Integrity checker report

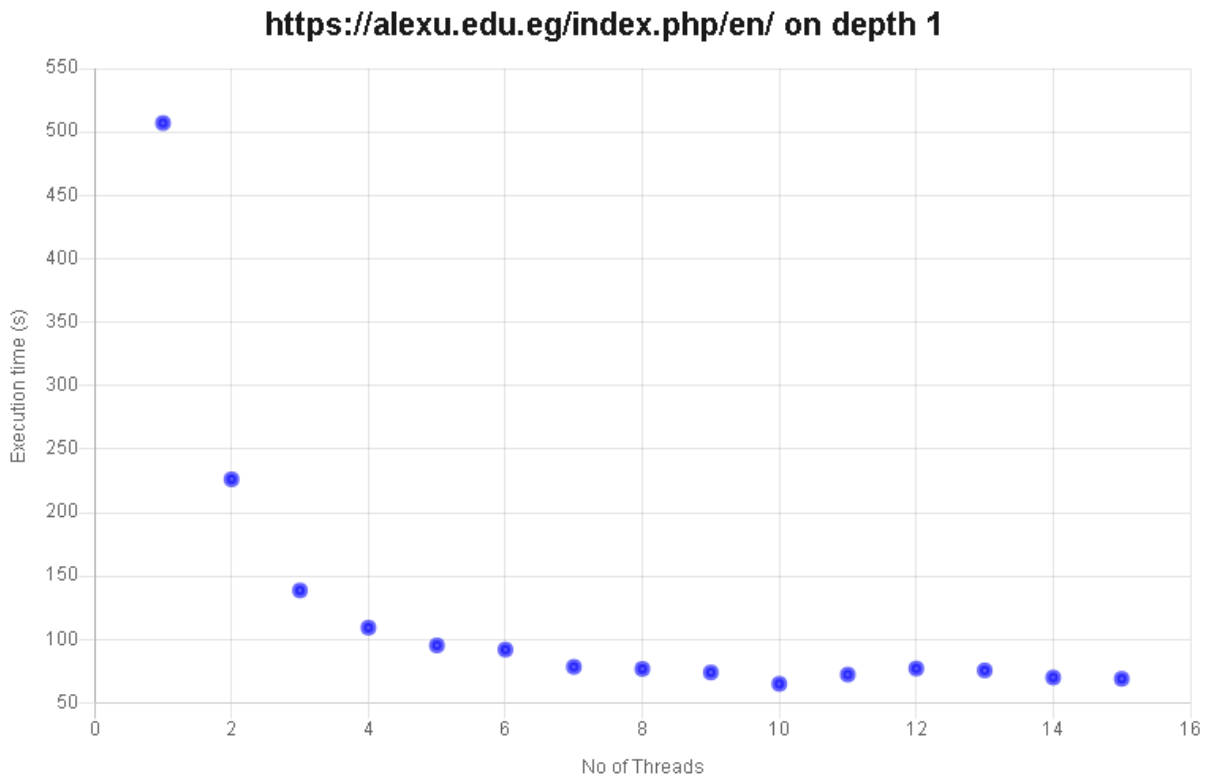
Implementation:

We used very simple implementation to make the program as simple and readable as possible while making it functionally complete. We start by taking the input from the user as a string for the initial URL and an int for the depth. Then we parse that URL and get all URLsin that page and add them to a queue. After that we create a static number of threads, 10 in our case, and they all do the same thing: get URLs from the queue and check if they're valid (if the depth allows it, also parse and get URLs). After all threads complete, the program displays execution time data along with the final count of valid and invalid URLs.

To find the optimal number of threads to use, we tested the University website at a depth of 1 with different numbers of threads as shown below:

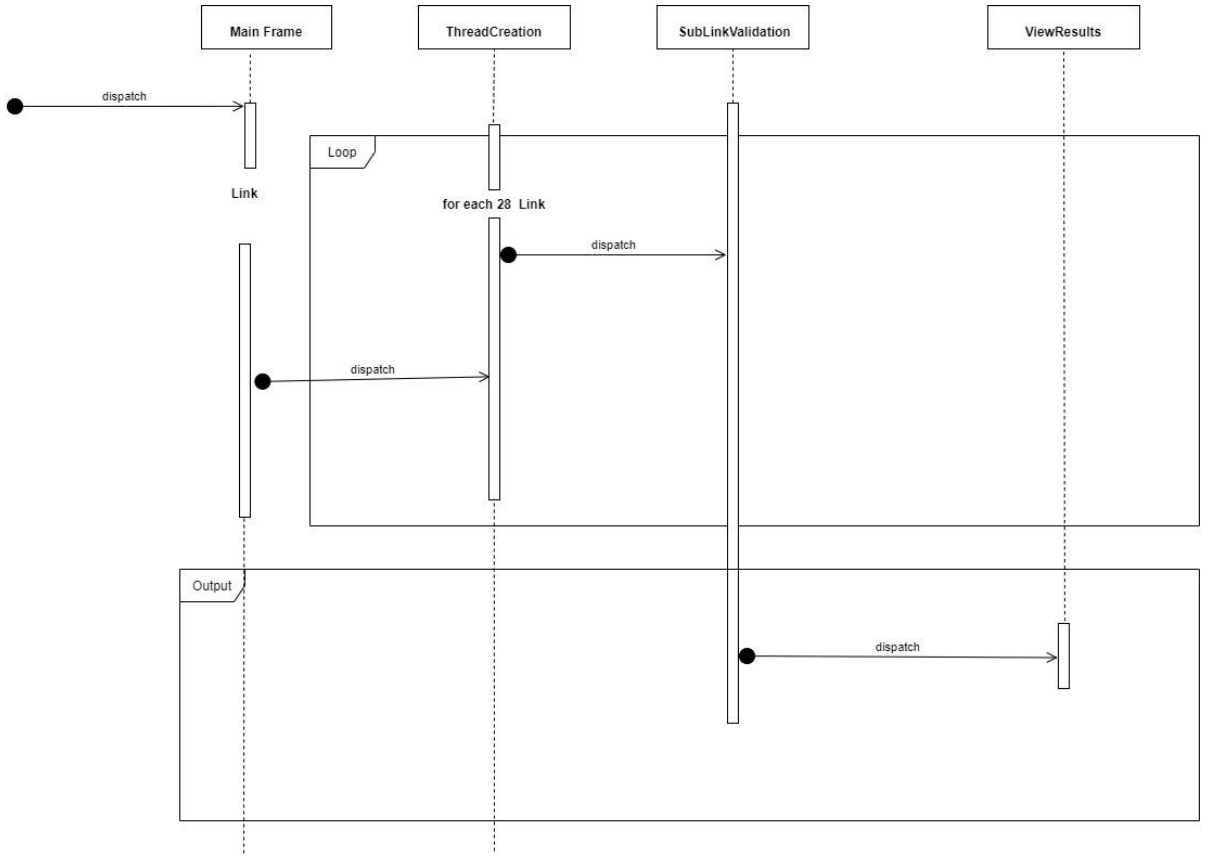
No of Threads	Execution time (s)
1	506.9
2	226.2
3	138.6
4	109.3
5	95.3
6	92

7	78.4
8	76.8
9	74
10	65
11	72.3
12	77.1
13	75.6
14	70
15	69

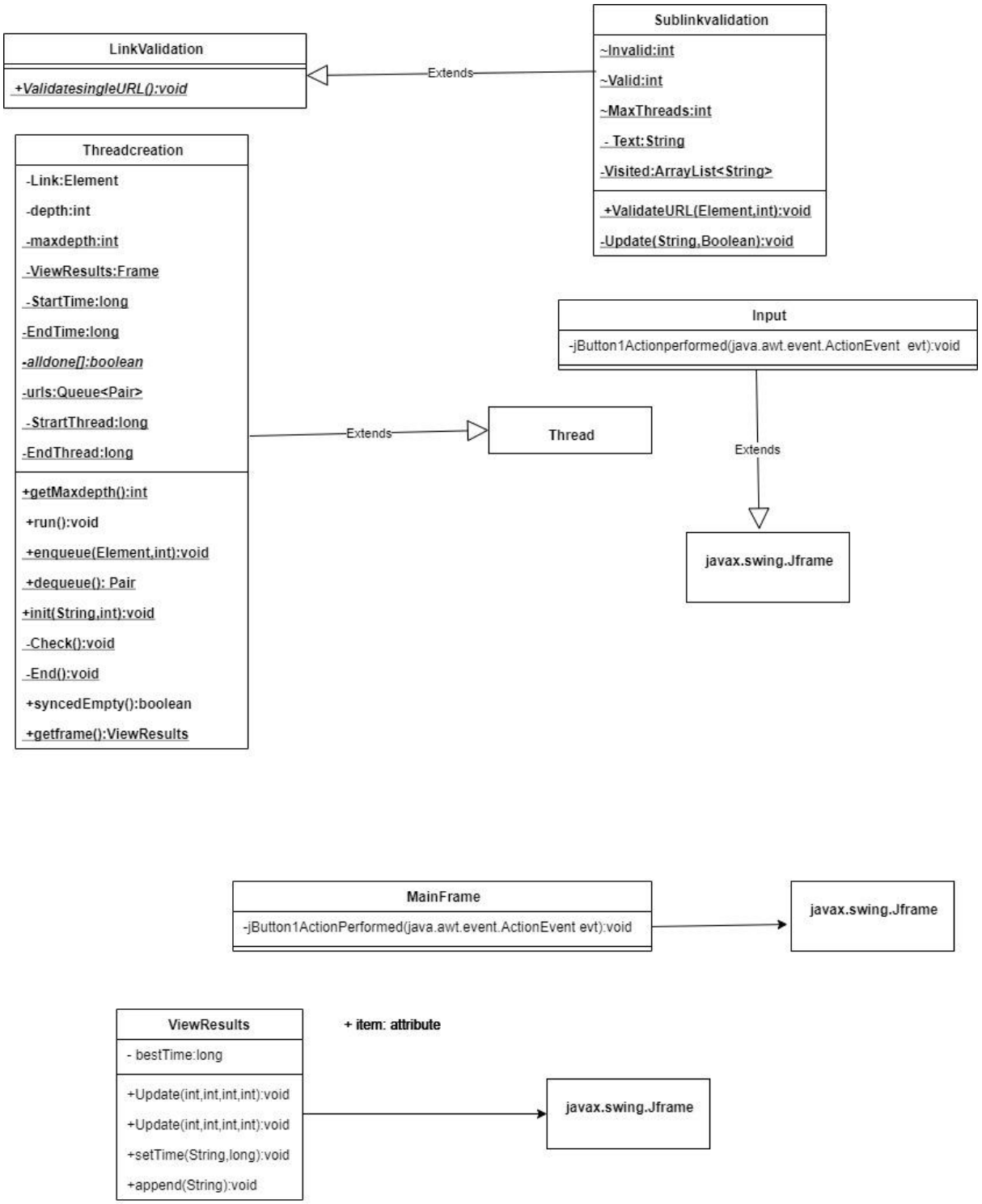


To show our program functionality, we designed these UML diagrams:

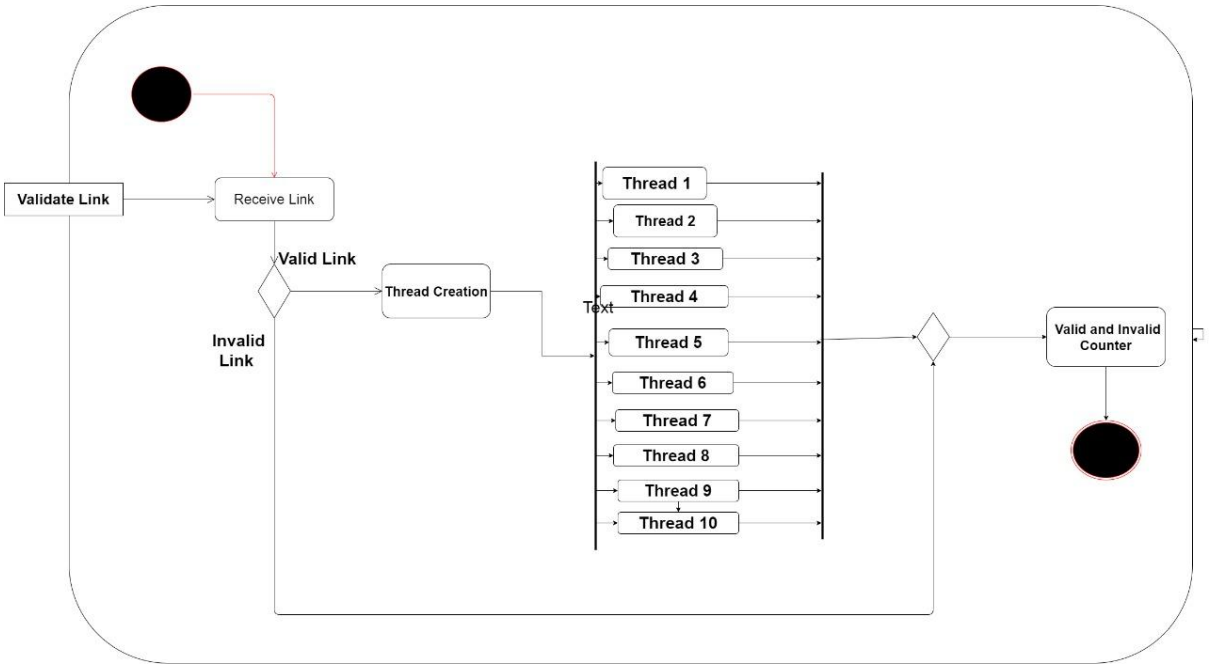
Sequence Diagram:



Class Diagram:



Activity Diagram:



Use Case Diagram: