1. Approach to project

1.1 Interaction with the team

The group was in regular contact throughout the assignment. In the initial phases where the programme design and development approach were being decided we had several face to face meetings in college. Once the APIs and data structures had been established we began working more individually on our own sections and kept in contact through whatsapp and emails.

The group had varying degrees of experience with coding so we allocated roles based on strengths. Florence was very effective as project manager and took a lead role in arranging meetings, writing up minutes and documenting decisions and setting timelines for tasks within the project.

1.2 Overall project requirements

Create a genome browser containing Genbank data for chromosome 6. The genome browser would interface with the user through HTML tables and allow searching to be specified based on accession, gene identification, product and chromosome 6 location.

The browser was constructed in three main layers

Front end – forms and tables to take user specified searches to pass down to the business layer and present data returned from the searches.

Business Layer - code to take user data entered into the front end and query the database by calling the python wrapper. Carry out manipulation on the data to pass to the front end.

Database – MySQL database containing relevant Genbank information for chromosome 6 and python wrapper. Parser to read genbank data and extract relevant information to populate the database (see 1.3 below for further details).

1.3 Requirements for my contribution

My section was to build the MySQL database containing relevant genbank data and the python wrapper to allow the business layer to query it. This was broken down into three main tasks:

Produce the code to parse Genbank data and extract relevant fields.

Create the MySQL database.

Write a python wrapper that the business layer could call to query the database. The python wrapper had to be able to search single entries based on accession number and return full information contained in the database for that entry including DNA sequence and protein sequence. In addition, a further search function would use multiple fields as the search criteria and return all matching entries. Data returned for this function would not include DNA sequence or protein sequence.

2. Performance of the development cycle

The group held a meeting in the first weeks where we reviewed the requirements and data in genbank. We set the approach on how the browser would work and highlighted the key

information from genbank that would be needed. Following that we defined the APIs and the data structure. We worked individually on the sections and set a deadline for bringing them together that would allow time for testing and any adjustments. Once the completed sections where committed to github we tested whether the BL could call the DB API. Once that was established the front end added and we did a full test of the browser.

3. The development process

I started the development process by mapping out how the code would work, breaking the code down into smaller sections. I then wrote the code for each section and tested each section individually. Once one section was complete and working I added it to the main piece of code. I then ran the full code once it was working. This approach worked well for me as at the start I was getting a lot of errors and it was helpful to have a small piece of code to check to identify what the problem was. Towards the end I was getting fewer errors and writing the code became much quicker.

4. Code testing

4.1 Testing the Parser

To test the parser I created a smaller dataset with a reduced number of entries and ran it on this to check the output. This was helpful in checking the data was coming out in the text file correctly. One example of an error that was picked up in this way was that some CDS regions occur over multiple lines in the genbank file and that the parser was only picking up the first line. The regrex to extract the CDS data was adjusted to [^\] to pick up all characters including newline except \ which marked the start of the next field in the genbank entry. Commas were also identified in some fields of the output file and the parser was adjusted to remove these as these are used a delimiters in the data output and upload to the database.

4.2 Testing the Database

Once the data was loaded into the database some dummy queries in SQL were run to check the table was working correctly. The output from each field was reviewed to ensure it contained the expected data. The number of entries in the database was checked to the output from the parser.

4.3 Testing the Python Wrapper

A dummy programme was made to call the dbapi to check the output and that the parameters worked. Initially this highlighted that that all the keywords for the getAllEntries() function had to be entered even if they had no user data despite being conditional arguments. The call structure was initially changed to keyword='' for arguments with no user data. When the browser was tested it became apparent that fields with no data entered in the front end returned None. The call structure was then changed to keyword= 'None'

5. Known issues and alternative strategies

Some entries in the genbank data reference coding regions in other accession entries. The assignment specification states that these entries can be ignored. We discussed two approaches to dealing with these in the group 1) Ignore the accession id in the cds region and treat these entries as normal or 2) filter out CDS entries with accession numbers in them. We decided the 1st approach was simplest.

6. What worked and what didn't; problems and solutions

Different skill and experience levels in the group

The members of the group had different levels of skill and experience coding. We set the group up based on people's strengths and gave Florence the role of the business layer and project manager. This helped the group work together as Florence's section linked in with both the front end and database. This helped other members of the group, such as myself, who were slightly less experienced with coding to focus more closely on their individual sections.

Florence was very effective as project manager keeping the group organised arranging meetings, writing up minutes and setting deadlines for tasks within the project such as drafting the APIs and uploading code to github to allow testing. As a result, we were able to have a completed set of code on github for testing by mid-April.

Group organisation

The way the group managed the task and communicated worked well. We had face to face meetings to agree the APIs and data structure and team members tasks early on. With this clearly defined at the start the group was able to work on their individual sections whilst communicating on whatsapp and email. This meant social distancing restrictions had minimal impact on the group completing the work.

7. Personal insights

Core coding knowledge and programme development process

This assignment has greatly increased by competence with python in particular how to manage a larger programme by breaking it down into smaller modules.

As a result of the assignment I've learnt how to use github to project manage development, document and test code.

Documentation

A key learning from the task has been the importance of documentation and presentation of code particularly in a larger project. Including proper docstrings and headers to code allows other users to quickly understand a section of code which is important to ensure the programme works together as a whole and helps when referring back to your own code. Making code as concise as possible makes it much easier to read and makes it quicker to update and alter in the future.

Importance of planning in the development cycle

The development cycle ran well within our group because we defined the overall approach and data structure up front. This meant that everybody understood the overall design of the browser before writing their individual sections and it was easy to pull together and start testing once the code had been committed to github. Mid way through the project we made a change to allow the user to search on multiple fields however this was easy to do as we had an overall plan of how the browser would work and it was easy to identify what sections would be affected by the change.

Importance of reviewing underlying data before starting

Reviewing the data thoroughly to check the format and any unusual characters or entries before writing the code helps you design a plan that will work. I found it was quicker to spend some time checking for unusual entries before starting than going back and adjusting code once you have found an issue through testing.