**The Carlin Index: System Requirements**

The system requirements have been divided into three main sections:

1. API Requirements

2. Front End Requirements

3. Monitoring and Infrastructure Requirements

Details for these sections are outlined below, respectively.

**API Requirements:**

**1. Calculate Carlin Index by User:**

A. Take in username

i. Handle invalid requests with HTTP status code and error message in JSON

* + - ALL (we should handle all invalid input, but I wanted to list special cases that the doc called out.)
    - Requests to private user called out in doc

B. Query the user’s feed and measure tweets that contain any of the 7 words within the given duration.

-The seven words to query are: shit, piss, cunt, fuck, cocksucker, motherfucker, and tits.

C. Return username, associated Carlin index, and top tweet

**2. Create a Leader Board:**

A. Store 10 highest scoring twitter users in a DB

B. Create an API call that returns the leader board in JSON format

C. Update the leader board user list when a dirtier user is found

D. Display the leader board with front end

**3. Database:**

A. Store Leader Board list of users

B. Cache Daily Carlin Index (refresh this no more than hourly)

**Front End Interface Requirements:**

**1. Single Web page**

**2. Username input field**

**3. Asynchronous page load (NO RELOADING THE PAGE)**

**4. Leader Board:**

1. Show the top 10 dirtiest users

2. If the most recent search yields a new entry to leaderboard, update the board.

**Monitoring and Infrastructure Requirements:**

**1. Host application on EC2 Small Instance**

**2. Host the DB on Amazon RDS**

**3. Set up elastic load balancer:**

A. Launch new AMI when EC2 has CPU or memory exceeding 50% for 2 mins

B. Email when any EC2 is above 50% cpu or memory usage (no duration, any time this happens... Seems like this requirement should have a time period. Seems like spam is imminent.)

C. If All EC2 instances are below 50% cpu and memory for 5 mins then decommission the instance from the load balancer

D. Terminate the EC2 instances that have been decommissioned.

**4. Monitoring Web Page:**

A. Entire web page is green when API is responding and red when API call times out

B. Poll the API every X seconds. X is up to developer discretion.

**5. Use Git:**

A. Host git repo on github

B. Pull down latest on all EC2 instances when updates are detected

C. relaunch the application on all EC2 instances when updates occur

D. STRETCH GOAL: Rolling restarts (app is never down for restarts)