Week 14

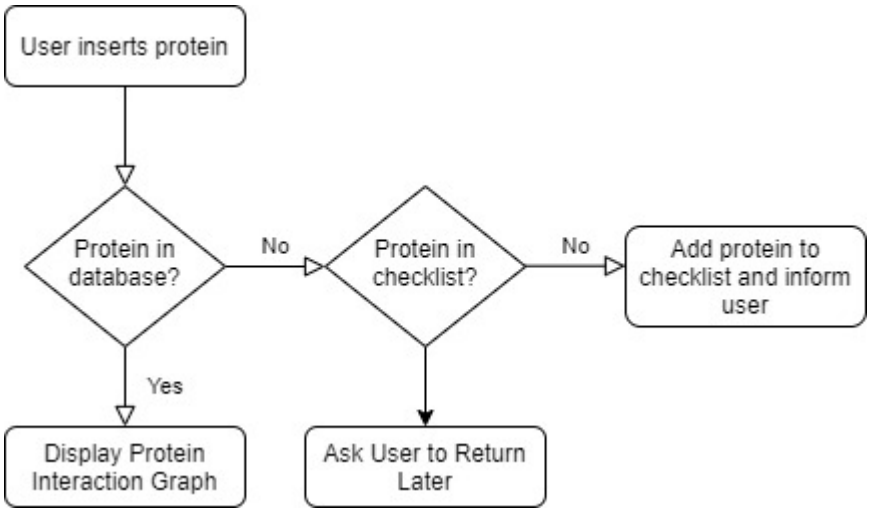
Work done

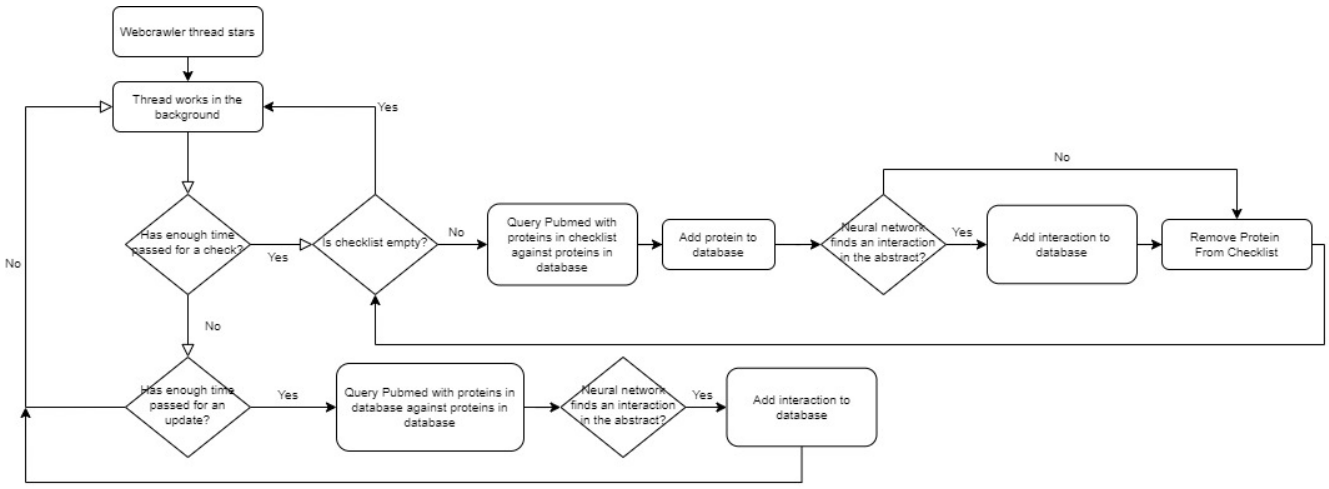
The week has been spent creating the final form of the Django webapp, attaching the trained neural networks.

In a final analysis, the webapp uses keras and spacy for a backend, while the models are loaded and used with joblib. One key understanding that was required in using Keras models with the Django framework is that a particular model must be loaded in the request used, otherwise it will not be available. Thankfully, as our webcrawler runs in a separate thread, our heaviest network, the MLP used for finding PPI in abstracts, is only loaded once, lessening the computational time.

A new addition to the overall architecture is an update function inside of the webcrawler, that once a day verifies all the proteins against themselves for new interactions.

Overall, the webapp relies heavily on the query box previously described. Below is a more intuitive and easy to follow description of the webapp in the form of two flowcharts.





Problems

The webapp's robustness is certainly not 100%, but for demonstration purposes it works well enough. Of course, it's main weaknesses are the neural networks, in particular the PPI network, to which a manual bias of 0.49 has been added for demonstration and testing purposes.

To do for next week:

The next week will likely be spent writing the scientific article that sums up the research.