**Software/Systems Design**

2.1 Design a data refresh system

*Overview*

The system must receive from the user a location in the form of a mobile GPS fix (latitude, longitude, accuracy) that represents a place the user has visited. Using the Foursquare search venues API call, the system looks up information about the place and outputs it to the user. The results are cached so that if the place is visited again, the information can be retrieved without having to repeat the API call. Cached items are deleted every thirty days, requiring results to be refreshed.

EXTERNAL PARAMETERS

The parameters needed to call the API are Client\_ID, Client\_Secret, and Version. These can be requested from Foursquare.

Client\_ID

Client\_Secret

Version

REQUEST.FORM

The application receives Latitude, Longitude, and Accuracy information from a GPS sensor.

Latitude (e.g., 40.66 N)

Longitude (e.g., 73.99 W)

Accuracy (e.g., 5 m)

CACHE

The cache is defined by Expired\_after, the number of days after creation a data item must be deleted to prevent accessing stale data.

Expire\_after (e.g., 30 days)

*Authentication*

As the venue data are public, it is possible to do userless venue searches. In that case, OAuth authentication does not have to be used and a client ID, client secret, and version parameters can be used in the endpoint. If the user wants to use his credentials, he has to first be authenticated using OAuth, which requires performing the necessary handshaking between the application and the Foursquare servers. Rate limits are more favorable for users. Each user is allowed to make 500 requests per hour, whereas 5,000 userless requests per hour are allowed for the whole application.

*API Call*

First a URL for Latitude, Longitude, and Accuracy is generated. A check is made to see if the URL is already stored in cache. If so, the JSON response for that URL is retrieved from cache. If not, an HTTP request on the URL is performed to search the Venues database to generate the JSON response and the URL and response combination are cached.

*Output*

The JSON string is parsed and the results are displayed to the user.

Sensor

Web Server