**Noun Identification Technique**

Step 1: Identifying all nouns

We have some customer records in a text file (customers.json) -- one customer per line, JSON-encoded. We want to invite any customer within 100km of our Dublin office for some food and drinks on us. Write a program that will read the full list of customers and output the names and user ids of matching customers (within 100km), sorted by User ID (ascending).

You can use the first formula from this Wikipedia article to calculate distance. Don't forget, you'll need to convert degrees to radians.

The GPS coordinates for our Dublin office are 53.339428, -6.257664

Step 2: Determining quality of candidates

Duplicate nouns and low-quality candidates will be removed. Their quality is determined by their usefulness as a class in the context of this project. Class names are also derived from other nouns. This leaves the following:

Customer (will have members “userId” and “name”)

JSONParser (derived from “JSON-encoded”)

GreatCircleDistance (derived from the noun “formula”)

Coordinate

This classes will be used as a foundation the build the rest of the project around.

General Notes: config/config.properties is used since in a live system, it would be wise to have the Dublin Office coordinates be easy modifiable in case Intercom moved offices to a different location in Dublin.