# **Technical Manual for Distance and Directions with Google Maps**

# By Greg Bayne, Kari Franklin, Michael Booser

**1. Overview**

This document describes how a system must be setup in order to be able to run the Google Maps program.

**2. System Requirements**

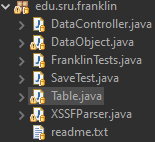
* Java version 1.4.0
* Memory 512 MB
* Free disk space 300 MB
* Processor speed 800 Mhz

**3. Assumptions**

* It is assumed the user has met all the minimum system requirements
* It is assumed the user has an active internet connection.

**4. Persistence**

The package edu.sru.franklin contains classes that are used to save the distance and directions between a set of two locations.



The distance and directions controller class uses the DataController class from this package to do this.

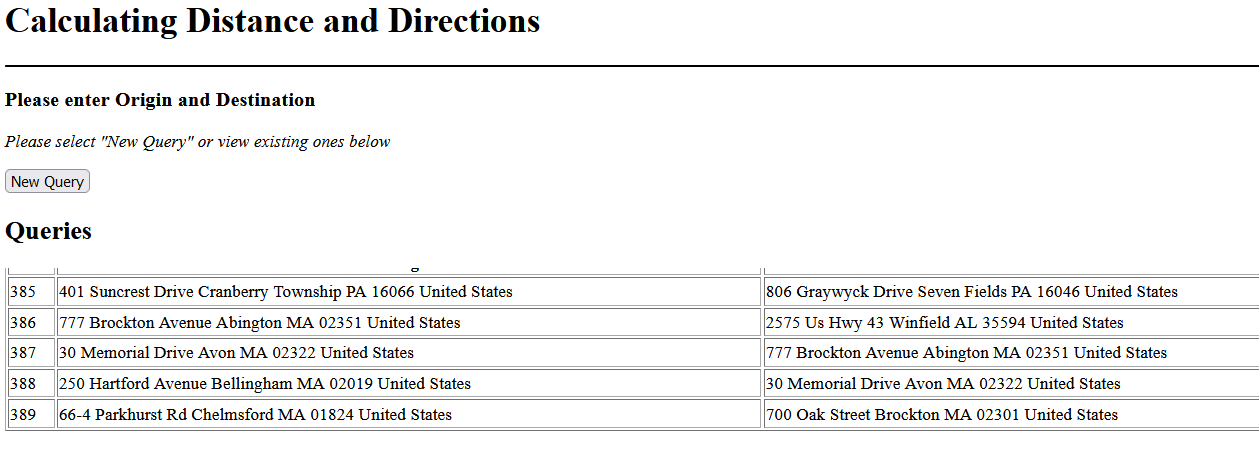
The package has classes to read and write the sets of locations.

1. Run the DistanceandDirectionsController.java



This will launch the spring application.

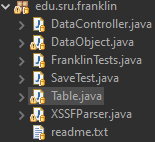
2. Go to “<http://localhost:8080/>” and add a new query you want to use.

3. Whenever you add a new query this will be saved to the hashtable. Every entry that is added to the table is then added to the data.txt document.

3.1 If the query already exists it will just pull that information from the table. If not, it will always make an API call and add it to the table.



4. In order to check what is being saved just go to edu.sru.franklin package and launch the SaveTest.java class and you will see any entries you have added.



5. This will show the contents of the excel file as well as any new or old queries you added.



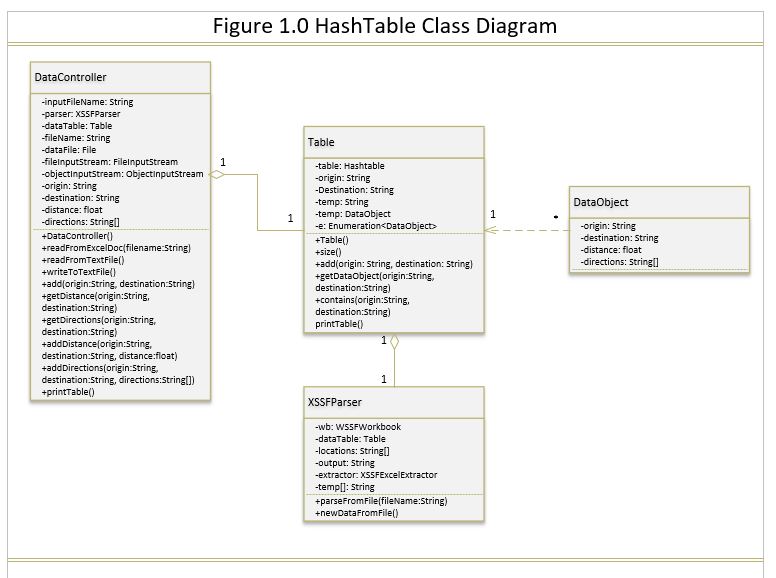
4. Running Google Maps

5. Google Maps Classes

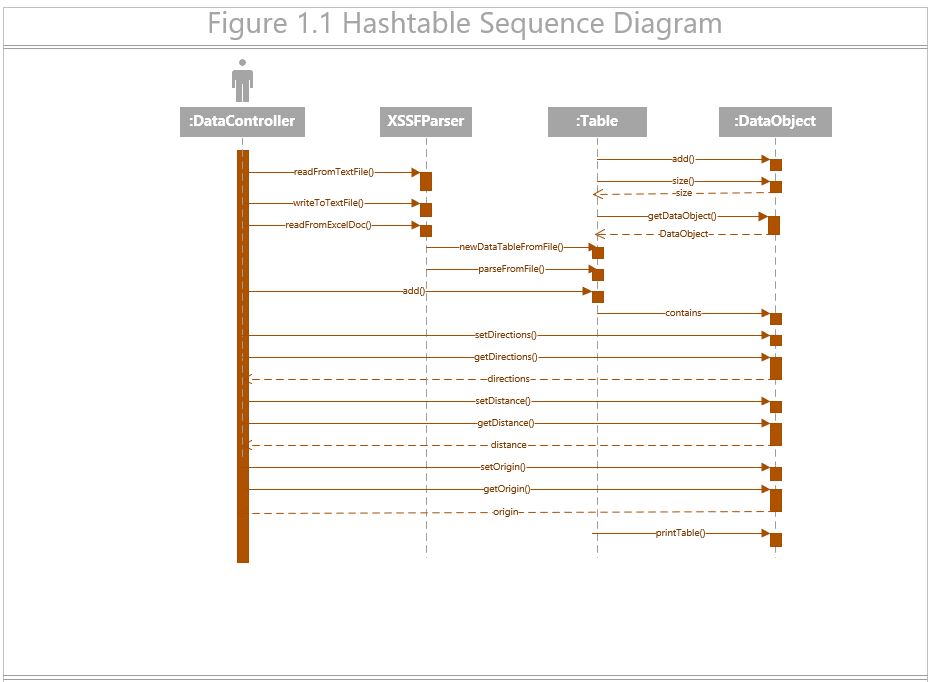
6. Interactions between Google Maps Classes

7. UML Diagrams on functionality of Google Maps

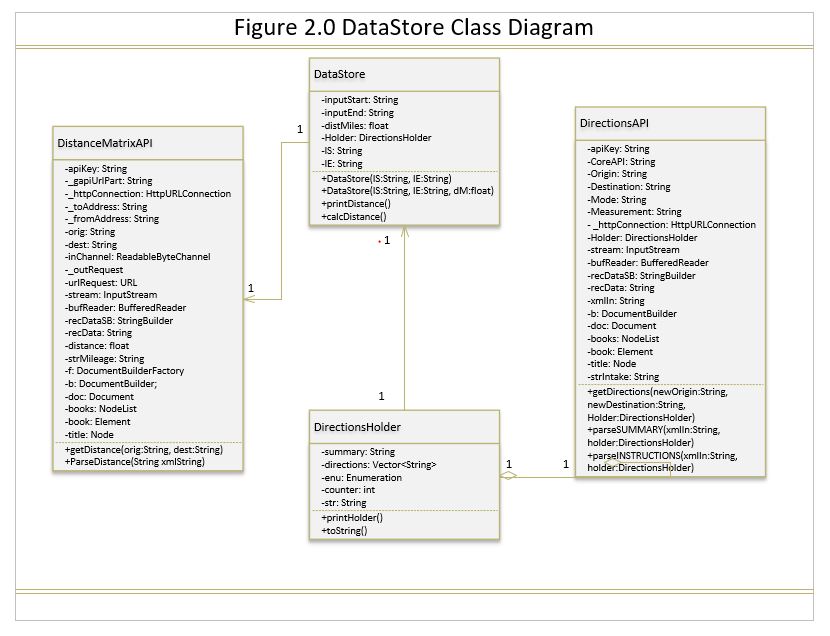
The following UML diagrams will show the interaction between different classes, how they are being used, and their functionality together as a whole.

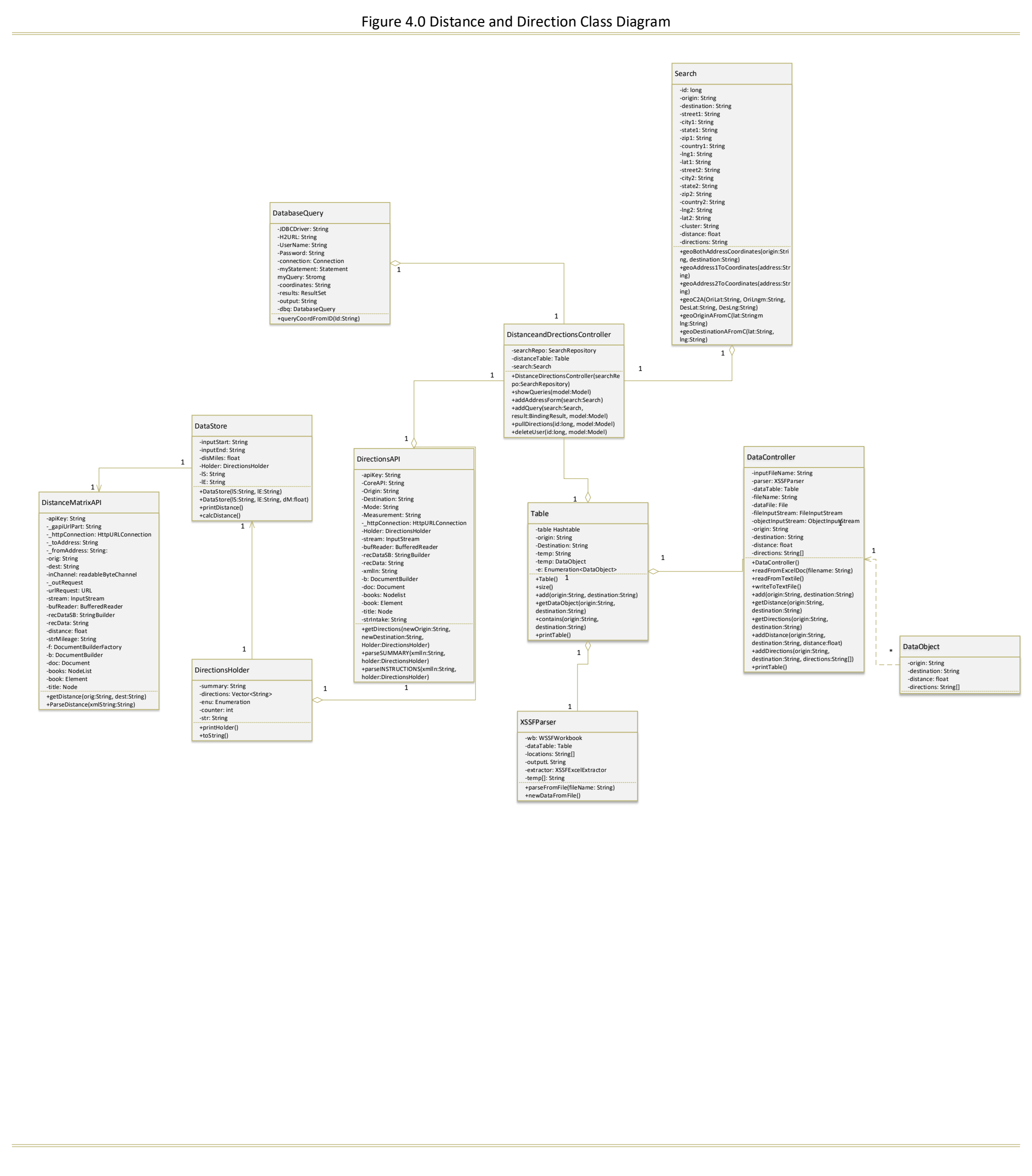


This UML class diagram shows the interactions of classes that are used to create and use the HashTable data structure. The Hashtable is what is used to load and save data for the project. There is a Table class that sets up the table, an XSSFParser that allows reading and writing to a text file. A DataObject that makes getters and setters for the Hashtable info and finally a DataController class that is used to implement all these classes together. Lastly, this shows the associations between these classes.

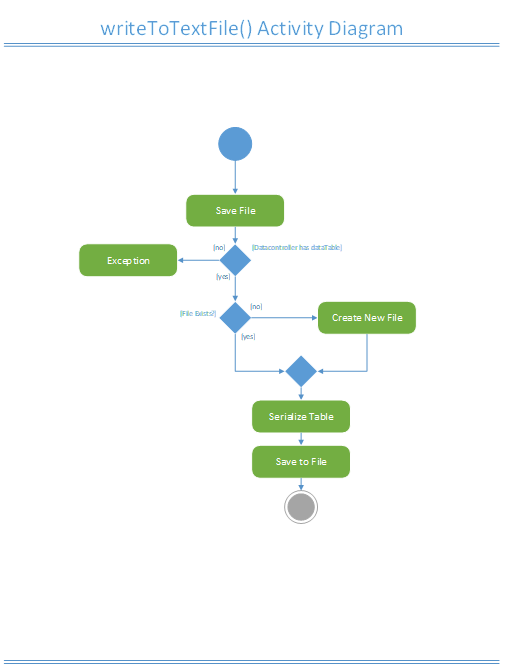


This UML sequence diagram shows the sequence of events existing between these four classes. This displays all the methods that are being used in each class as well as where they are being used and how they’re being used based on each class interaction. This diagram shows the sequence of information being used for the HashTable.

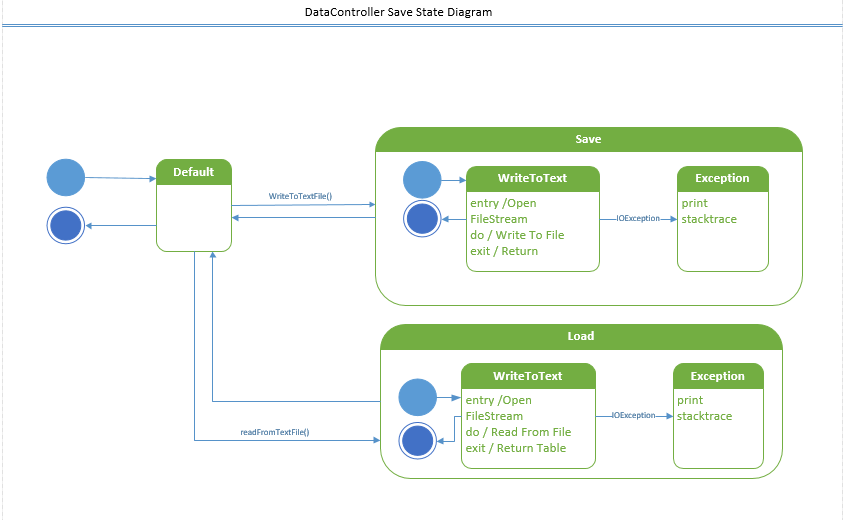




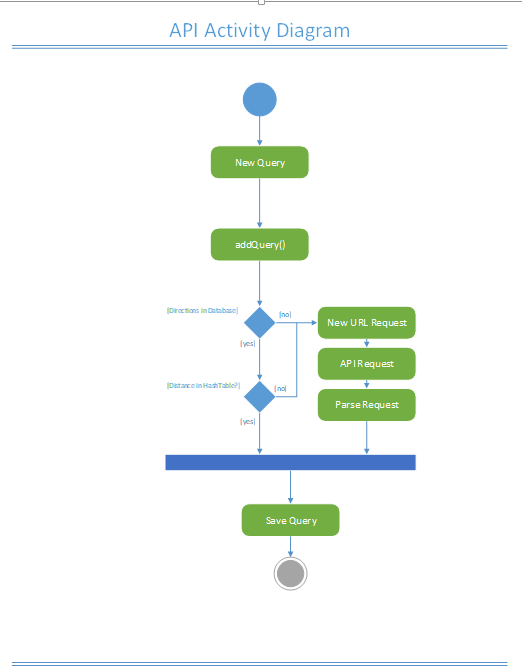
This UML class diagram shows the interactions of all the important classes in the program.



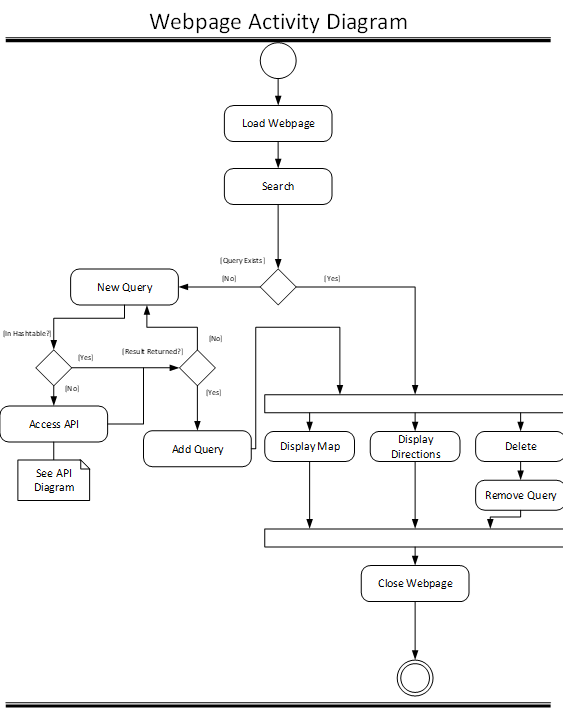
This is an activity UML diagrams that shows the different states involved in writing to a text file as well as the different steps towards saving it.



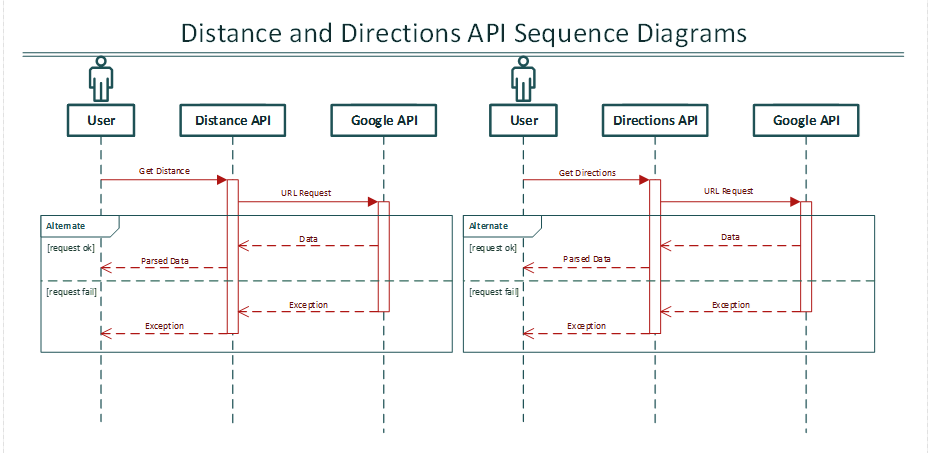
This is a state UML diagram that shows how the loading and saving works in the DataController class for the program.

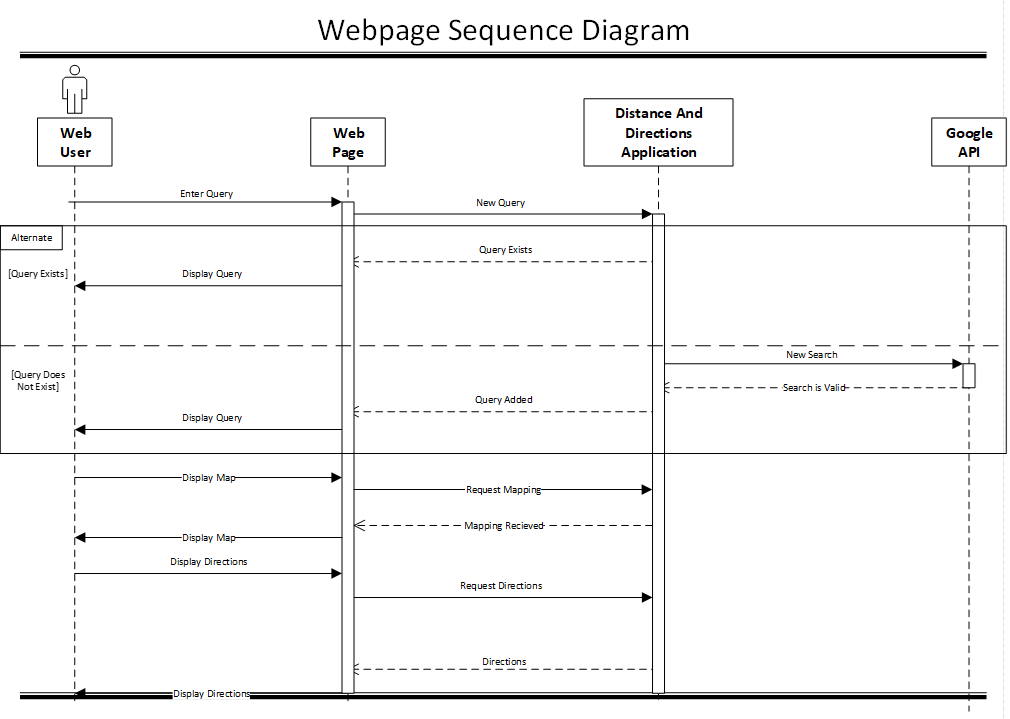


This is a activity UML diagram that shows the steps towards creating a new query and saving the query.

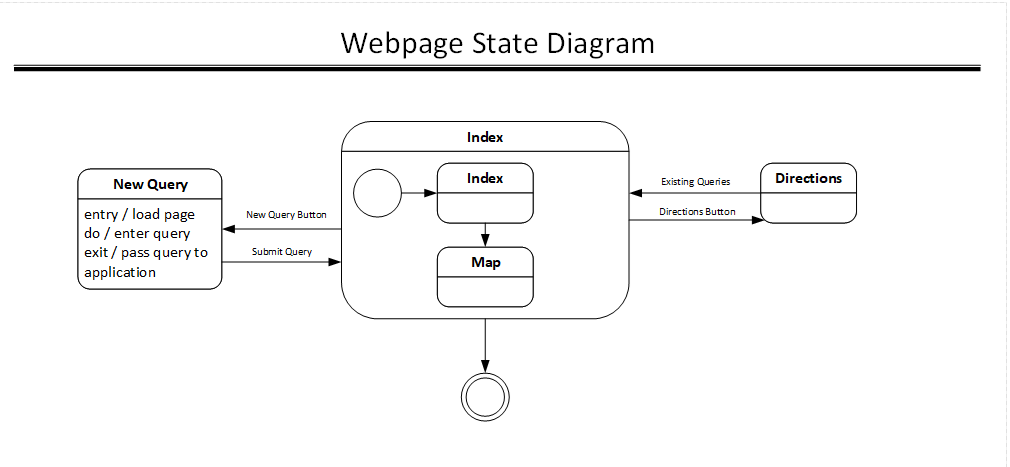


This is an activity UML diagram that shows the different flow of events for the webpage.

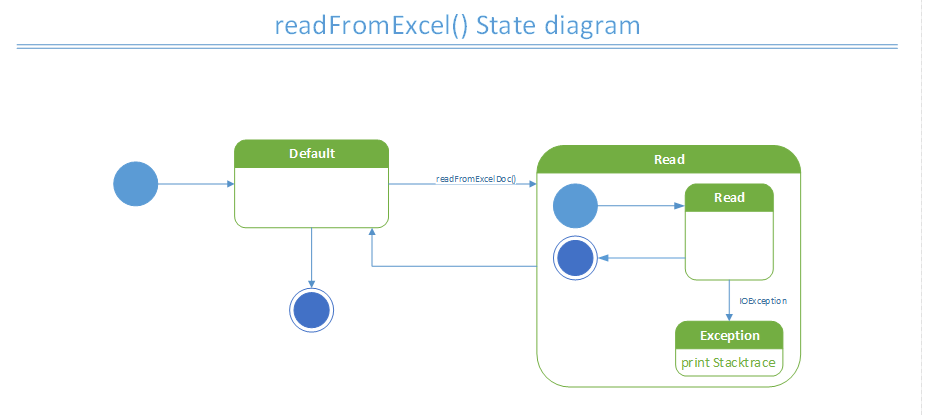
This is a UML sequence diagram that shows the parsing of data methods involved with the DirectionsAPI and the DistanceAPI classes.



This is a sequence UML diagram depicting the sequence of events involved with the operation of the webpage showing the different methods used.



This is a State UML diagram that shows the different states of the webpage and how it’s working.



This is a state UML diagram that is depicting how the readFromExcel() function is working.