

1. Approach to the project

The Group 6 worked well throughout the duration of the project as a team from the initial assignment of the project till the end of the project. The group met as twice each week on a Tuesday and Thursday evening to develop the code for each layer of application. The group also spent several days at the Birkbeck library, so it was a combination of remote working and face to face interaction between the team members to share ideas, and also to help one another out. This way of working was really productive for the project as the working between the different layers was very smooth.

The team worked in a really collaborative manner and adopted the pairing methodology. The group worked together on each application layer by problem solving to develop the code where the lead developer role for each layer was alternated by Denzel for Data Base, Amber for Business Layer and Farah/ Hirushi for Front End. This was to facilitate the different levels of programming knowledge within the team members. This also created a pleasurable environment for the group to learn in.

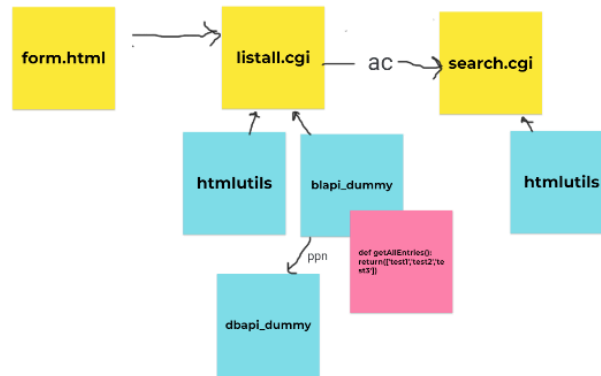
We started by cloning the repository from the GitHub and making individual branches for all the team members for the project. We then made sure that all the different layers were pulling the dummy data and that the different layers were running smoothly. A few screenshots have been included as an example of how the group set the targets using a sticky note software for the development of the genome web browser.

1. Interaction with the team

The interaction between the team members was very smooth and all the team members played their part in problem solving. Any difficulties arising within the different layers of the project were dealt with by all the members of the group by sharing of ideas and brain storming so that a solution to any problems arising was reach amicably.

The team members had regular team meetings over 'Google Meet'. The team met on video conferencing twice every week on Tuesday and Thursday evenings. These were the times when we would normally have our regular lectures, so we made sure we had a regular routine to work on the project and did not leave anything for the last minute. In these sessions the lead developer for a particular layer would present on the 'Google Meet' and work their way round building the layer and all the other team members would contribute with their feedback in any area if we got stuck. In these meetings we would have a set objective of what we wanted to achieve from that that session, and all the members of the team worked together in achieving that objective. The team also shared regular updates over the 'WhatsApp' chat group. The whole project was planned in a way so that we were on track to finish the project on time.

We had some sessions over the weekend if we got stuck at any point or if any of the team members could not make the session due to some other commitments, so that we could all work together and no-one missed out on the learning process. We also spent several days in the Birkbeck Library on campus where we mad use of the conference pods and the presentation facilities in the library to work independently and also collaboratively on different aspects of the different layers and share different ideas. This was a great way to interact between the group members and made the whole process very enjoyable.



Sample code

This is the business logic API (confirm whether to include more updated version?)

```
"""
```

```
# Add the bl sub-directory to the module path (for testing this routine)
# and the directory above to import the config file
import sys
sys.path.insert(0, "../db/")
sys.path.insert(0, "../")
```

```
import dbapi_dummy # Import the database api
#import config # Import configuration information (if needed)
import re
```

```
def getAllEntries():
```

```
"""
```

```
...Function comment header goes here...
```

```
This is a very simple function that just calls the database API to do the SQL to
obtain the full list of entries. It doesn't need to do anything else.
```

```
"""
```

```
all_entries=dbapi_dummy.getAllEntries()
return(all_entries)
```

2. Overall project requirements

We worked as a team to ensure that all the different layers of the project ran smoothly and all the requirements of the project were met. Our group were allocated the chromosome 1 to work on. We started by downloading the human genomic DNA records which contains the coding sequences from Genbank for that chromosome onto the Pandora account.

We started with building the Database first to make sure that all the different search parameters of the website were extracted and that a database was developed on the Birkbeck server and that the website would be able to extract the required data and to

Web Page Requirements

Identifier (CDS / gene = XXXX)
Protein product name (CDS / product = XXXX)
Genbank accession (ACCESSION)
Chromosomal location (eg. 8q24.3 source / map=XXX)

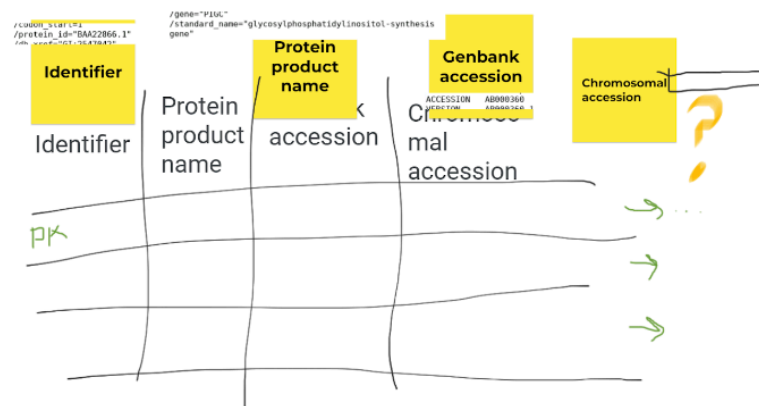
>> details of the gene

- DNA sequence with coding regions highlighted
- Amino acid sequence with coding DNA
- Codon usage frequency within the coding region (plus coding usage frequency across whole database)
- Identify sticky end restriction enzymes in both coding and non coding regions (EcoRI, BAMHI and BsuMI + other enzymes)

```
wget  
http://www.bioinf.org.uk/teaching/bbk/biocomp2/project/data/chrom_CDS_1  
.gz
```

Jamboard -

-<https://jamboard.google.com/d/1VDgQ1Vpj1EN-TQuj5VMNGCziZkp85-ak3SK26FU4zLI/viewer?f=1>



search on the different parameters. This would be the database API which would need to access the MySQL database. These were to be done as described in the project requirement description.

The Business layer function would be able to call the database API which would connect directly to the MariaDB which would contain all the raw data extracted from genbank. The function for this layer would return the data in the form of a list of dictionaries, each dictionary returning each attribute required for each genbank entry. This would be done by creating entity specific attribute extraction functions.

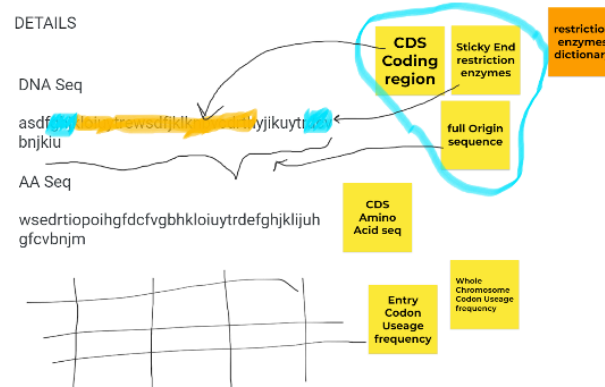
The front end would in turn be able to call the functions from the Business Layer to be able to search the database to find an entry based on gene identifier, protein product names, Genbank accession, or chromosomal location.

3. Requirements for my contribution

I and Hirushi were leading the development of the Front End of the web page and we were working in collaboration to develop an alternative Front End as we had four members in our group. This involved developing a front end consisting a set of web pages A (set of) web pages that allow the user to query the database by accessing the

data via the middle tier. This would allow the user to be able to view and query the database to find an entry based on gene identifier, protein product names, Genbank accession, or chromosomal location also providing the user with the ability to identify restriction enzyme cut sites within the DNA. This would provide an online database containing information for various genes found in Human Chromosome 1. This database would allow the user to search for specific genes based on multiple parameters including: Accession Number, Chromosomal Location and Protein products among others. The results page would be able to return a table of information regarding the gene that had been searched for. For example, if one searches 'RHD' in 'Gene ID' the results would show the user information only for genes with that Gene ID, as well as their accession numbers, protein products etc.

A table would also be returned for each specific entry, highlighting respective codon usage in said entry, as well as across the entire database. Upon clicking a specific entry, a more detailed breakdown for the search could be seen, with information such as restriction enzyme sites. This would have to be achieved by providing supporting Python/ CGI scripts that access the business logic layer of code when forms are submitted and generate new pages.



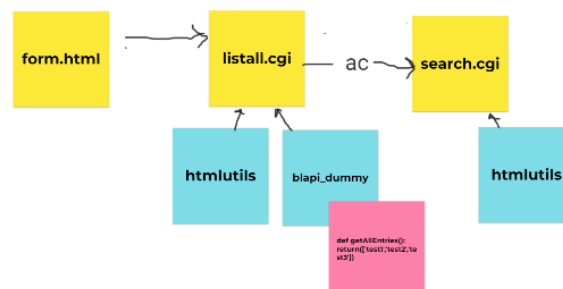
First time getting the CGI and API's to work

Page 1: <https://student.cryst.bbk.ac.uk/cgi-bin/cgiwrap/ha002/listall.cgi>

Which takes you to

Detail page 2: <https://student.cryst.bbk.ac.uk/cgi-bin/cgiwrap/ha002/search.cgi?ac=test1>

By pulling through the accession number



For the front end to work properly, we made sure that all the required information for the different search parameters were pulled through from the Business Layer End. This was achieved by collaborating regularly with the Business End lead. We also made sure that all the search results were presented in a well presented table with colour coding all the different search results so the results would be easy to interpret. Also, the final page of the search contains a key to make it easier for the user to interpret the different results like:

- Count of Codons in genbank entry (1st Number)
- % of codon found in entry (1st percentage)
- % of codon found across all chromosome coding regions in database (2nd percentage)

The required tables were created in the CGI scripts and CSS was used to make changes to the different classes of parameters like background, font size, colour changes, etc.

2. Performance of the development cycle

A lot of time and thought process went into brain storming and discussing various options at tackling a given problem in an efficient manner. As a result I went through a lot of iterations of different approaches at tackling the problems encountered in the duration of this project. This helped in arriving to the most appropriate solution to a given problem. I found that making a plan from the start on how to tackle a given problem works best and then one can work through that process by trying to solve a given problem in different ways unless we arrive to the desired conclusion. That is how I tried to tackle and solve the different problems faced during the process.

3. The development process

This project was a great learning process for me. If something was not working for me, I tried to go back to the basics to find a solution and to pinpoint any mistakes I was likely making. I went through the lecture recordings from this module and the other modules related to this project i.e. Biocomputing 1 and Database. I also gave my feedback in finding the solutions to solving the different problems to help the team out. It has broadened my horizons and I feel that it has helped to be a more confident programmer.

4. Code testing

The codes for the different layers were tested to make sure that they worked properly and efficiently, for example while developing the database the code for search limit was initially set to select 20 entries from the chromosome 1 to ensure that all the layers worked smoothly without putting too much strain on the system, with the plan to remove the limit at the end of the development procedure. We also tested the different codes by running them on the Jupyter Notebook and applying different test conditions and printing each line of any broken codes to pinpoint and solve any problems in the code.

5. Known issues

We encountered some issues while extracting the CDS DNA String entries for the database layer. Some of the entries had overlapping CDS regions which might cause issues when presented to the front end.

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

I had to delete my git repository branch and create a new repository one day before the deadline as I had been working on another group member's repository branch by mistake as we were working together on the code development on a different branch. When I tried to commit and push the code over to the git, it threw an error. All the code being worked on the whole day was consequently lost. So it had to be written again, but it was a bit quicker to write the code the second time round. But I gained an important lesson from this experience to always save, commit and push the different versions of the code you are working on onto the GitHub to get all the backup of different versions of the code that had been worked on. Same happened with another group member's repository as the same mistake was made.

7. Alternative strategies

We would have liked to use more interactive webpage by using the javascript for the webpage development, but we decided as a team to keep it simple and stick to the basics in case anything breaks in the webpage. So we decided as a team to make sure everything was in working order. One of the factors for this was time taken for cleaning the data for the database. I think given more time and more programming skills, we could have added more features to the web page to make it more interactive. But given the circumstances and the other problems we had to encounter during the database development, I am pleased with what we have achieved as a team.

8. Personal insights

This project has been a really good exercise for me in my development to be a programmer. I have learnt quite a lot from the other members in the group. I think working as a team has helped my development as we went about how to tackle the different problems. I learnt a lot by being a part of the project team and from the feedback of the more experienced members of the project team. I think we gelled in really well as a team, trusted and helped each other out. This has specially helped as we have been doing online lectures and the face to face interaction with my fellow class fellows has helped me share ideas and develop my thought process to tackle different problems and to become a better programmer.

Author: Farah Khan

Date: 02/05/2022