Notes:

This document contains all the links and personal notes pertinent to the development of the various modules of this project:

1. REST APIs and HTTP

**Recommended books and Resources on REST according to Quora Answers**

<https://www.quora.com/Which-book-is-the-best-for-learning-about-Restful-Web-APIs-in-Java-Python-C>

**The best introduction to REST APIs**

<https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm>

Architectural constraints imposed by REST:

1. *Client-server*: Separation of concerns between the client and server-side parts of the software (user interface and data storage, respectively).
2. *Stateless:* A request from the client to server must contain all the necessary information necessary to understand the request. In other words, the client cannot use any stored context on the server to process information. To be specific, only the *server is stateless.* State about the session must be kept entirely on the client side. (*client-stateless-server)*
3. *Cache:* Cache constrainsts require that the data within a response to a request be implicitly or explicitly labeled as cacheable or non-cacheable. If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests. (*client-cache-stateless server)*.
4. *Uniform Interface:* REST emphasizes the implementation of a uniform interface between components.
5. *Layered System:* Each component cannot see the layer above or below it (abstraction).
6. *Code-*On-Demand: REST allows client functionality to be extended by downloading and executing code in the form of applets or scripts.