 318 when an http scheme is used it bypases the isValidAuthority function defaulting to it being a valid authority. This is coupled with the fact that the UrLValidator was modified to uppercase all passed in values (so http becomes HTTP, and then http != HTTP when doing a later comparrison, and the scheme would come back invalid). A close up of a sign
   
   Description automatically generated