

## DATABASE IMPLEMENTATION

Janani Govindarasan

Approach to the project

### **Interaction with the team**

Working with Coco and Ang was a very rewarding experience. We met at least once a week to discuss progress, exchange ideas, and resolve persistent issues. Early on in the project, we set up a GitHub repository for the project.

### **Overall project requirements**

Our group read through the project requirements (<http://www.bioinf.org.uk/teaching/bbk/biocomp2/project/page01.html> and <http://www.bioinf.org.uk/teaching/bbk/biocomp2/project/page02.html>) together to ensure we understood the requirements.

The main task was to build a genome browser for chromosome 7, the data was available through the genbank text file.

We decided that we needed the following functionality from the website:

Front end: This should include forms and tables that allow the user to specify searches and pass it to the business layer. Details for each gene entry should be accessible through a search, this includes dna sequence, accession number, codon usage etc.

Business Layer: This should connect the front end and database layer. Data entered at the front end should query the database through a python wrapper.

Database: Relevant genbank information for chromosome 7 will be used to populate the MySQL database.

My contribution to the group project:

My role was to create the database and this task involved three parts. The first one was to create a parser; this should return the data we are interested in from the genbank text file. The extracted information should be in a format that can be loaded into a database. The second part involved designing a database to load the information. The final part involves creating a data access tier that can be imported to the middle layer.

## **Performance of the Development Cycle**

I discussed with Ang and Coco the data format that was essential for the middle layer so that the database tables and access functions could be designed according to the requirements.

It was decided early on that the summary tables required on the website clearly translated to an identical table in the database, that could be accessed with a sql query.

As mentioned above, my first task was to create a parser to retrieve the information needed for the browser. I initially went through the file to make sure I knew what the format was like and what sort of information was available.

I decided to use regex expressions to retrieve the list of data I needed. 'stackoverflow' and <http://regexr.com/> were very useful for creating patterns and match records.

Next step involved creating the tables and populating the tables. For this part I relied heavily on tutorials such as [Python MySQL Create Table \(w3schools.com\)](http://www.w3schools.com/python/mysql_create_table.asp) and [How to Create and Manipulate SQL Databases with Python \(freecodecamp.org\)](https://www.freecodecamp.org/news/how-to-create-and-manipulate-sql-databases-with-python/).

## **Coding testing**

### **Testing parser**

I created a separate chromosome 7 genebank text file containing the first 100 lines. I used this file to ensure that the parser retrieved all the necessary information.

### **Testing database**

After the data was loaded, dummy queries were used to ensure the table was running properly.

### **Testing python wrapper**

## **Alternative strategies**

I believe using the biopython package would have made my task a bit easier.

## **Personal insights**

Coming from a biological background, I found this project very challenging. I thought communicating with group members was particularly challenging since I am not accustomed to using coding related terms. At the beginning of the project I had to do quite a lot of research regarding what my part should include and what it should look like. Lecture notes and 'stackoverflow' were very helpful for this assignment. I.e. if my code didn't work, I could look up solutions on 'stackoverflow' and get an idea on how I could resolve this problem. Lecture notes helped to figure out what my code should look like overall, i.e. writing clean codes. I often found myself spending a lot of time writing long codes without doing any tests. Unfortunately, when I got around to testing, they often didn't work, and I had to delete and rewrite them entirely.