The Google Distance – Documentation

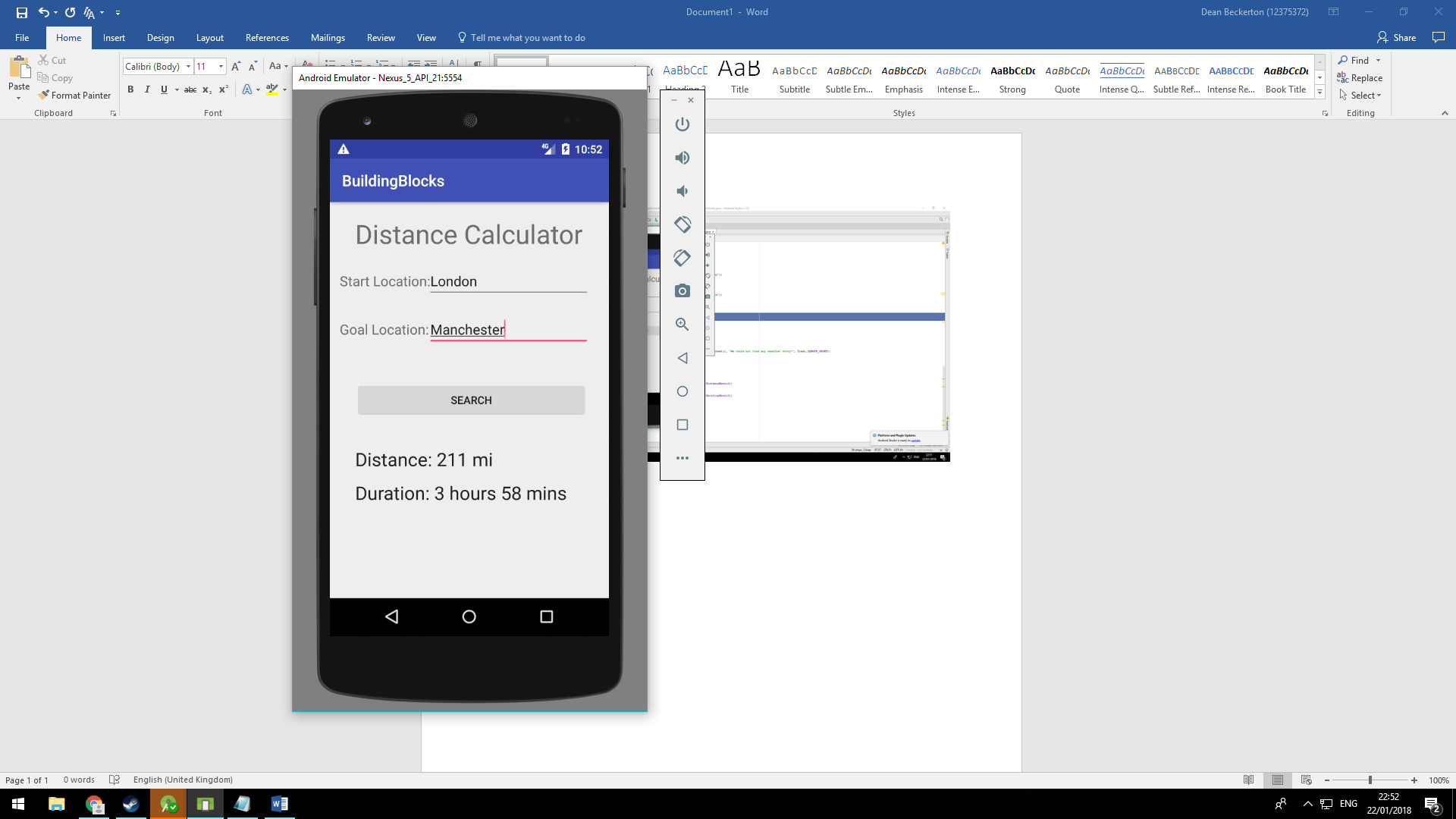
Dean Beckerton (21/01/2018)

**Requirements**

* Android Studio **(**[**https://developer.android.com/studio/install.html**](https://developer.android.com/studio/install.html)**)**
* Android Device or Emulator to run the application (lollipop 5.1 or above)
* WinRAR or another software to Unzip the file.

**Overview -** Google's Distance Matrix service computes travel distance and journey duration between multiple origins and destinations using a given mode of travel. This service does not return detailed route information. Route information, including polylines and textual directions, can be obtained by passing the desired single origin and destination to the Directions Service.

**Solution** – I decided it would be most appropriate to develop this application on a mobile platform (Android), this was due to the advantage of having the ability to have access to this application while out of office which could be likely when dealing with themes such as location to location.



**Main Activity. JAVA**

**Address** - MyApplication/app/src/main/java/com/example/mobileassignment2/buildingblocks/

Functionality

1. **Command Button** (Start) – This will firstly complete initial error handling by checking if the user has inputted any data into the text fields (Start, Goal) on the condition it fails it will prompt the user with a message box. Otherwise it will begin an Async Task to begin the service call and at the same time start a progress bar spinning by making it visible.
2. **Async Task** – This will begin by taking the values inputted within the two input text fields (start, goal) and use them within the service call. At this point the httpconnect class will then be called, this represents the HTTP, get request and error handling (200,201) once completed this will then retrieve data in the form of JSON. To make this data more meaningful I would suggest using a tool to format (<https://jsoneditoronline.org/>) from here the JSON can be broken down object by object, until you reach the meaningful data you wish to use, here is an example “*JSONArray array = object.getJSONArray("rows");*” This will eventually lead to a Post Execute, here the data is converted to a string and outputted to the labels and will also disable the progress bar, making it invisible.
3. **Reset** – This is a simple void method that resets all meaningful values to null, allowing the user to repeat the process with a reduced likelihood of failure.