Reflective Essay

Annabel Page
Birkbeck, University of London
MSc Bioinformatics and Systems Biology
BioComputing II

Approach to the project

As a team we assigned each layer of the project depending on what the individual felt most confident working in or whether they wanted to develop skills in a previously unexplored area. As one individual felt more confident working on the database and the other wanted to practice html for the front end, I was naturally left with working on the middle Business Layer which I was happy to do.

Requirements for my contribution

Working as the middle tier, I understood that I was required to act as a mid-point of communication between the front end and the database layer. My initial understanding was that the middle layer would be required to take user input from the front end, send this query back to the database, return this back to the middle for data manipulation ready to be sent back to the front end.

While this seemed straight forward on paper, we ended working through multiple iterations of how the project should run before getting to our current point.

The development process

Initially we worked on parsing the GenBank file into the database and then sending the whole database to the business layer as a Python Pandas dataframe, the business layer would then return a row or value from the dataframe based off of a search input from the front end. We soon realised however that this would negate the point of a relational database as if the database itself ever amended or updated, the business layer would be sending out of date information to the front end as the relationship has been lost. We went through a few more ideas of querying the database once and returning a whole list of dictionaries, but realised we would still be running into the same issue.

We finally settled on our current system that would take a user input query from the front end, use this input to query the database via the business layer and return back one dictionary that would have all the raw information for that particular row in the database. The front end can also call functions from the business layer that will return manipulated data for display. We initially thought of having one function that will return a dictionary of all different data manipulations but later settled on one function per data manipulation.

After establishing that I would be receiving one dictionary from the database layer, I was able to start writing the business layer functions and ensuring that they worked correctly. I tested this using dummy data that I put together in a dictionary so I knew that the code would work once the database API for retrieving the data was working. I was able to work together with the database layer to develop the original API that would retrieve the correct row of data in regards to the search input. I later went on to create a further API that would allow me to collect the data from a whole column of the database that could be used for the front end to display. These columns were the accession number, gene_id , protein_id and location. Putting these values into a list would allow the front end to produce a drop down of values that can be stored as a search input.

Unfortunately, due to current circumstances around the outbreak of COVID-19, contact was lost slightly within the group as everyone worked on their individual elements of the project. We heard from our frond end developer that they had been unable to work on the project due to illness; however we had another group member who had decided to work on a second version of the front end.

Code testing

For the business layer, I decided to create one module that would have all appropriate functions for the front end to call. Within this module, I imported the database module API's that allowed me to communicate with the database within my functions. I spend a lot of time testing the code to see how it worked under different conditions, I made sure the correct results would be produced if an appropriate search function was entered and I also tested and employed certain error messages depending on the type of error made. That way, it would be easier for the front end to know what the issue if an error message is returned when calling a function. I made each function so that it can have the same parameter input (except when requesting a column), this can be one variable which is just the stored search input.

Known issues

I have tried to test all the code to limit the amount of errors or issues that may occur when querying the database however I am sure there are certain characters or pieces of information that will return from the GenBank file that I have not accounted for. When I created a function to take the DNA sequence for the whole chromosome and return a list of codon, I did find that it returned a lot of one off peculiar codons that probably shouldn't be there. I mentioned this to the database level but we were unsure if this could just be from unexpected data in the GenBank file rather than an issue with parsing the file as the DNA sequence has worked fine for all other testing.

What worked and what didn't - problems and solutions

I think under the current circumstances, it is quite difficult to navigate a group project without being able to meet in person. It was quite difficult to fully explain ideas to each other over a call, however we were able to send videos to each other explaining thought processes and how code should run. I think in hindsight it would have been good to set up a weekly call/Skype to see how everyone was getting on with their portion of the project and make sure everything was working together cohesively. With regards to the functions and my code, I worked hard to make sure all the code was as concise as it could be and avoid any unnecessary processes to achieve the end result, however there were some areas that I found quite tricky to get as succinct, as such, there probably are better ways of carrying out the process if I were to look into it again.

Personal insights

This experience has helped me understand how to work with and import code from different files and work in more than language for one project. I enjoyed understanding how the process worked and learning from making a lot of errors along the way, but I think it is a valuable skill to understand how different layers of code interact with each other. I spent a lot of time learning to debug my code and employing different processes to see which parts of my code worked and which parts failed so I could pinpoint an issue much quicker. I think this is important as I used to spend a lot of time rewriting code that didn’t work without fully understanding where the problem is. It has also helped me realise how important it is to keep in touch and work together as a group to ensure we are all on the right track to achieving the same goal.