Date-A-Dog Code Review

Doing code review was more complicated than we had imagined, particularly because everyone who had experience with that specific language/package had already worked with that piece of the application. It was also challenging to look at code that someone else had written and try to make it more efficient without knowing the ins and outs of the algorithm.

For code review, the following class was analyzed:

DADAPI.java located in app/src/main/java/dateadog/dateadog

The following changes were made:

First iteration:

1. Line 87: Character limit exceeds 80 with a total length of 119. Recommended splitting up into two different lines.
   Reason: Coding convention typically specifies 80, for readability and to minimize errors. Change rejected because we are following the Android Studio Project convention for this part of our code.

2. Line 165: Parametrize the function such that it can accept a count (number of dogs to receive) and zip (zipcode of shelter) and if those are not specified, then use the defaults.
   Reason: If we do implement the stretch feature of allowing dogs from different zip codes to populate the application, the code is modular. Change accepted.

3. Line 234: The function getDateRequests() has a lot of string literals that should be made into static constants.
   Reason: For ease of modification and to follow convention. Change accepted.

Second iteration:

4. Line 94 & 261: The method makeRequest() still contains string literals. Developer advised to change them to constants.
   Reason: See 2.
   Change rejected because they do not improve readability and the literals in question are never used again.

5. Line 174: The comment for the method getDogsAtURL() does not clearly state what the given callback listener is. The developer is recommended to add more comments.
   Reason: Current status lacks clarity. Change accepted.

Doing code review helped us to see just how difficult it is to accurately perform a helpful code review. It requires experience with the particular language being implemented as well as understanding the underlying logic of the algorithms being reviewed. You also have to be able to get the original writer of the code to agree with your recommended changes, or disagree if your recommended changes are not necessary. In addition, we needed to ensure that we had enough time to implement the requested changes and to ensure that no new bugs were introduced or tests were broken after the code review was completed.

The following commits from Fri Dec 2 19:47:42 2016 -0800 to Fri Dec 2 20:20:27 2016 -0800 with the following hashes are for the code review described above:

3f95bdea35ccd2f574c3fefa1a30c92015128cee

9817cdc014bfdd136d2d2f622405a9f21bff0b29

bd848897d2e93946747fbb2de959e4838fc35076

55f6676c84cbc5e978527b26145d100aa6339986

4346322e3ebbe13fdcb41f54d07499ffdd986c50