**REQUIREMENTS AND MAIN USE CASES**

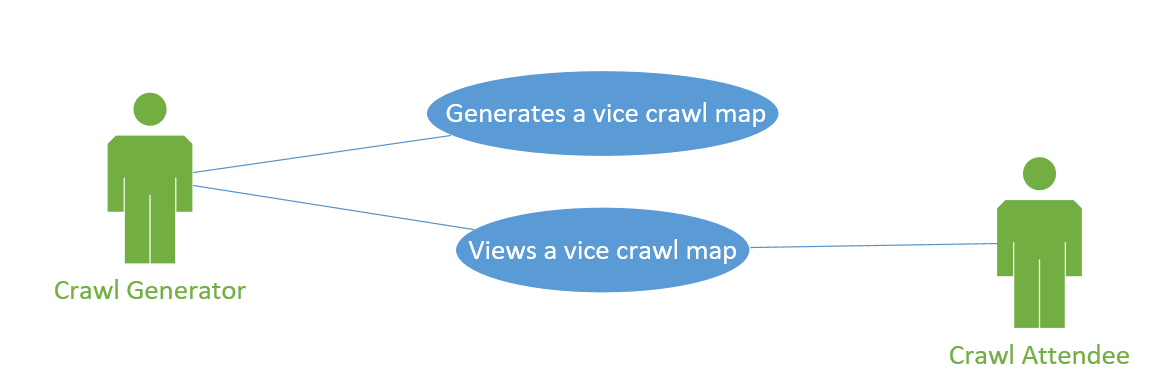
**TOP 5 FUNCTIONAL REQUIREMENTS:**

1. A user should be able to input a list of stops.
2. The system should generate a path between the stops.
3. The path should be feasible and optimized.
4. The path should have clickable detail view for each stop.
5. The path should also display distance and time between each stop.
6. A user should be able to view another users crawl.
7. A user should be able to view past crawls.
8. A user should be able to edit crawls
9. A user should be able to re-use past crawls.
10. A user should be able to send crawls to other users.
11. A user should be able to receive crawls from other users.

**TOP 5 NON-FUNCTIONAL REQUIREMENTS:**

1. The program must be written in Javascript.
2. The program will be used as a web app.
3. The program cannot make you violate local laws or safety (trespassing).
4. Process development will initially using the waterfall method and change to agile method.
5. The path must be generated within a few seconds (user acceptability).
6. The path must be generated with minimal backtracking as the first priority.
7. The program must use Google Maps API.
8. The program must be easy to use.
9. The program should minimize storage space used.
10. The program should be constantly available.
11. The program should provide consistent, mostly unchanging, results.
12. The program should be functional across most browsers.

**MAIN USE CASES:**

****

A crawl generator can input a list of places. A crawl will be generated. The crawl generator can publish the crawl. The crawl attendee can view the crawl.