

## **Executive Summary**

Information has been increasing since the birth of technology, hence it is getting difficult to analyze and understand all the information we have. Information is increasing rapidly whereas the methods to visualize the data have not changed as often. Thus researchers are faced with a problem, as they cannot analyze such an amount of data, if it is not represented correctly.

Visualizing data in a manner that is scalable has always been a challenge in the field of computer science. The Internet and biology are two main sources of rapid increase in information, in this project I am trying to understand the current methods of visualizations and changing them to fit an environment in which data is expanded very rapidly which makes it very interesting as well as challenging.

In this project I am trying to solve this problem using data visualization as the best way to show any information is through visual means; that is my approach to this problem. Humans are naturally good at detecting patterns and giving a graphical representation of the data would allow them to make observations which were not possible using previous approaches.

To solve the problem I have used a technique to dynamically generate graphs and to show the data in an elegant manner, as the data to be shown on the graph increases the previous values are moved in order to give priority to the new data. This technique is being used to display trees with a huge number of nodes. This technique is not affected by the increase in the amount of data and hence a sound approach to the problem.

This technique mainly focuses on trees but it can be applied to display vast numbers of data types. This method uses similar but richer and more dynamic visualization, than visualization methods currently used, and allows users to view data in a more appropriate manner without having to understand the new system.

## **Acknowledgements**

This project would not have been possible without support of Julian Gough, his advice and guidance was very important to the project.

I would also like to thank Matt Oates, his feedback; guidance and encouragement helped me a lot during the project.

Thanks to my friends for their support and advice during this project.

I would like to thank my family and my girl friend for always supporting me and believing in me.

## Table of Contents

<b>Introduction .....</b>	<b>3</b>
<b>Goal .....</b>	<b>4</b>
Goal of the project .....	4
Features and benefits.....	4
<b>Background and Motivation .....</b>	<b>4</b>
Setting and History.....	4
<b>Bioinformatics.....</b>	<b>4</b>
<b>HTML .....</b>	<b>5</b>
<b>AJAX .....</b>	<b>6</b>
<b>Web Service.....</b>	<b>6</b>
XML.....	7
WSDL.....	7
UDDI .....	7
<b>Types of Web Service .....</b>	<b>8</b>
SOAP.....	8
The request:.....	9
The response:.....	9
REST .....	10
The request:.....	10
The response:.....	10
<b>Benefits .....</b>	<b>10</b>
<b>Data Visualization .....</b>	<b>11</b>
JavaScript InfoVis Toolkit.....	13
Protovis.....	14
<b>SUPERFAMILY Database .....</b>	<b>14</b>
<b>Web Services in Bio Informatics .....</b>	<b>15</b>
<b>Web Services &amp; SUPERFAMILY Database .....</b>	<b>15</b>
<b>Current Projects .....</b>	<b>16</b>
myGrid project.....	16
Problem Statement .....	18
Initial Business Case .....	19
How is this different? .....	20
<b>Proposed Solution.....</b>	<b>20</b>
Data acquisition .....	21
Data mapping.....	21
Rendering.....	21
<b>Technology .....</b>	<b>23</b>
Major Tools.....	23
Target Platform:.....	24
<b>Basic design.....</b>	<b>24</b>
<b>Non-functional Requirements .....</b>	<b>25</b>
<b>Initial Use Case .....</b>	<b>25</b>

<b>Detailed Use Case .....</b>	<b>26</b>
<b>Class Diagram .....</b>	<b>27</b>
<b>Sequence Diagrams .....</b>	<b>29</b>
Statistics of Genome.....	29
Children by NodeID .....	30
<b>System Architecture Diagram .....</b>	<b>31</b>
<b>Deployment Diagram .....</b>	<b>32</b>
<b>Database Schema: .....</b>	<b>33</b>
Entity-Relationship Model: .....	33
<b>Guidelines, Standards &amp; Templates.....</b>	<b>34</b>
Coding Standards.....	34
Data Naming Standards .....	34
Development Tool Standards.....	34
Documentation Standards.....	34
<b>Testing .....</b>	<b>35</b>
Unit Testing .....	35
<b>Results.....</b>	<b>41</b>
Data Visualization .....	41
Web Services .....	47
Request:.....	47
Response .....	47
<b>Critical Evaluation.....</b>	<b>50</b>
<b>Conclusion .....</b>	<b>52</b>
<b>Future Work.....</b>	<b>53</b>
<b>Glossary .....</b>	<b>54</b>
<b>References .....</b>	<b>55</b>
<b>Appendix .....</b>	<b>59</b>
Functionality available in Web Service.....	59
User Manual.....	61

## Introduction

Information has been increasing since the birth of technology. The amount of information on the Internet is enormous. When scientists started to discover secrets about life and its building block that is DNA. The field of biology started generating a lot of data. Now we have hierarchies of genomes and lot of information about their sequences. As such, a large source of information is the online results of biological experimentation. When this much amount of information started to generate, biology joined hands with Information Technology to give birth to bioinformatics. To process and understand this information a lot of computing techniques such statistical analysis, data mining and result visualization were used in bioinformatics.

The rise in the amount of information is useful but has its affects as well. Now that so much information is available to us. We can't visualize that information correctly and neither can we share that information effectively, as everyone uses his or her own way to show the information. This project discusses the problems that we have because of the amount of information that we have today.

In this report first we will have an overview about web services, how they work and why do we need web services from there we move on to visualization and why it is important to have a clear and effective visualization. We will than briefly discuss the field of bioinformatics and how Information Technology contributes to the field of biology.

Moreover, we will look at some of the problems that exist because of the amount and characteristics of the data in the field of biology. Further more we will see how a technique to visualize the data using AJAX can be applied to rectify this problem. We will also discuss how web services can be used to solve some of problems researchers have today.

The dataset that I will be using to present and solve this problem is from the superfamily database that holds information about genomes and proteins. I will talk about why this dataset is a good choice for this project further into the report.

This would give us a good idea about some of the problems faced and a possible solution. Further into the report we will see how this system has been designed, implemented and most importantly the results of the system. Finally we will evaluate the system based on the objectives and opinions of people about the system.

## **Goal**

### **Goal of the project**

The goal of the project is to improve the data visualization methods implemented on the SUPERFAMILY database and introduce web services so that the data can be retrieved in a standard format.

### **Features and benefits**

The system features all the required functions of the superfamily database to be available through a web service interface, and also incorporates the visualization of the data from the database. The visualization will be interactive, informative and will change and adapt according to users needs.

## **Background and Motivation**

### **Setting and History**

Web services are common and are being used in most of the new systems being developed. Amazon is also one of the companies using it, with products like S3 storage service, EC2 compute cloud, and SimpleDB online database. There are a lot of advantages of using web services as it allows sending data in a standard format, allowing everyone to be able to use it as desired. [1]

If I talk about research in bioinformatics that is the combination of biology and Information Technology. No matter where a person lives, whatever is his social background, he always wants to get the best and correct information available on the Internet and get it easily.[2] Current methods of visualizations are not good enough to represent the data in a manner that can clearly and effectively provide the information to the user.

### **Bioinformatics**

Bioinformatics is the combination of Information Technology with biology. It began with designing database to store information about genomic data, which was producing a lot of information. It also required that an interface be provided which would be complex as to show the existing information from the database and allow addition of more information as well which extended it to web. Now bioinformatics deals with more complex things than just storing data from the genome revolution, as this new amount of information is huge things like data mining, machine learning algorithms, data visualization are also required as visualization needs to incorporate these techniques such as data mining on the biological data. [36]

## HTML

Hypertext Markup Language is a markup language for web pages. As DNA is building block of life similarly we can say HTML is building block for webpages. HTML consists of tags that are enclosed in angle brackets.

A simple HTML page would look like this

```
<html>
<head>
  <title>MSc</title>
</head>
<body>
Web Services and Data Visualization
</body>
</html>
```

**Listing 1: simple HTML page with 'MSc' on the title bar of the browser and 'Web Services and Data Visualization' in the page.**

HTML as seen in Listing 1 has always been used to communicate between server and the user. The user would submit a form and the server would respond to the request of the user. This was working fine but after some time users of the application started complaining about the speed of the response they get, they were not satisfied with the responsiveness of the server. [7]

A simple HTML form that user would submit would be something like this

```
<html>
<head>
  <title>MSc</title>
</head>
<body>
<form name="unameForm" action="server page" method="get">
Username: <input type="text" name="uname" />
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

**Listing 2: Simple HTML form**

The trend shifted towards improving the servers, to improve the responsiveness of the server but still after some time users still thought it was not good enough. Then came AJAX it changed the concept of moving from page to page for information and gave a true application based environment on the web. [7]

## AJAX

AJAX stands for Asynchronous JavaScript and XML. It allows web application to send data to and from the server asynchronously.

Some of the major benefits of AJAX include

- User- interface
- Reduces refresh rate
- Increases client responsiveness [37]

How AJAX does all this? Let's see a basic diagram of how AJAX makes a call and responds.

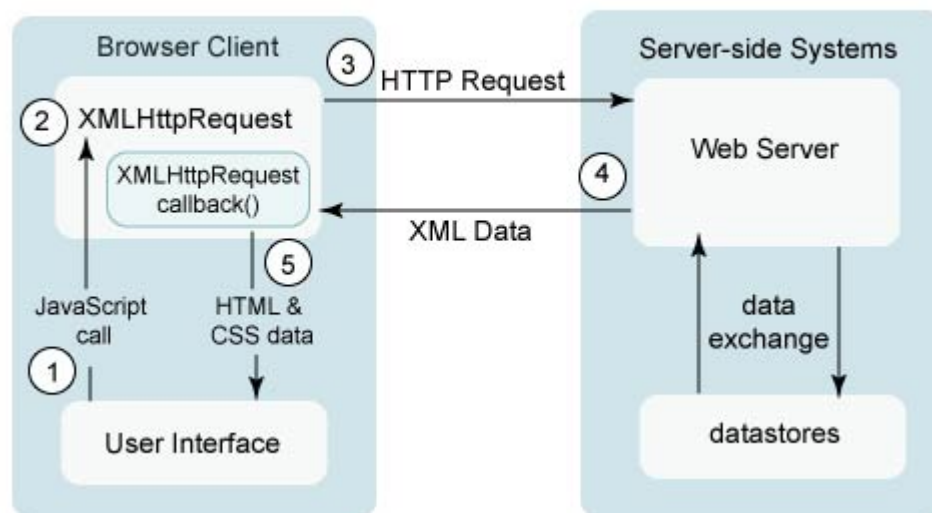


Figure 1: AJAX call

Source: [6]

The diagram illustrates how a user invokes a call from the user interface, which invokes the request object, to communicate with the webserver and return the data in XML, that data is then retrieved by the request object and displayed on the webpage. [6]

## Web Service

Web service allows the communications based on standards using HTTP and XML-based messaging. Web services can be used to perform simple and complex tasks. As they are based on open standards such as HTTP, XML, SOAP and WSDL they are not dependent on things like hardware, programming language, and even the operating system. Enabling application on different hardware made in different programming language and using a different operating system to easily communicate with each other using the web service. [25]



Web services are based on XML and have three important components WSDL, SOAP and UDDI.

The first component is WSDL. When the web service is being created a WSDL documents is created which has all the information about the web service, such as its location and the functionality it will provide. The second component is UDDI is an optional part but can be important for some. Once the WSDL is created its information is than added to UDDI registry this allows the web services to be searched and found by the people willing to use it. Based on the information in the UDDI registry its gets the instructions from the WSDL document. SOAP messages are than formed that are exchanged between the services over HTTP. [25]

### **XML**

XML is a W3C standard which defines a meta-language for describing data. XML is basis of all modern web services as they use XML based technologies to explain their interface and encode their messages. WSDL, UDDI and SOAP and all XML based technologies and can be interpreted by any machine. [25]

### **WSDL**

WSDL is also a standard set by W3C for describing web services. Anyone who wants to access the web service and can read and interpret it. It provides all the details to the user or the program requesting it, that from where the service can be accessed what functionality is provided by the web service and what protocols should be used to communicate with the web service. [25]

A WSDL file as discussed above is based on XML and has six main elements:

1. Port Type – groups and describes the operations performed by service
2. Port – specifies an address for binding
3. Message – describes the names and format of the messages supported
4. Types – defines the data type used for sending data between client and server
5. Binding – defines the communication protocols supported by the operations
6. Service – specifies the URL for accessing the service

This documents is a contract between the client and the server if the rules are followed communication can be done. [25]

### **UDDI**

UDDI is a standard, which is sponsored by OASIS. It is often referred to as yellow pages of web services. It also based on XML it allows business to list

their web services. They can be private or public. The communication is done through SOAP. A developer willing to use the service can query UDDI to find the service. [25]

## Types of Web Service

### SOAP

Microsoft developed SOAP that is “Simple Object Access Protocol” in the year 1998 and since then it has become a standard in web services. It is used in web services combined with WSDL and XML to exchange messages. [9][13][14][21]

It is also a W3C standard and the ability for SOAP to exchange messages between clients irrespective of the programming language is what made it an important in web services. Same as WSDL, SOAP is also an XML based document has these elements

1. Envelope – specifies that the xml documents is SOAP message; encloses the message itself
2. Header - this is optional it has information related to message the time, authentication data etc.
3. Body - has the message pay load
4. Fault – this is optional as well it contains the error within a SOAP message.

Data between users and web services is exchanged based on SOAP as it is mentioned in the WSDL contract web service is guaranteed to work with SOAP. The data is exchanged using request and response SOAP messages.[25]

The basic structure of a SOAP object:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <!-- Header information here -->
  </env:Header>
  <env:Body>
    <!-- Body -->
  </env:Body>
</env:Envelope>
```

**Listing 3: Basic SOAP Structure**

This is the basic structure now let's see how can we use SOAP to request a stock quote.

So a basic example of SOAP for requesting a stock quote from a web service would look something like the following:

**The request:**

```
GET /StockValue HTTP/1.1
Host: auction.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: xxx
<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:s="http://www.auction.org/StockValue">
  <env:Body>
    <s:GetStockValue>
      <s:Item>APPLE</s:Item>
    </s:GetStockValue>
  </env:Body>
</env:Envelope>
```

**Listing 4: SOAP request for Stock Value**

The request clearly shows the host to which the request has been made the content type and the item for which the value is required. To understand further we see the response of this request.

**The response:**

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: xxx

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:s="http://www.auction.org/StockValue">
  <env:Body>
    <s:GetStockValueResponse>
      <s:StockValue>1499</s:StockValue>
    </s:GetStockValueResponse>
  </env:Body>
</env:Envelope>
```

**Listing 5: SOAP response for stock value**

The response shows the value that has been returned for the stock is 1499. This gives a general idea on how SOAP works. To see actual requests and response of this system see Listing 8

## REST

“REST was introduced by Roy Fielding in his thesis in which he wrote the main idea of REST (Representative State Transfer). It is based on architectural style and it can be explained easily by comparing the 4 words.” [9][21]

HTTP	Equivalent
GET	Read
POST	Create, update, delete
PUT	Create, update
DELETE	Delete

Table 1: Rest HTTP equivalent

So to give a similar example of a RESTful service compared to SOAP we use the same scenario as in SOAP

### The request:

```
GET /StockValue/APPLE HTTP/1.1
Host: auction.org
Accept: text/xml
Accept-Charset: utf-8
```

Listing 6: REST request for stock value

### The response:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: xxx
```

```
<?xml version="1.0"?>
<s:Quote xmlns:s="http://auction.org/StockValue">
  <s:Stock>IBM</s:Stock>
  <s:StockValue>1499</s:StockValue>
</s:Quote>
```

Listing 7: REST response for stock value

The RESTful compared to SOAP is much lighter in weight and simpler to use as well when comparing to SOAP, as SOAP sends HTTP requests and the load of XML for the action of the request. [9] SOAP is more general and can have any kind of action where as REST is better for database interaction.

## Benefits

Web services has many advantages using them can lead to technological and even business benefits. Some of these benefits are

1. Application and data integration
2. Code re-use

3. Time Effectiveness
4. Supported on more platforms.

### **Application and data integration**

Web service does not need to know on which system or on which language it is being implemented or used. It can be integrated with applications, which are in different language than the service is implemented in. [25]

### **Code Re-user and Time effectiveness**

One of the most important benefits is re-use: once a web service has been created it can be used by many applications, if the functionality requires that service. These benefits result in time effectiveness of the over all system as it supports all platforms that communicate based on a WSDL mediated contract, these platforms are both innumerable and varied. Hence no need to implement a different service for every platform. [25]

### **Data Visualization**

The Gestalt law of proximity states that

*"Objects or shapes that are close to one another appear to form groups".* Even if the shapes, sizes, and objects are radically different, they will appear as a group if they are close together. [33]

This principle is also known as "grouping", which means that when objects are put together they make more sense than if the object is seen separately. Hence if you consider one piece of information you would not be able to get that much information but if you put together similar pieces of information together and view that, it is more likely to give more information. [33][34]

Data visualization is very important when creating a system. If the system does not present the data in a manner that could be easily interpreted by the user, the chances are user will not use your system again, if an alternative is available. To ensure that this does not happen with the system built. I have used toolkits to give an elegant yet informative display.

The question is why is data visualization so important, that it is a separate field in computer science. Lets take a simple example. Consider the letters in Figure 2. Now lets try finding the number of times the letter 'V' repeats in this figure.

```

MTHIVLWYADCEQGHKILKMTWYN
ARDCAIREQGHVLKMFSTWYARN
GFPSVCEILQGKMFP SNDRCEQDIFP
SGHLMFHKMVPSTWYACEQTWRN

```

Figure 2: Visualization Example 1 Source: [7]

It seems to be a little hard doesn't it? Yes you can count the number of times the letter 'V' repeats but it would take you a little time. Some times you might even have to start again, for something that should be easy and should not consume that much time.

```

MTHIVLWYADCEQGHKILKMTWYN
ARDCAIREQGHLKMFSTWYARN
GFPSVCEILQGKMFP SNDRCEQDIFP
SGHLMFHKMVPSTWYACEQTWRN

```

Figure 3: Visualization Example 2 Source: [7]

Now consider Figure 3. Now try to count the number of times the letter V repeats. Simple isn't it. The reason is better visualization because what you wanted to see was visualized correctly. There is a vast difference in the amount of time required to do the same thing on the same data set, the difference is the result of visualization. If the data set is visualized accurately you can get the information very quickly if not it can be very time consuming.

Gestalt theory states that

*"Things which share visual characteristics such as shape, size, color, texture, or value will be seen as belonging together in the viewer's mind."* [33]

Hence if the information being displayed has similar visual characteristics it is likely that the viewer will not be easily able to distinguish, which is why it was not easy to locate the 'V' in the first image, but it was comparatively easy in the second image. [33][34]

The example we considered has a very basic data set. It was very simple and yet the difference in both the visualization is enormous. This gives a good idea on why visualization is important. Now consider a data set, which has more than 10,000 values. It might not even be possible to get the information clearly and effectively if the visualization is not accurate. If the information can be retrieved it would be very time consuming. This projects aims to use the raw data and visualize the data in a manner that would give the result of the example above. So that researchers can focus on the information they actually want instead of spending valuable time in getting the information.

The system developed tries to achieve the balance between showing a lot of information or too little information and tries to show the accurate amount of data that should be shown. System will not in anyway replace the current visualization as these visualization may be required in some cases when looking at graphs like these can help visualize the information in different way. When lot of information is shown it will give a bird's eye-view to the data and when information is less it would yield a more specific result.

Goal of visualization in bioinformatics are

1. To visualize huge amounts of data
2. To improve and aid in pattern and trend recognition.

Visualization is very important when considering biological data and hence I have considered different visualization libraries so that the data is visualized accurately.

### JavaScript InfoVis Toolkit

JavaScript InfoVis Toolkit gives the ability to draw many different types of graphs and has many visualization options as well. [12]

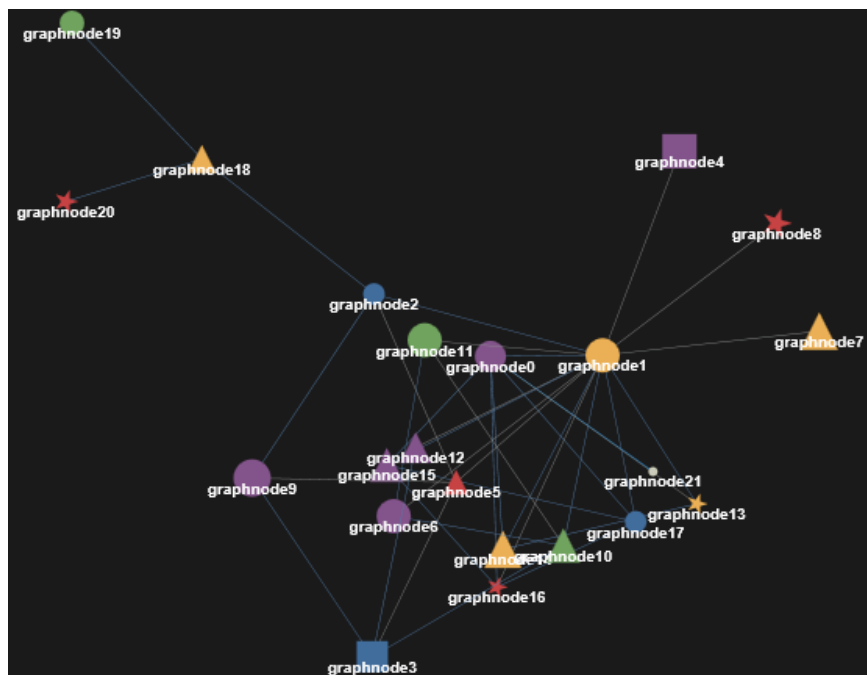


Figure 4: JavaScript InfoVis Toolkit Graph

Source: [8]

Figure 4 shows one graph available in this library, this is just one example, the toolkit has many more functionality and is a complete package for implementing such graphs in a webpage.

## Protovis

This is another library that provides similar visualizations.

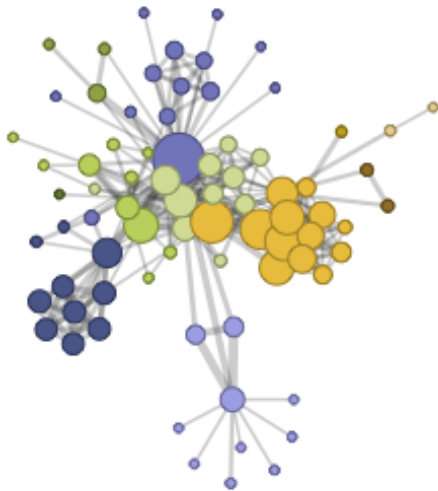


Figure 5: Protovis Visualization Sample Source: [9]

It is a product of Stanford Visualization Group, and it provides similar functionalities [22] but after using both of the libraries I found that the JIT is more suitable and has options that can benefit the user more. For this reason I have used JIT in this project.

## SUPERFAMILY Database

Superfamile database is a resource for comparative genomics/proteomics and is based on hidden Markov models. “The purpose of SUPERFAMILY is to detect and classify in protein sequences evolutionary domains for which there is a known structural representative. Given a protein about which nothing is known other than the amino-acid sequence, the object is to assign known structural domains or more specifically domains at the SCOP superfamily level.”[10][11]

The database currently is accessible using either a cloud image or downloading the data and setting your own server, both of these solutions are not really suitable for a single novice biology user such as an undergraduate student. Even for me setting up the database was little troublesome. Trying to setup a database that is around 30GB in size can be little tricky. The database as a resource for comparative genomics is not as well exposed as the annotation assignment services that the website provides.

The Superfamily database does provide some web services [15] but they are not sufficient enough, I will talk about Distributed Annotation System and the



services provided later. But the services need to be expanded and more functionality needs to be made publicly available through web services to the users.

## **Web Services in Bio Informatics**

In the last decade the ability to perform experiments and research has increased a lot and that is due to the advancements in technology, but this research and experiments means a lot more data. Individual researchers or groups all around the world produce these data. As the technology has advanced that information is available on the Internet, but all the groups maintain the data sets they have themselves and they use a variety of data types and standards.

This information is widely spread and in a form that an individual researcher cannot get all that information, even though information is out there. The information is likely to be disorganized HTTP file indexes or FTP with no meta-data over multiple servers. The superfamily database is built from several hundred such sites by hand! Screen scraping fails on something like a JavaScript driven web app interface. These interfaces are common with Genome browsers such as GBrowse from GMOD[35]. The need to have a system that allows that all the data is available in a manner that every person can use is vital. This gives rise to the need of web services in bio informatics.

Another advantage of having web services is that it can be used with Taverna as Taverna works with WSDL Web services and would be compatible with the web services developed. Taverna is a product of the myGrid team and is a domain-independent Workflow Management System. It provides tools to design and execute scientific workflows and aid in silico experimentation. [38]

## **Web Services & SUPERFAMILY Database**

Superfamily in general wants to scale to the “Cloud” and implementing web services would be a great way to progress. Since we could have many web services servers and load balance them with DNS using services like Amazon Cloud front. This project introduces more services to the users so that they can access much more data from the superfamily database.

The superfamily database currently provides some web services using the DAS service specification. DAS stands for Distributed Annotation System. So users go to DAS registry site and then get forwarded to the distributed annotation servers for which superfamily is a single example. Assignment here is assigning an informative tag to a biological sequence; in the case of SUPERFAMILY, tag is the presence and absence of protein domain super

families. This project is focused on providing web services, which is outside of this scope of domain assignment. So this projects does not replace the current web services rather complements the system by adding new features. [31][32]

## Current Projects

### myGrid project

myGrid project is a project by the government of UK under its E-Science Programme. This project has adopted the technology of web services to provide an open source based web service to facilitate experiments in silico, in the field of biology. [2]

## Current Visualizations

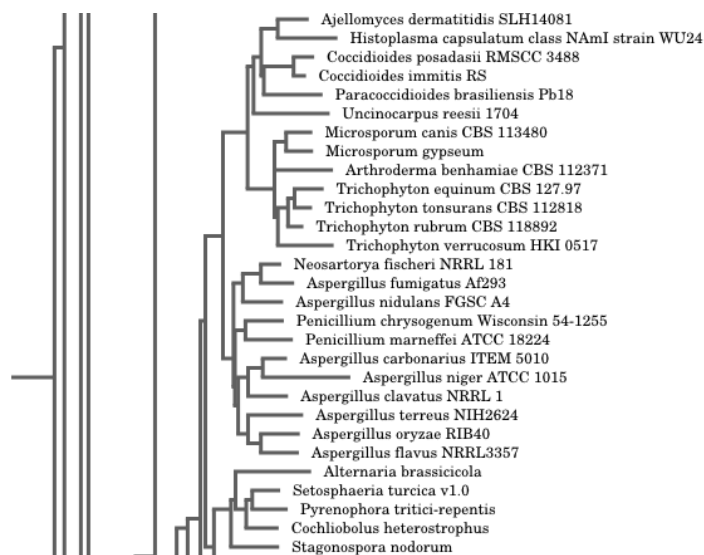


Figure 6: Current Visualization

Source: [1]

The above image is a Binary tree, which is currently implemented. This is a portion of the graph as the information is a lot on a single graph to be easily understood and effectively retrieved by the user.

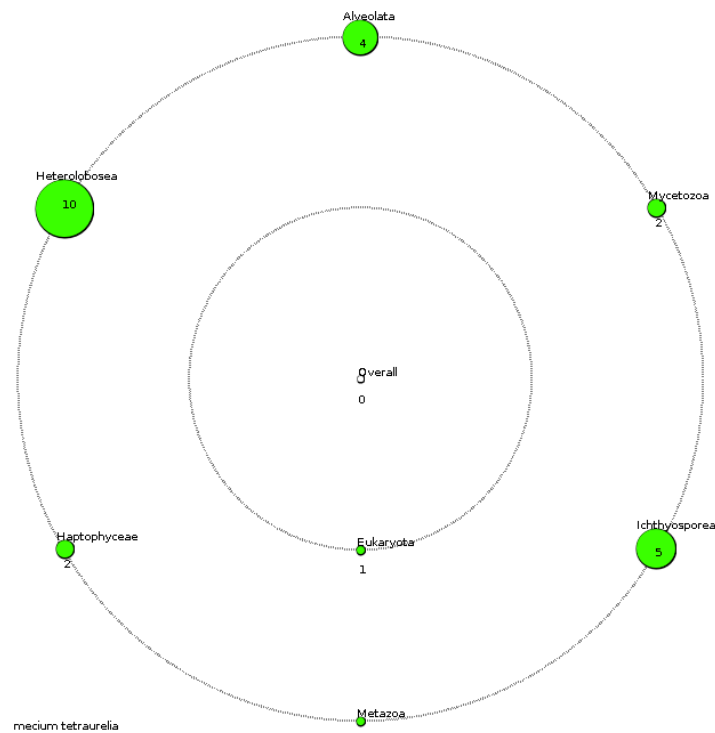


Figure 7: Current Visualization

Source: [1]

This is another implementation, which is currently being used. In this graph the information that is shown can be increased.

Similarly other projects like MOBY-Services and Semantic-MOBY have also implemented the new technology to solve the problem faced by the biologists. [2] They have command line tools already wrapped as SOAP/WSDL services but not a database.

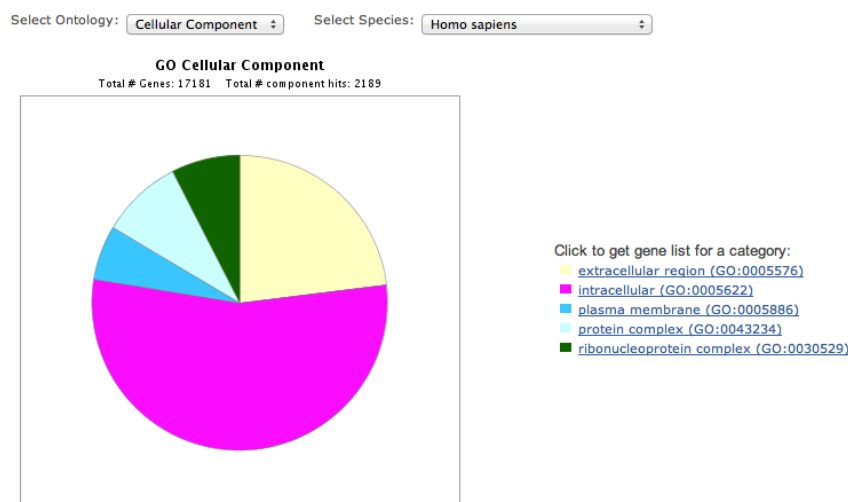


Figure 8: PANTHER Visualization Source: [2]

PANTHER is a very similar service to SUPERFAMILY, so the use case is the same and so are the kinds of visualization to be done. The above page is from

their ontology browser. Clicking the category on the pie chart would take you to another pie chart that shows the details of that portion.

**No hit for lower level categories.**

Figure 9: PANTHER Visualization Source: [3]

If that category has no lower level categories this message is displayed and is a dead end. You have to go back in your browser to change the query. The pie charts are not interactive as well.

So, Websites providing similar information do have some kind of visualization but that visualization is not good enough, it makes it time consuming and hard for a researcher to go thorough the data and understand it. Detecting patterns between genome statistics becomes very hard, as detecting patterns is much easier for humans rather than observing numbers. Some of the current approaches show the data in a table and some of them show them in a graph. Where raw data is used the problem of understand large quantities of data arises. Where graphs have been used the information provided is very limited, the graph does not populate data as the user requests rather a graph containing all the information is shown. So these are some of problems in current approaches.

### Problem Statement

When we talk about genomes, proteins, DNA a huge amount of data and large databases come to mind. The field of biology and bioinformatics requires that the data is available to everyone. Technology has spread to almost everything now days and this field have also adapted technology. Some of the information and databases are available online through World Wide Web (WWW). So is the problem solved? NO, the data can be viewed online but not received in a programmatic form so that it can be manipulated and processed for further research and experiments. Some of the researchers use the technique of screen scraping to get the data. This technique allows you to extract the information on the screen, but this is not a proper or even a long-term solution. [2][20]

Since technology has progressed so much and it is now capable to provide a system that can present complicated data in such an informative manner that the user can easily understand. The data available in the superfamily database is being used in a specific manner, people who want to use that data and process it in a different manner cannot do so directly.

Moreover, if I talk about supporting more platforms it would require that the user would have to actually deal with SQL servers themselves and know SQL. All of which is not suitable for a naive user such as a Biologist. People on the other hand understand the use of a website.

The amount of information available today is immense, hence it is getting difficult to analyze and understand the entire information we have. Information is increasing rapidly whereas the methods to visualize the data have not changed that often. Thus researchers are faced with a problem, as they cannot analyze such amount of data, if it is not represented correctly. Visualizing data in a manner that is scalable has always been a challenge in field of computer science.

Information that needs to be visualized is huge and but that is not the only thing. Visualizing that much information is a tough but what makes it a challenge is the type of the data the biological sciences has, data is not only non-homogenous it also scales in a non-linear way in places, and the relative importance is biological and numerical. There can be infinite number of mathematical projections and possible visualizations, but the challenge is to find the one that scientifically is the most meaningful and helps the biologist visualize the data.

We looked at the biological importance of the data and its characteristics, but from computer science point of view the data that I have is complex real-world data and which should make sense in an analytical way. To visualize this type of data that should not only make sense in a biological manner but should have visualization that makes it easy, for the user. Even with the current advancements in technology it makes it a challenge to visualize this data. This approach to visualization frees the biologist to define what should be visualized and how, rather than limiting the analysis to something that is statically composed offline.

### **Initial Business Case**

Considering all the above facts and considering the importance of web services in today's world, I have developed in this project, web services to solve the problems and give the freedom to researchers to use the data as they wish, so that they can carry out their research more easily. As all the users do not wish to process the data further, a visualization method is designed to help users visualize the huge amount of information in a manner that is suitable, fast and appropriate to the needs of the users. Visualization technique does not only focus on this field or type of data, technique developed can be applied on any data, similar to this and the concept can be used in many of current visualization methods.

This project introduces a new way to visualize the data in bioinformatics. This is a new approach and will have its advantages and disadvantages. Data visualization implemented in the system is such that user only sees that information that he requests and so the information visualized would be accurate and exactly what the user wants.

### **How is this different?**

The important question is what is so different about this project. What makes it's so challenging and worth making. This system tries to overcome the limitations of the system that already exists. Some of these systems have web services to provide data to researches but they are very limited. Secondly the visualization is informative but it cannot be easily understood and there is a lot of data, or the data being shown is not enough or easily navigable, or customizable to the specific science question the user has.

This project overcomes these limitations by providing many of the functionality by web services enabling researchers to get the data in a standard format so they can process it further. Visualization is very different from what is being currently offered by similar projects. Information visualization is not an easy task, the challenge is to make the visualization informative and yet easily understood, and that the information being displayed is never too much or too less.

- Improved Visualization
- Visualization based on requirements
- Customizable Visualization
- Many more features through web service

This differentiates this system from all others, it gives informative visualization of the data with many options to change, how the data is displayed, and that data is only shown if requested by the user. Once data has been requested it is rendered to the visualization. Enabling user to view just the data he/she wants on the screen instead of having to search through a lot of information to get the information they need.

### **Proposed Solution**

Information visualization is very important in bioinformatics as the field is known for its vast amount of data and the type of the data it has, as discussed above. Information visualization in the field of bioinformatics has some steps as defined by Lang *et al* [29], these steps include

- Data acquisition

- Data mapping
- Rendering

### Data acquisition

This step deals with acquiring the required data from the database and processing it so that it can be used in the next step. Once the data has been acquired the data manipulation can begin which focuses on formatting the data in a format so that it can be used in data mapping. [29]

### Data mapping

This step requires that the data retrieved from the first step is now mapped to shapes and their characteristics are defined such as color, size & location. In this step the shapes are decided in which the data is to be shown and different IV techniques are used for this. [29]

### Rendering

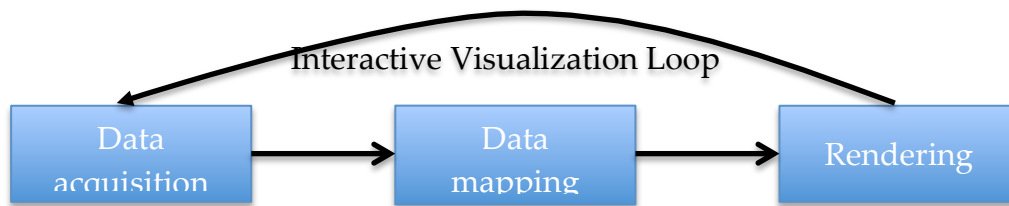
This is the final step once data mapping has been done it is time to render this data to the visualization so it can be visualized on the screen and user can see the data. [29]



This method works fine in visualizing the data but if the amount of data is huge than the visualization will be too big to understand and detect patterns for a researcher. In this project the data set that is considered to visualize is huge, this technique needs to be altered in order to fit the scenario.

Allowing user to decide how much information and what information he/she wants is the key to this project the technique is altered so that the data acquisition is repeated when ever a request is made and rendering accumulates everything so that as the user moves forward he/she can still go back in the visualization.

Data mapping is not a major portion as the visualization that is being considered at this point is a tree where the location, color and size are already decided, but this does not mean it has been removed because as this technique will not only apply to tree but other visualizations as well.



So this allows going from rendering back to data acquisition. Once the data has been rendered, user can select any information he wants from the rendered view a request is made for data acquisition and then is data is mapped, finally it is rendered again on the screen but this time is altered. The new data is added to the already rendered data on the screen. Hence showing only the information the user needs instead of rendering all the information that there is, which makes this a powerful tool for visualization in bioinformatics.

This technique focuses on how data can be delivered and rendered and that the information shown is what the user wants. How to display that information in a manner that user can easily adapt and understand the visualization is another thing.

It is very important to know how the data will be displayed what visualization will be used and will it fulfill the objective. This is very challenging as the data at hand has unique characteristics.

When implementing the hierarchy of genomes the best way to show such visualization was to use a tree graph. I have already discussed some of the java script libraries, which has been used in order to implement this solution. The visualization methods used have been selected on the basis on three things

- Gives user visualization options
- Can be changed/altered to adapt the technique above
- Information can be visualized clearly

It is hard to measure difference between visualization for that purpose a criteria is set to judge whether visualization is informative and appropriate for the dataset being considered or not. The criterion that was considered for judging visualization has been discussed above. Much visualization were considered but rejected as they failed one or more of the things above, such as PlotKit, flot and Rgraph etc. These libraries did not meet the criteria discussed



above. The libraries `protovis`, JavaScript InfoVis Toolkit, and `highcharts` all fulfill the above criteria and were considered. The visualizations were selected after shortlisting the different visualizations that met the criteria and then discussing these visualization with the supervisor and advisor. The one, which was most suitable for viewing the type of data being considered, was used to ensure that the visualization is informative.

The second problem is unavailability of data in a standard format from the SUPFERFAMILY database. Some previous approaches have used different methods to solve this problem; the method I choose to go ahead with is to make web services. Web services are standard and are used in most of the tools being developed as it has many benefits. We discussed the web services above and to solve this problem I will use SOAP, as it is an industry standard and more general than RESTful. So if the system is expanded later on it will not be limited by the limitations of the RESTful protocol.

## Technology

Java is my first choice of the language to use, as it is an open source language the tools available for it, like Eclipse are free of cost. Whereas using .NET would require that I have MS Visual Studio and would be platform dependent. As this project might be used in the university I have given preference to Java so that it can be hosted on the university servers without any problems. [17]

Mainly Java, Java Script, JSP and AJAX have been used to implement the system. AJAX has been used for retrieving the data asynchronously where as Java Script libraries have been used for visualization.

## Major Tools

Item	Applied for
<b>Tools</b>	
MS Visio	Design
Eclipse	Implementation
MySQL	Database
JavaScript Libraries	jQuery based visualizations
<b>Languages</b>	
UML	Design
Java	Web Services and Server Side Scripting
Java Script & AJAX	Client Side Scripting

Table 2: Major Tools used in the Project

In this project for designing the system I have used MS Visio, all the diagrams are in UML and made using Visio software. Eclipse tool has been used for

complete implementation including the data visualization and web services. The database being used is in MySQL. AJAX has been used for quick retrieval and fast interactive visualizations.

### Target Platform:

Item	Specifications
OS	All current generation
Browsers	All current generation

Table 3: Platforms Supported

The system developed will work on all current generation systems and will support all current generation browsers as well.

### Basic design

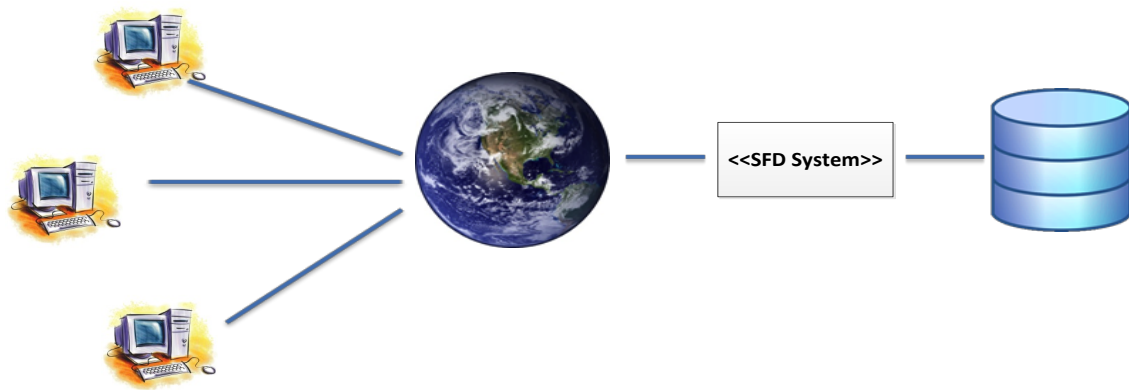


Figure 10: Basic design of the system

This basic diagram shows the basic structure of the system. The system have used the database and provided a WSDL interface on the Internet and everyone who wants to use it can access it through the Internet. Not only the people wishing to view the data but also the people who want to process it further can access the system and retrieve information in a standard form. The system provides both a web service interface and a web page form, on which the data can be viewed.

This also allows the people to use the web services and create their own system using the data from the web service so different platforms can be added as well.

## Non-functional Requirements

- Available 24/7.
- User friendly.
- Scalable

These non-functional requirements were considered when designing and developing the system. These requirements will be met in different stages of the project some would be met in designing where as some would be met in implementation.

## Initial Use Case

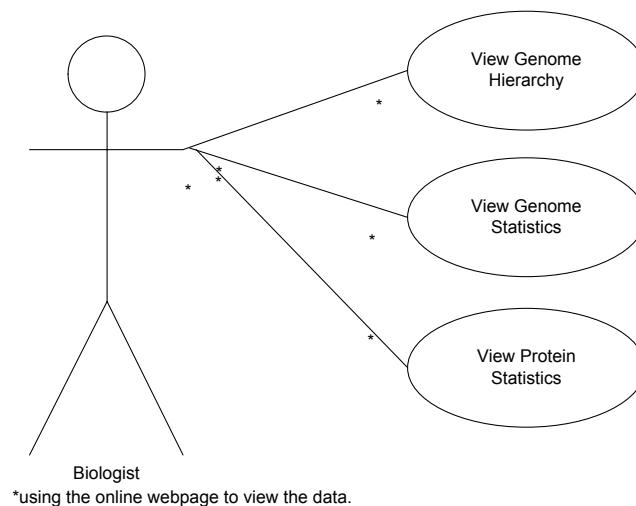


Figure 11: Biologist initial use case

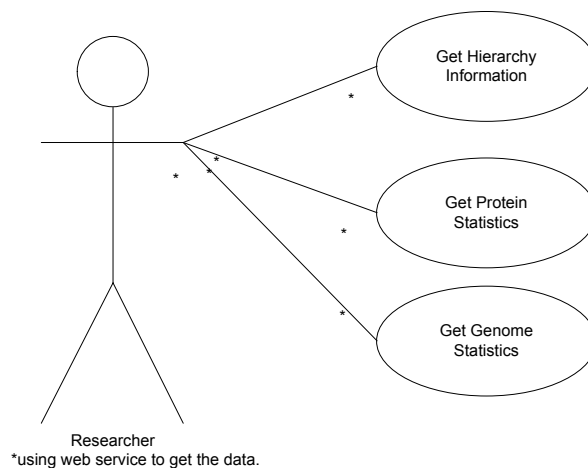


Figure 12: Researcher initial use case

Use case shows in the simplest way that which user would use the system for what functionality. Here the biologist would use the system to view the data

and the researcher would use the system to retrieve data from the web services so that he/she can process it further.

The initial model does not tell any details of how these functionalities will be achieved to understand a little more on how these functionalities will be performed, detailed use case model has been designed. It shows what options user has to view the data and finally the actual data that use wishes to see.

## Detailed Use Case

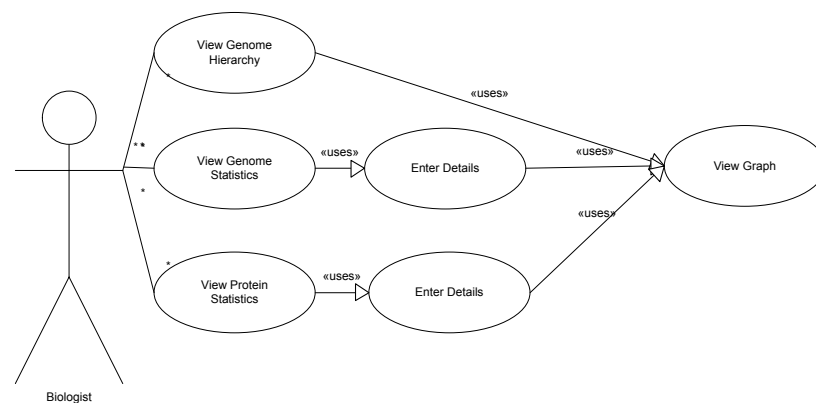


Figure 13: Detailed biologist use case

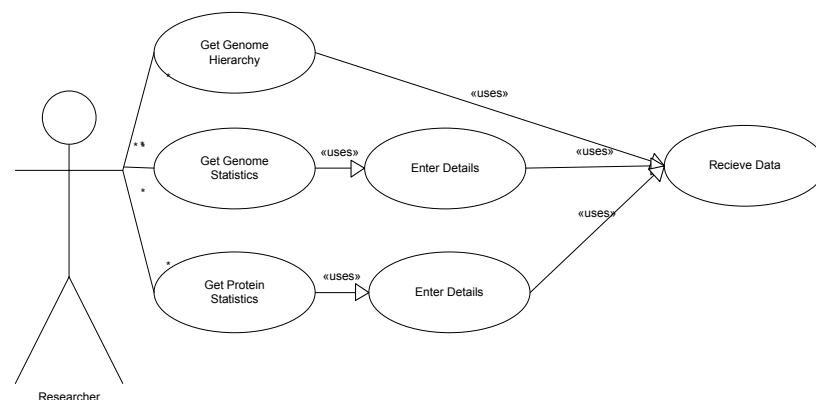


Figure 14: Detailed researcher use case

The detailed use case shows how biologist will view the data and what steps are used to get there. Biologist can search for which he needs to enter details. He/she can also select a genome or protein and view the graph.

Researcher can receive the data by providing the details about what he/she wants and make that request to the web service. Web service would than send the required data to the researcher in a standard format.

## Class Diagram

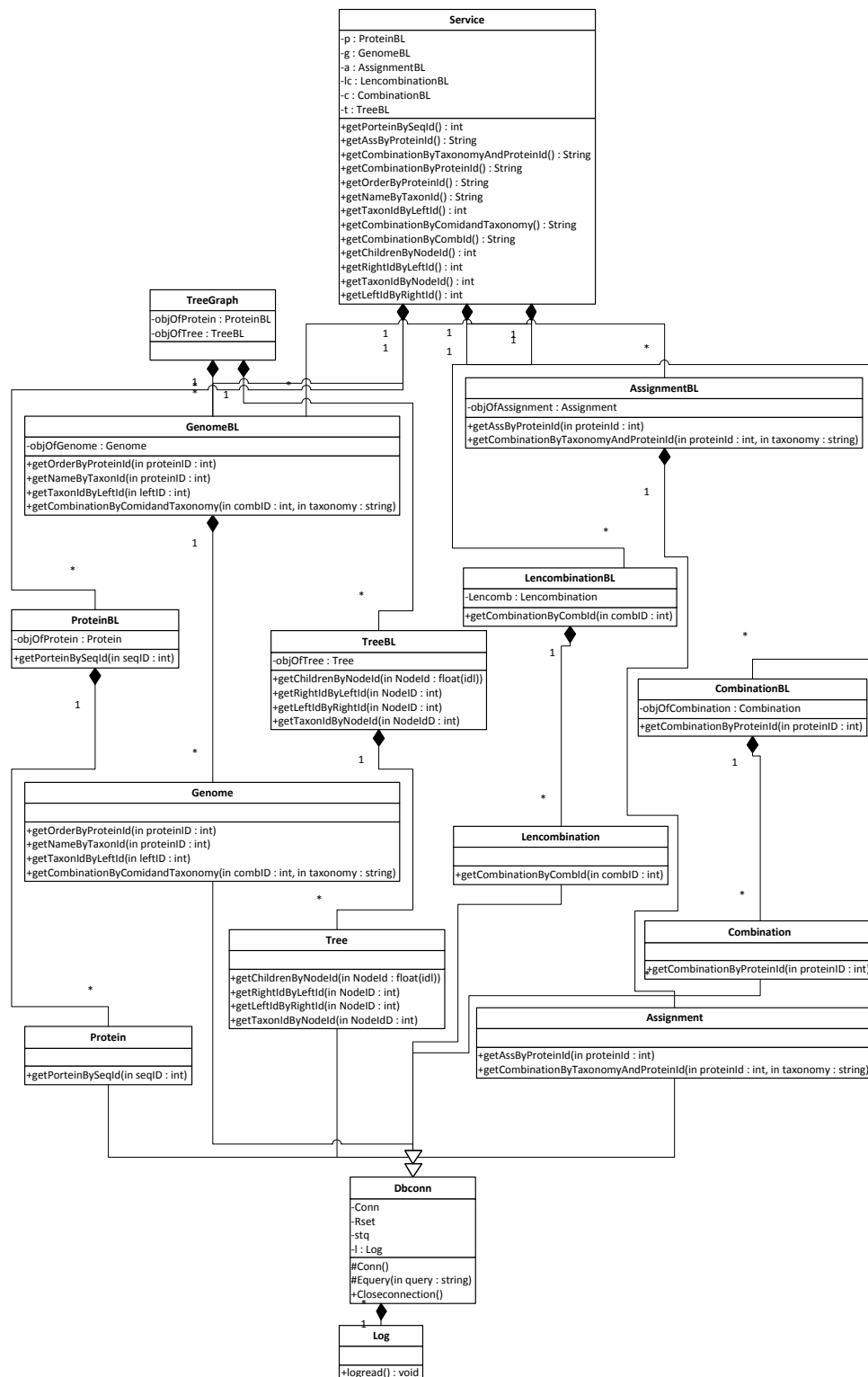


Figure 15: Class diagram for SFD

The class diagram shows the implemented classes and the design of the system. Each of the modules has been implemented in a separate class having a separate DAO. All the DAO's are inherited by Dbconn, which has methods

to connect to the database and execute queries. This allows the system to be expandable later on. As each of the modules are in a separate class new methods and functionalities can be added without any problem. OO patterns were also considered such as factory. It is also important that the new functionalities that are added are available through web service. For this there is a service class that lists all the functions currently being provided by the web service. Once a new functionality is added to the system in one or more of the modules all the developer has to do is link those functionalities in the service class and that will ensure that those methods are now also available.

This design maintains a low coupling between the classes and shares the data based on data encapsulation. This ensures that when a module is updated it can be easily changed and does not affect other modules, as it has less inter-module dependency.

Some times designing a system can lead to over-design which is a waste of time and requires more time in adding functionalities to the system as well. To ensure this does not happen I have used traceability matrix Figure 16. according to the requirements of the project. Each requirement is mapped to the class that it is using to see whether any of the classes are useless and does not need to be created.

Requirement	Tree /Tree Graph	Genome	Protein	Combinat ion	Len Combination	Assignment
User can view tree	✓	✓				
User can search genome		✓				
User can Search Protein			✓			
User can view comparison by taxonomy		✓	✓	✓	✓	✓

Figure 16: Traceability matrix for class diagram

Each requirement is met through a class, which performs further functions to fulfill that request. Matrix in Figure 16 shows information about requirements and classes used by each of those requirements.

## Sequence Diagrams

### Statistics of Genome

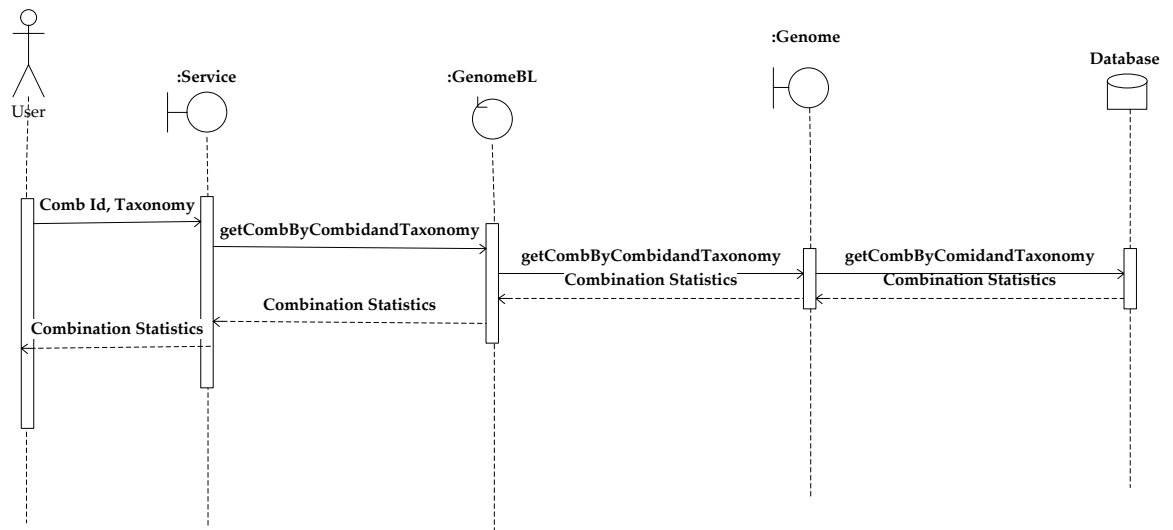


Figure 17: Sequence diagram for Statistics of genome

Above sequence diagram clearly shows how the process works, in this case how a researcher can retrieve statistics of a genome. The researcher first calls the web service and gives details like combination id and taxonomy the service then call the genomeBL class, which is the business logic class. GenomeBL then calls the genome class, which is the DAO. The genome requests the information from the database and sends it back to the GenomeBL which sends it back to the service and finally to the user.

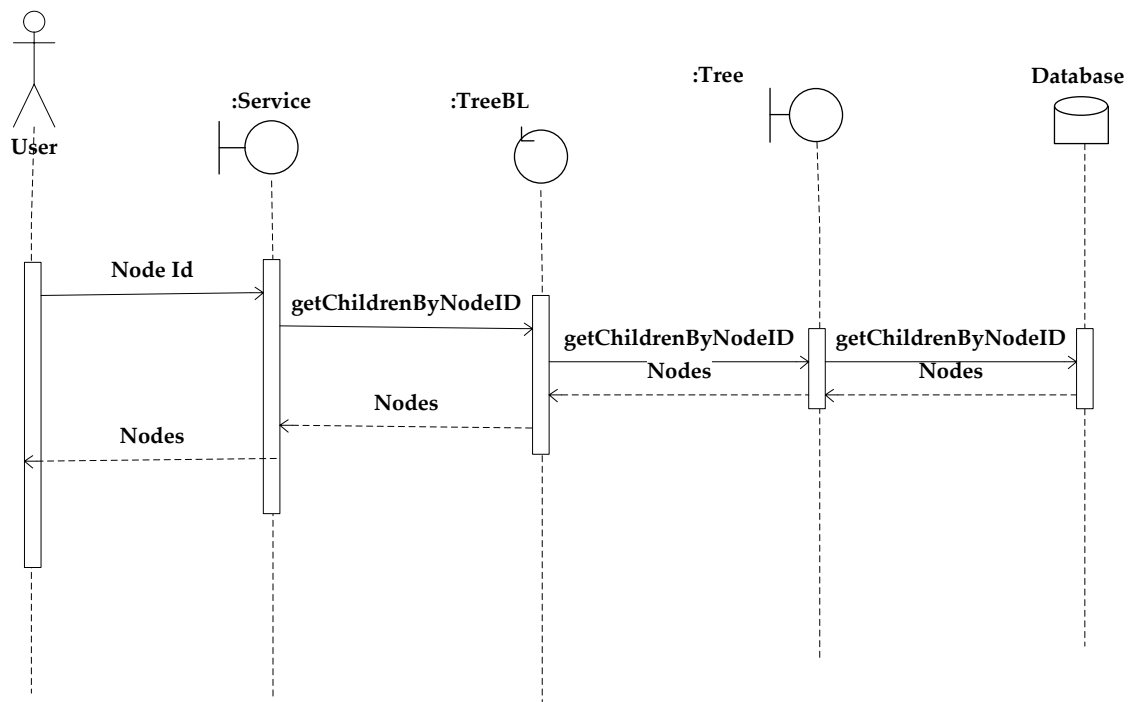
**Children by NodeID**

Figure 18: Sequence diagram for Children by Node ID

The sequence diagram above shows how children for a specific node can be retrieved. Researcher first calls the service and sends the node id. The service then calls the TreeBL class, which is the business logic class. TreeBL then calls the Tree class which acts as a DAO for Tree which gets the information from the database and sends it back to the TreeBL which sends it to the service and finally to the user.



## System Architecture Diagram

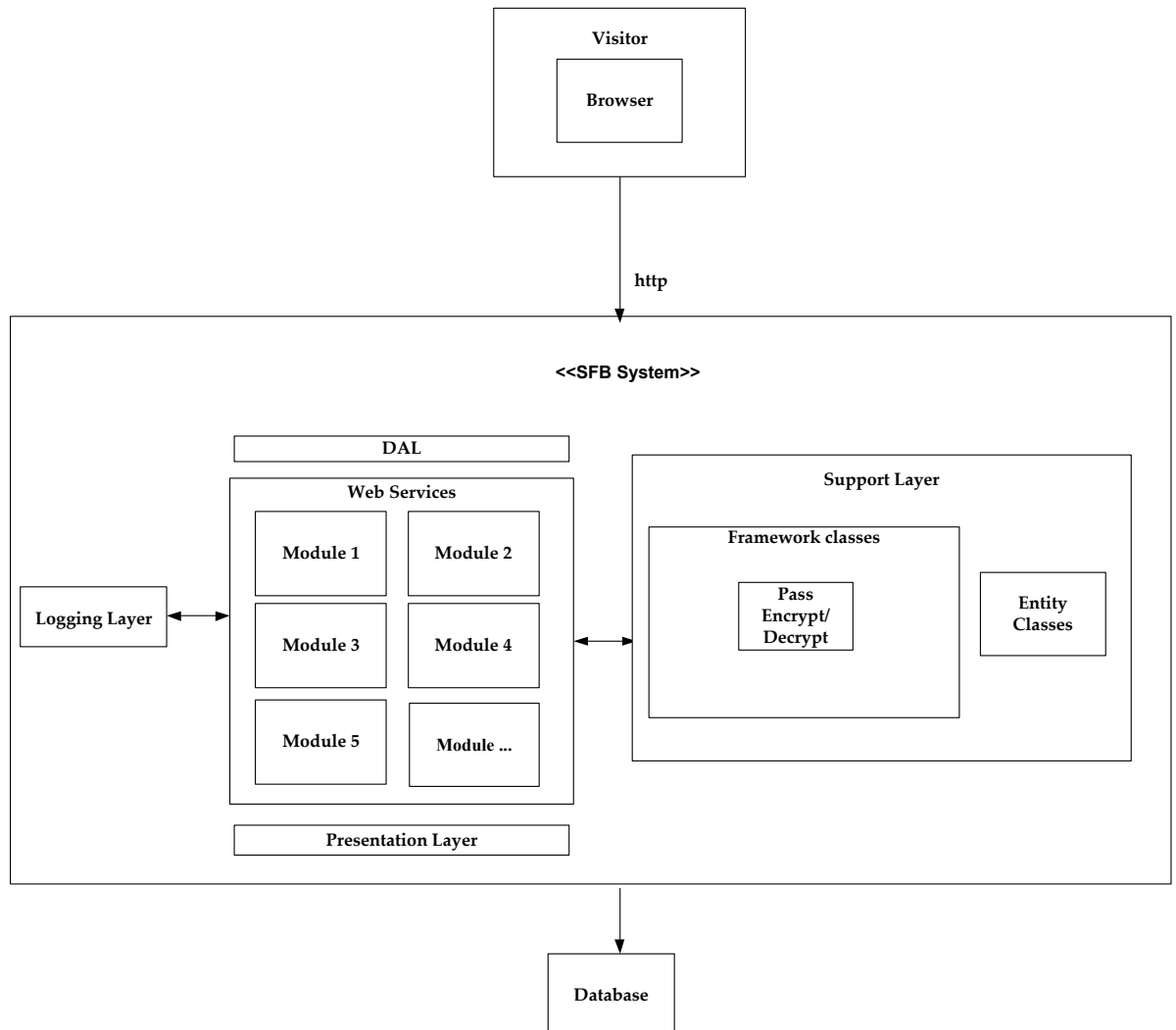


Figure 19: System architecture diagram for SFD

The system architecture gives detailed information about the system and how the system works. The system is based on 5 basic layers Data Access Layer that deals with data access objects that retrieves the information from the database; Presentation Layer deals with the display of the data that is being retrieved from the database. Whereas the Web Services provides a WSDL interface to system so that the data can be retrieved in a form that can be computed easily rather than just available through web page. Further more we have a Logging Layer; it is used to log details. Support Layer have the entity classes and the framework classes these both support the system. The system architecture has been designed keeping in mind the fact that the system might be expanded in the future, to minimize the complexity and to

keep the expansion simple for the system, the system has been divided into layers.

## Deployment Diagram

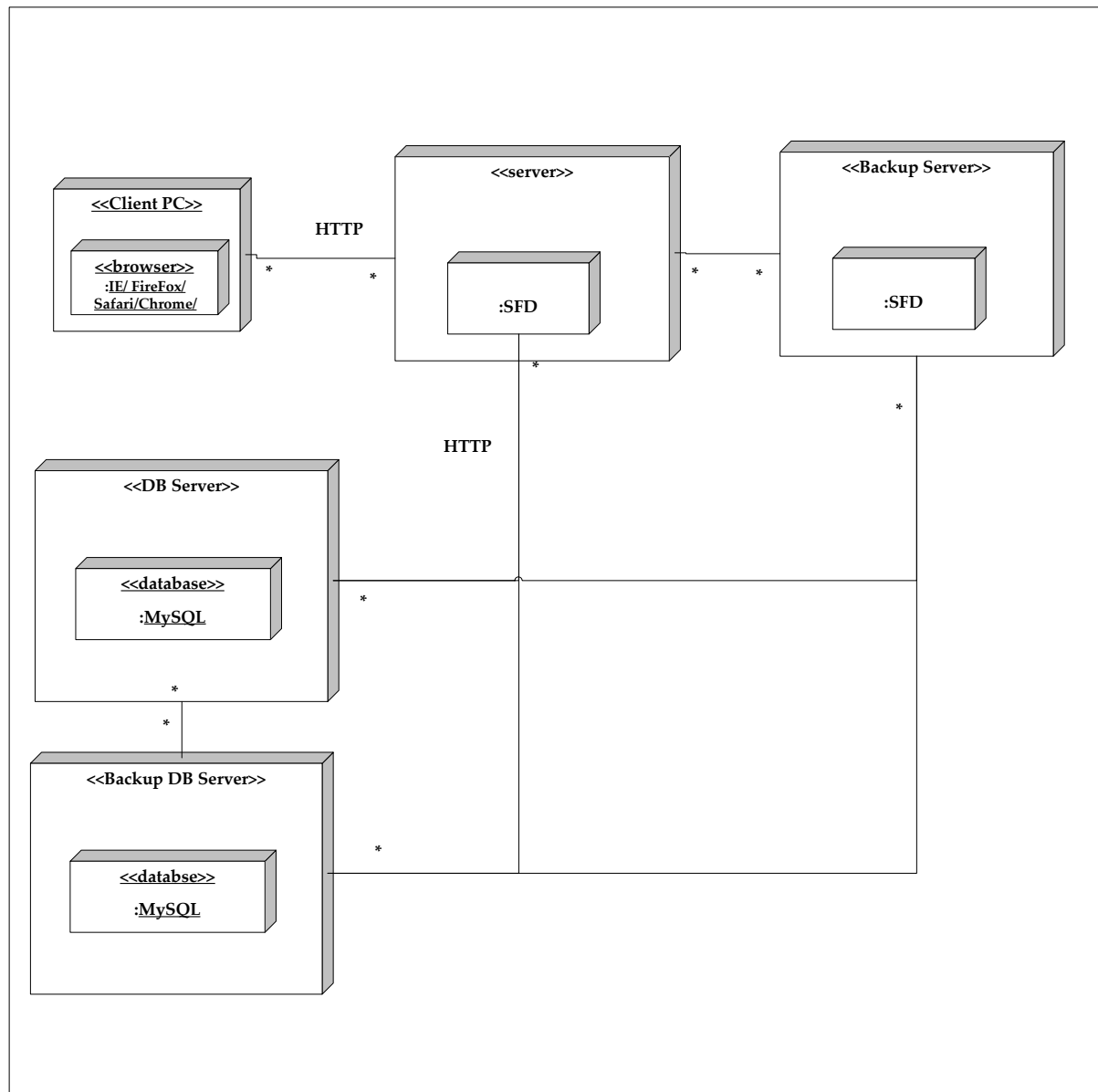


Figure 20: Possible deployment diagram

Deployment diagram shows a possible deployment for the system. It has been designed keeping in mind that the some of the non-functional requirements are met. Both the database server and hosting server has a backup in case anyone of the server goes down. It is also necessary to have mirrored databases.

Consider the system that is currently the development system for the project. For this my laptop is the server for the SFD system as well as the database server. If my system crashes I will not be able to access any of the web services. But let us assume we have a backup but the database is not mirrored. So the backup started serving the requests, the information on the backup database is out-dated, why? Because the backup database was not mirrored with the main server the information was never updated, hence the information is lost. Hence it is important to have mirrored databases.

## Database Schema:

### Entity-Relationship Model:

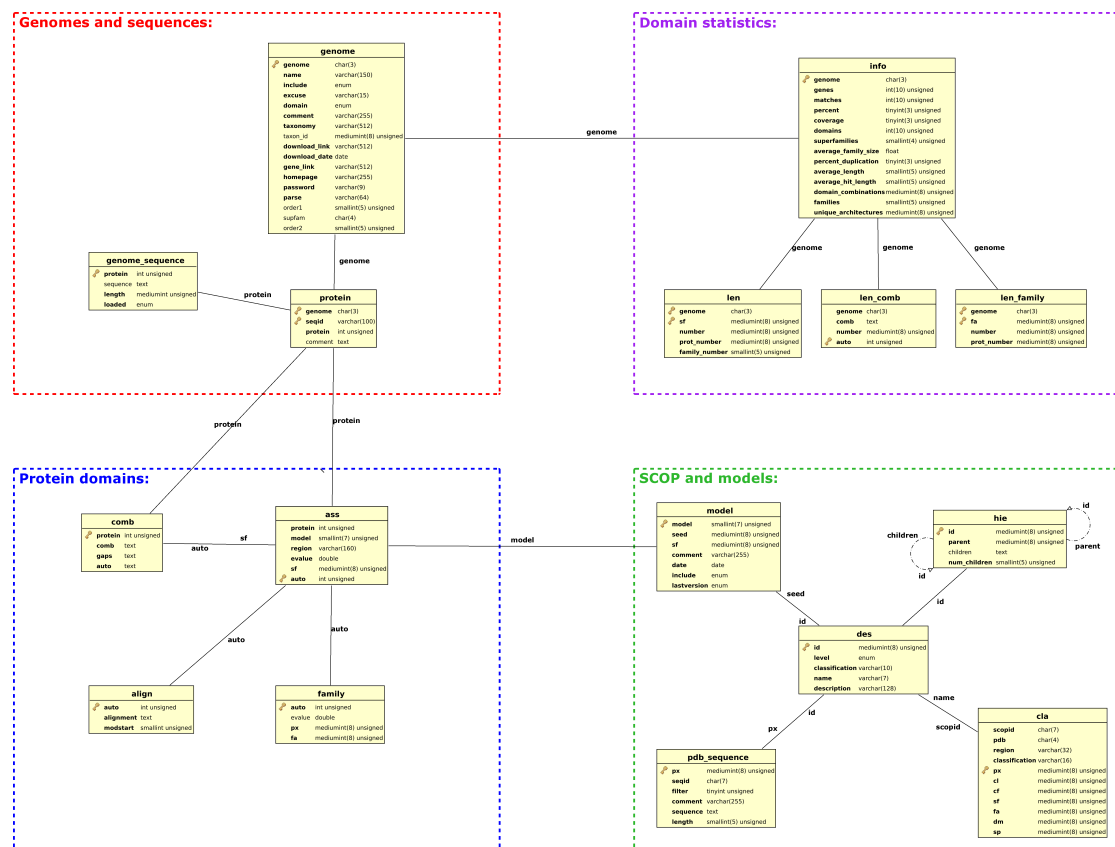


Figure 21: ERM model of the database

Source [4]

Above image shows the ERM diagram of the database that the system is developed for, it outlines all the tables and their relations. The class choices that I made during the implementation were based on this model. As it has a Protein, Genome, Tree table that holds the respective information about the relation. The system has these classes as well, which perform the functionality related to these tables. We also have tree graph class in the system, which uses the functionality in the tree class and manipulates the data to form

binary tree visualization. Moreover we have ass, comb and len comb which holds statics data about proteins and genomes.

## Guidelines, Standards & Templates

### Coding Standards

- Object oriented approach
  - Classes
  - Functions
  - 3-Tier Architecture
    - Controller Classes
    - Interface Classes
    - Entity Classes

### Data Naming Standards

I have followed those names that satisfy the "name for clarity" objective. This approach says that all the names should have a concise, descriptive definition.

### Development Tool Standards

Throughout the project, I have used the following tools:

- Eclipse – For development purpose
- MySQL - For Database.
- Apache – For hosting the website and web service

### Documentation Standards

Documentation standards have been followed by Roger S. Pressmen's Software Engineering: A Practitioner's Approach. Moreover, few of the sample templates are also downloaded from the internet; IEEE standard documents are mostly opted.

Document Name	Descriptions	Standards and Guidelines
Use Case Model Sequence Diagrams	Analysis Modeling based on the artifacts of UML	UML
Software Architecture Deployment Diagram	Modeling of Analysis into Design Phase	UML

Table 4: Documentation Standards

Standards have been used to ensure that everyone understands all the documents. In an event when expansion of the current system is required any developer will be able to easily understand and change the system according

to the requirements. These standards also ensure that the system have been built, designed and tested properly and hence the chance of failure reduces.

Testing standards are also used to document the tests done on the system the unit testing has been done on the system to ensure each unit of source is working in accordance with the requirement each of the units have been tested twice to ensure system does not have a bug. Each of the units has been tested separately and the final testing results are shown below.

## Testing

SFB testing activity includes following testing strategies

### Unit Testing

- **How** – Make short test methods to test the code at different levels
- **When** – Unit testing was done simultaneously with the development of small units.
- **Why** – To assure that all the individual units of source code are fit for use.
- **On What** – On every individual unit of the source code
- **Test Iterations** – Two Iterations

<b>Class Name</b>	Assignment				
<b>Method Name</b>	getAssByProteinId				
<b>Test Class Name</b>	AssignmentTest				
<b>Test Method Name</b>	getAssByProteinId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – protein ID	Valid Protein ID	Valid Protein ID	True	Pass	None

<b>Class Name</b>	Assignment				
<b>Method Name</b>	getCombinationByTaxonomyAndProteinId				
<b>Test Class Name</b>	AssignmentTest				
<b>Test Method Name</b>	getCombinationByTaxonomyAndProteinId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – protein ID	Valid Protein ID	Valid Protein ID	True	Pass	None
String – taxonomy	Valid taxonomy	Valid taxonomy	True	Pass	None

<b>Class Name</b>	Combination				
<b>Method Name</b>	getCombinationByTaxonomyAndProteinId				
<b>Test Class Name</b>	CombinationTest				
<b>Test Method Name</b>	getCombinationByTaxonomyAndProteinId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – protein ID	Valid Protein ID	Valid Protein ID	True	Pass	None

<b>Class Name</b>	Genome				
<b>Method Name</b>	getOrderByProteinId				
<b>Test Class Name</b>	GenomeTest				
<b>Test Method Name</b>	getOrderByProteinId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
String – taxonomy	Valid taxonomy	Valid taxonomy	True	Pass	None

<b>Class Name</b>	Genome				
<b>Method Name</b>	getNameByTaxonId				
<b>Test Class Name</b>	GenomeTest				
<b>Test Method Name</b>	getNameByTaxonId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – taxon ID	Valid taxon ID	Valid taxon ID	True	Pass	None

<b>Class Name</b>	Genome				
<b>Method Name</b>	getTaxonIdByLeftId				
<b>Test Class Name</b>	GenomeTest				
<b>Test Method Name</b>	getTaxonIdByLeftId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – taxon ID	Valid taxon ID	Valid taxon ID	True	Pass	None

<b>Class Name</b>	Genome				
<b>Method Name</b>	getCombinationByComidandTaxonomy				
<b>Test Class Name</b>	GenomeTest				
<b>Test Method Name</b>	getCombinationByComidandTaxonomy ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – comb ID	Valid comb ID	Valid comb ID	True	Pass	None
String – taxonomy	Valid taxonomy	Valid taxonomy	True	Pass	None

<b>Class Name</b>	Lencombination				
<b>Method Name</b>	getCombinationByCombId				
<b>Test Class Name</b>	LencombinationTest				
<b>Test Method Name</b>	getCombinationByCombId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – comb ID	Valid comb ID	Valid comb ID	True	Pass	None

<b>Class Name</b>	Protein				
<b>Method Name</b>	getPorteinBySeqId				
<b>Test Class Name</b>	ProteinTest				
<b>Test Method Name</b>	getPorteinBySeqId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
String – seq ID	Valid seq ID	Valid seq ID	True	Pass	None

<b>Class Name</b>	Tree				
<b>Method Name</b>	getChildrenByNodeId				
<b>Test Class Name</b>	TreeTest				
<b>Test Method Name</b>	getChildrenByNodeId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – node ID	Valid node ID	Valid node ID	True	Pass	None

<b>Class Name</b>	Tree				
<b>Method Name</b>	getRightIdByLeftId				
<b>Test Class Name</b>	TreeTest				
<b>Test Method Name</b>	getRightIdByLeftId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – left ID	Valid left ID	Valid left ID	True	Pass	None

<b>Class Name</b>	Tree				
<b>Method Name</b>	getLeftIdByRightId				
<b>Test Class Name</b>	TreeTest				
<b>Test Method Name</b>	getLeftIdByRightId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – right ID	Valid right ID	Valid right ID	True	Pass	None

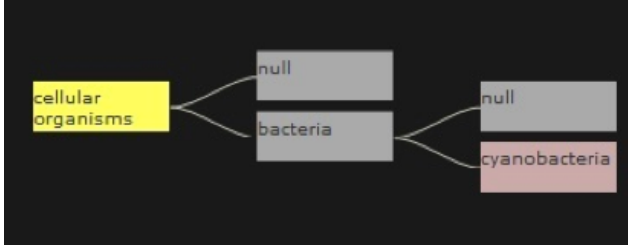
<b>Class Name</b>	Tree				
<b>Method Name</b>	getTaxonIdByNodeId				
<b>Test Class Name</b>	TreeTest				
<b>Test Method Name</b>	getTaxonIdByNodeId ()				
<b>Input (Method Parameters)</b>	<b>Expected Value</b>	<b>Actual Value</b>	<b>Expected Result</b>	<b>Result (Pass / Fail)</b>	<b>Error</b>
Integer – node ID	Valid node ID	Valid node ID	True	Pass	None

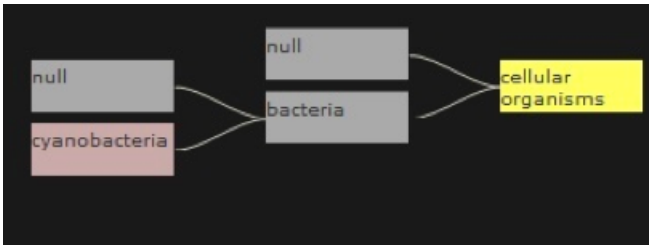
The results of unit testing shows that testing was satisfactory and no problems were detected at unit testing level and hence the system will not face any problems at this level. Each of the classes have been tested by creating a test class for that specific class, and all the methods in the class were tested for problems.

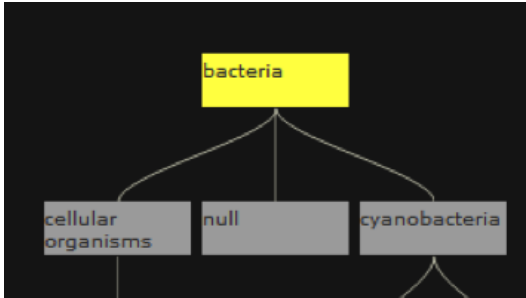
Different cases were considered including null values, positive values, negative values, and actual values. The methods were changed to ensure no problems arise in any of cases. Integration testing was also done on the system. All the modules were tested once they were integrated with each other to form the complete system.



## User Testing

<b>Test case Purpose:</b>	To check whether a user can traverse through the tree
<b>Pre requisite:</b>	Tree visualization with a node that has children.
<b>Page:</b>	TreeGraph
<b>Procedure:</b>	<ol style="list-style-type: none"> <li>1. Enter the Url in the browser</li> <li>2. Click the node</li> <li>3. Wait for the tree to populate</li> </ol>
<b>Screenshot</b>	
<b>Expected Results:</b>	Node loaded
<b>Pass/Fail:</b>	Pass

<b>Test case Purpose:</b>	To check whether a user can use the tree orientation
<b>Pre requisite:</b>	Tree visualization
<b>Page :</b>	TreeGraph
<b>Procedure:</b>	<ol style="list-style-type: none"> <li>1. Enter the Url in the browser</li> <li>2. Click the desired orientation</li> <li>3. Wait for the tree to adjust</li> </ol>
<b>Screenshot</b>	
<b>Expected Results:</b>	Orientation changed
<b>Pass/Fail:</b>	Pass

<b>Test case Purpose:</b>	To check whether a user can change the root
<b>Pre requisite:</b>	Tree visualization
<b>Page:</b>	TreeGraph
<b>Procedure:</b>	<ol style="list-style-type: none"> <li>1. Enter the Url in the browser</li> <li>2. Click the set as root option</li> <li>3. Wait for the tree to adjust</li> </ol>
<b>Screenshot</b>	
<b>Expected Results:</b>	Root changed
<b>Pass/Fail:</b>	Pass

<b>Test case Purpose:</b>	To check whether web service would accept wrong parameters
<b>Pre requisite:</b>	Web Service available online
<b>Method:</b>	getChildrenByNodeId
<b>Procedure:</b>	<ol style="list-style-type: none"> <li>1. Select the method to be called</li> <li>2. Give letter instead of a number</li> </ol>
<b>Result</b>	Error validating parameter
<b>Expected Results:</b>	Error occurs
<b>Pass/Fail:</b>	Pass

Similarly more testing was done to ensure that system does not break when user is using the interactive visualization.

## Results

### Data Visualization

Data visualization is the key in this project and the technique developed above has been used to implement the system, which allows user to have a different and more effective method to visualize large amount of data. The result of the implementation gives a good idea about how a user can visualize the data.

I have implemented two different visualizations to consider the possibilities of this technique being applied on data from any other field. The first implementation is for a tree graph. As the tree has more than 3000 nodes visualizing it becomes a problem. Graph implemented focuses on data visualization and gives user the option to view the data as he desires.

#### Tree Orientation

- Left ☐
- Top ☒
- Bottom ☐
- Right ☐

Figure 22: Tree Orientation in SFD

To accomplish this tree orientation feature is there which allows the tree to be displayed in four different ways, user can change the orientation of tree to left, top, bottom and right.

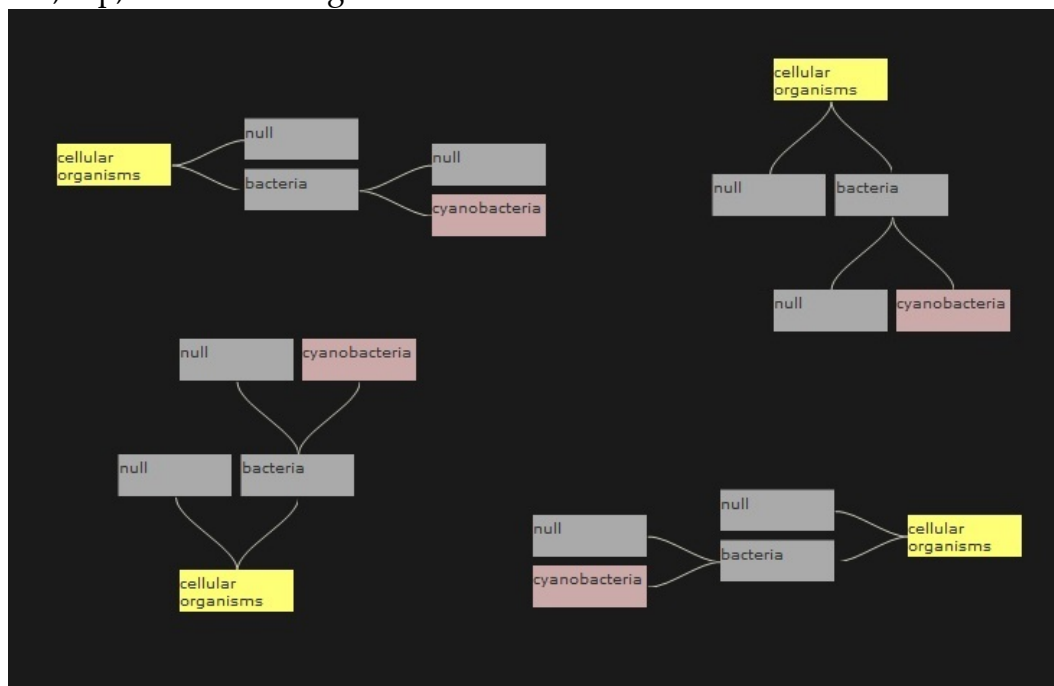


Figure 23: Results of Tree Orientation

Figure 23 shows different tree orientations that a user can select from. Tree orientation can even be done once the graph is loaded. Allowing user to view data in any orientation without even refreshing a page. Tree can be adjusted at any point regardless of any depth of the tree the user is at. The nodes allows user to traverse through the tree. The node selected is brought to the center so that the user can visualize both portions of tree ahead and the point from where he got to the node.

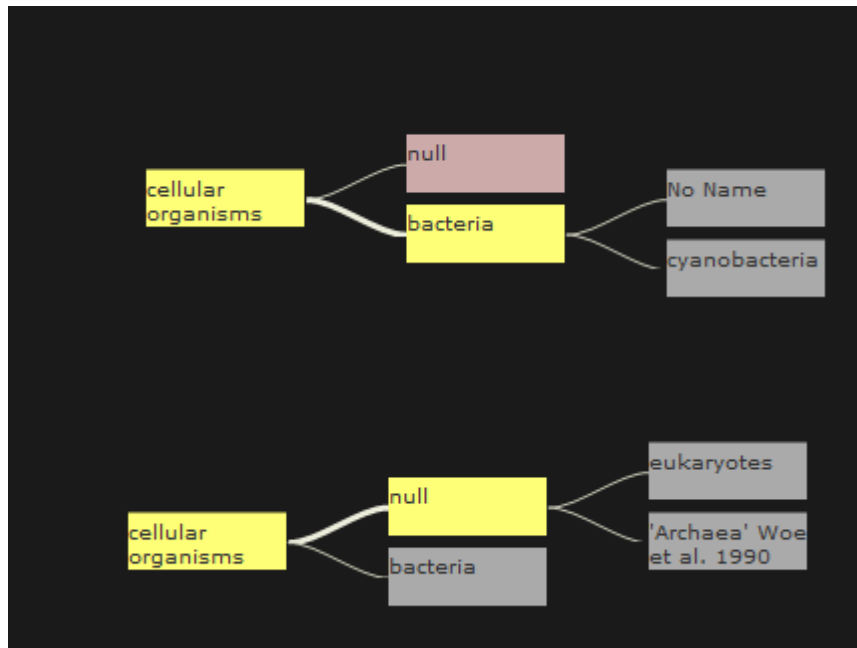


Figure 24: Results of traversing through tree

As the technique focuses on targeting huge amount of data the graph shows only the nodes that have been requested by the user. If the user reaches a point where they do not see any further nodes they can select the last node to load the next set of nodes. Once the nodes have been loaded the center is adjusted again to node select allowing user to view the newly loaded nodes. This is done through AJAX so that once the user starts to visualize the data he does not have to refresh or reload the page. The data that is loaded aggregates with the already loaded graph giving user more information but still only the information that the user requires. So a user can move from level 1 of the tree to level 10 and still only see the information user wishes to. This only does not allow user to skip a lot of nodes that are not important for the user but also gives a lot more space to the user to load more nodes and still have a visualization which provides the information in a manner that can be understood easily.

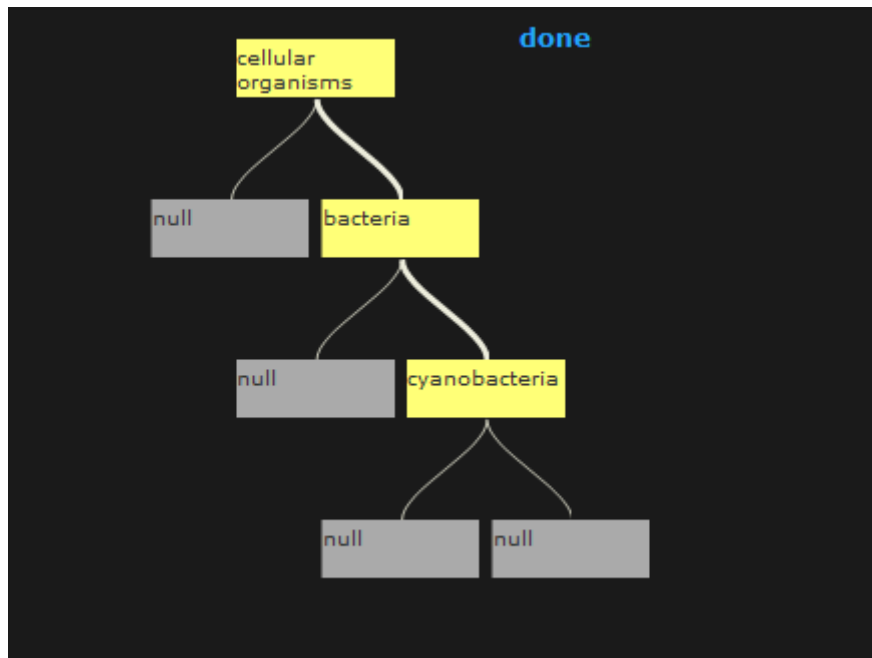


Figure 25: Results of path

It can be important sometimes to know how you managed to get to the position at which you are now in the tree, if you are 20 or more levels deep. This visualization keeps track of the your path and changes the color of the nodes to yellow through which you manage to get to the point you are at now. Ensuring that you do not loose the way to node at hand at any point during the visualization experience provided. Even loading new nodes does not affect the path.

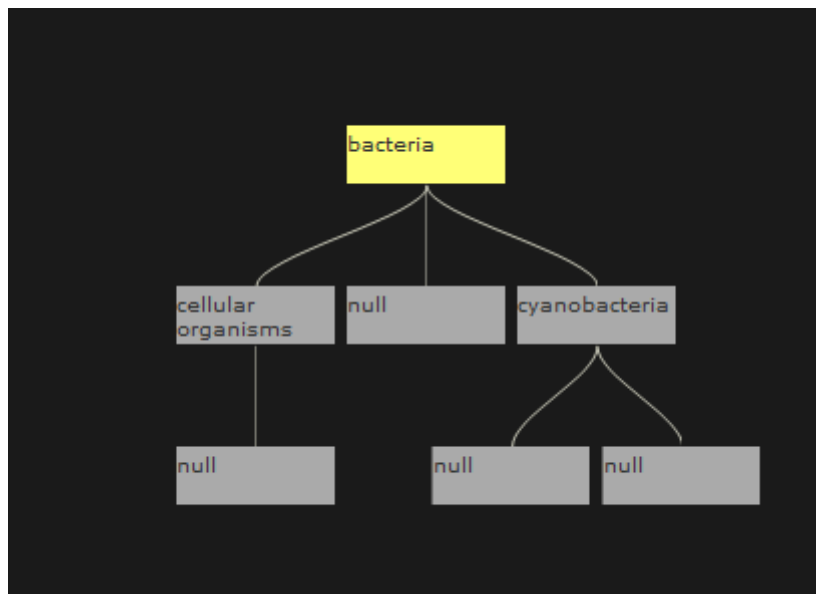


Figure 26: Results for changing root

Sometimes when researching it is required that you have a fresh look at things, a new perspective that can enable you to consider possibilities you did

not think of before. The visualization method tries to achieve this by letting the user select the root node if he wishes to. The actual root is selected by default, if the user wants to change the root node and consider looking at the information from a different point of view user can do so just by selecting the set as root option and selecting the node to convert to root. This process also does not require user to refresh or reload the page so user can enjoy hassle free visualization experience.

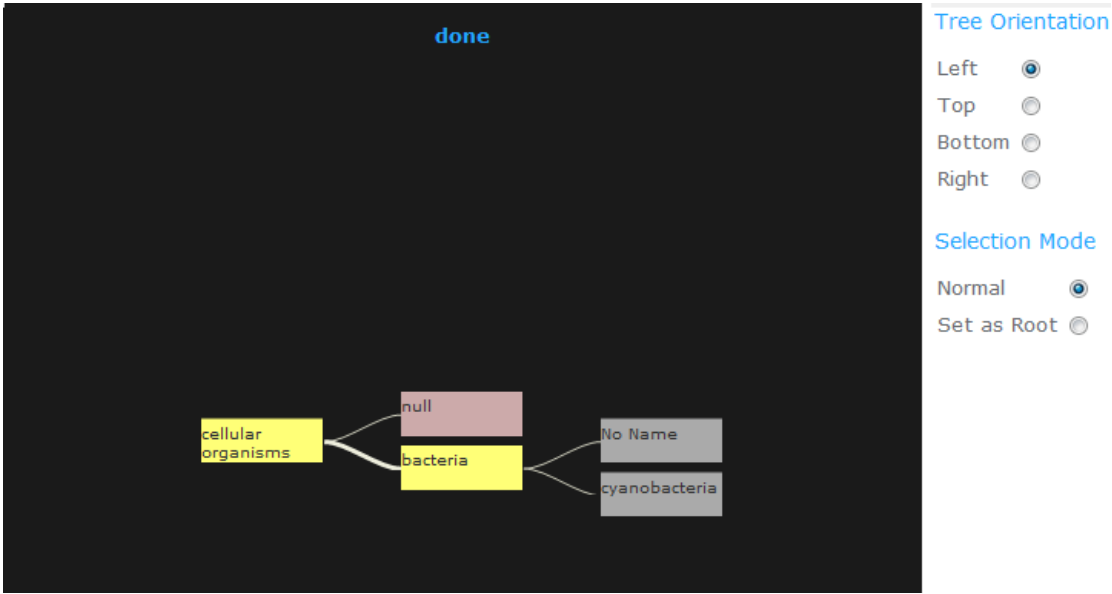


Figure 27: Tree implementation

This graphs enables different and more efficient method to visualize huge amount of data. Giving user the freedom to select the data he wants view and as he wants to view by selecting the tree orientation, it stores the path so user knows how he got to the position even at a depth of 50 levels and finally it allows user to view the data in a way he would not have before by changing the root node.

Genome	Number
<a href="#">Homo sapiens 63 37</a>	(Human) 102
<a href="#">Pan troglodytes 63 21</a>	(Chimpanzee) 49
<a href="#">Gorilla gorilla 63 3</a>	(Western gorilla) 40
<a href="#">Pongo abelii 63 1</a>	(Bornean orangutan) 43
<a href="#">Macaca mulatta 63 10</a>	(Rhesus monkey) 46
<a href="#">Callithrix jacchus 63 321</a>	(White-tufted-ear marmoset) 61
<a href="#">Otolemur garnettii 63 1</a>	(Small-eared galago) 23
<a href="#">Microcebus murinus 63 1</a>	(Gray mouse lemur) 22
<a href="#">Tarsius syrichta 63 1</a>	(Philippine tarsier) 18
<a href="#">Rattus norvegicus 63 34</a>	(Norway rat) 48
<a href="#">Mus musculus 63 37</a>	(House mouse) 61
<a href="#">Spermophilus tridecemlineatus 63 1</a>	(Thirteen-lined ground squirrel) 16
<a href="#">Dipodomys ordii 63 1</a>	(Ord's kangaroo rat) 22
<a href="#">Cavia porcellus 63 3</a>	(Domestic guinea pig) 27

Figure 28: Current implementation

Source [5]

This is an implementation showing results about a combination being repeated in different genomes. The list of genomes goes on, this is part of the results. Yes viewing the data in this manner gives the information clearly but if you want to detect patterns and unusual changes it wouldn't be easy if the list were too long instead having a visualization of the data would improve that.

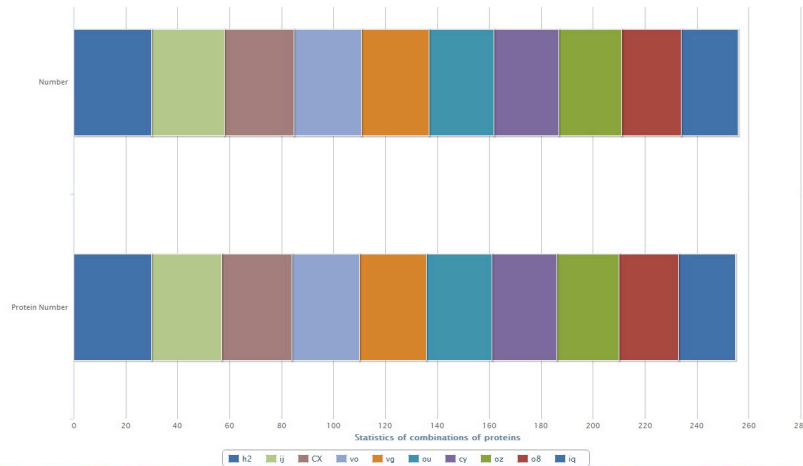


Figure 29: Result of improved visualization

The second graph focuses more how the data can be displayed in a more simple visualization yet the information is presented, as the user wants it. Again the amount of data is huge and hence making a simple visualization is difficult to keep both the information and a good visualization experience. Above image shows a stacked bar chart with a lot of information, as it is required to compare different genomes, hence to ensure that all information is retrieved and showed but still is easy to understand, many options are there for user to change the information as he desires.

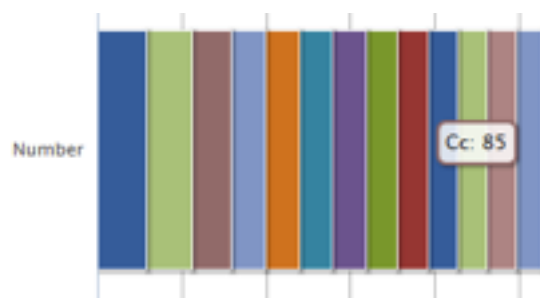


Figure 30: Feature of improved visualization

Above image shows that the user can roll over at any point in graph and it shows the exact value and genome that the value is for so that the information can be extracted easily even if the data is a lot.

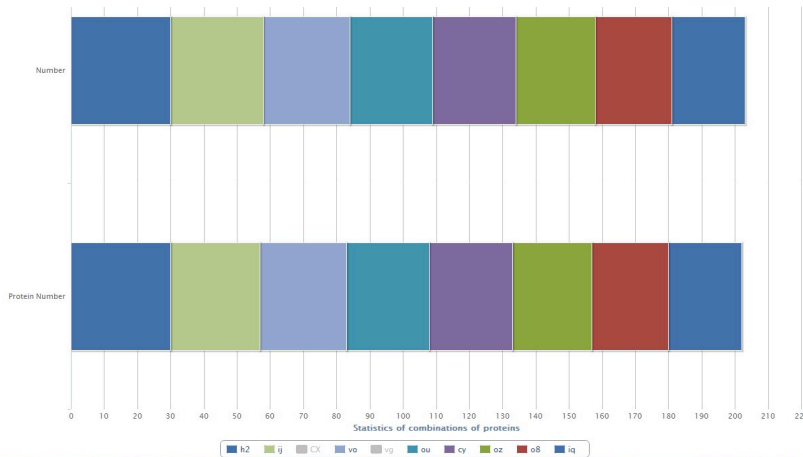


Figure 31: select/deselect genome in improved visualization

User also has the ability to select/deselect the genomes which he does not want, just by clicking at the genomes would make them grey and removing the data from the chart so the user can easily sort out the genome he finds are not required.

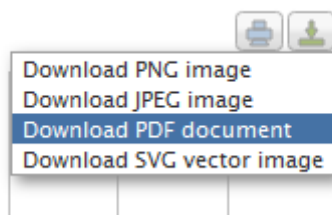


Figure 32: Feature of improved visualization

Sometimes you want the data and would like to store it. The graph has an option on top of it allowing the person to save the graph as a PNG, JPEG, SVG vector image or a PDF document. Enabling user to keep the data for further research. If required the second option allows user to print the graph directly from the web.

Graphs shown above represent the huge quantities of data, which is not only non-homogenous, it also scales in a non-linear way in places, and the relative importance is biological and not numerical. These graphs convey the information, which would be very hard for a researcher so see if just raw data was provided, as just looking at numbers would not help. This graphs tries to achieve the concept of 'A picture is worth a thousand words.' By showing a graph that represents such a data in a scientifically meaningful manner.

The visualizations implemented can be composed together so that the user can click on the genome in the tree visualization and be taken to the graph comparing them. This can be implemented just by making a new html page and requesting both the services that have already been implemented.



## Web Services

These were the results of the data visualization part of the system. Now we move on to the web services. Many functions are now available on the web allowing the user to get data in a standard format. Now let's consider an example, the tree graph that was discussed above showed some nodes with names on it or null. They all also have an id that can be used in a certain manner to manipulate the data. So here I show the request and response envelope of soap to the service if you request children of a certain node. In this case that node is 1. The method that we call from web service is `getChildrenByNodeId`.

### Request:

```
<soapenv:Envelope
  xmlns:q0="http://DefaultNamespace"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
  </soapenv:Header>
  <soapenv:Body>
    <q0:getChildrenByNodeId>
      <q0:NodeId>1</q0:NodeId>
    </q0:getChildrenByNodeId>
  </soapenv:Body>
</soapenv:Envelope>
```

**Listing 8:** SOAP request for child nodes

The request envelope sends the node for which you need the children to the database. In this case 1 is sent to the server for which in the response, children are returned.

### Response

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getChildrenByNodeIdResponse xmlns="http://DefaultNamespace">
      <getChildrenByNodeIdReturn>1684</getChildrenByNodeIdReturn>
      <getChildrenByNodeIdReturn>2</getChildrenByNodeIdReturn>
      <getChildrenByNodeIdReturn>2</getChildrenByNodeIdReturn>
      <getChildrenByNodeIdReturn>-1</getChildrenByNodeIdReturn>
    </getChildrenByNodeIdResponse>
  </soapenv:Body>
```

```
</soapenv:Envelope>
```

**Listing 9: SOAP response for child nodes**

Now as we remember all the nodes had names on them, to get the names you need the taxon id. So in the response you see nodes and taxon ids, for each node. 1684 is first node id, and 2 is the taxon id for that node, 2 is another node for which taxon id is -1.

The structure followed in the return from the function is

```
<node id>first node<node id>
<taxon id>taxon id for first node<taxon id>
<node id>second node<node id>
<taxon id>taxon id for second node<taxon id>
```

Now that we have the node id and the taxon id, we want the name of the node. For that we again have to call the web service in this case we request the name of node 1 which is 1684 where taxon id is 2. The method we use for this is getNameByTaxonId.

### Request

```
<soapenv:Envelope
  xmlns:q0="http://DefaultNamespace"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
</soapenv:Header>
  <soapenv:Body>
<q0:getNameByTaxonId>
  <q0:taxonId>2</q0:taxonId>
</q0:getNameByTaxonId>
</soapenv:Body>
</soapenv:Envelope>
```

**Listing 10: SOAP request for name by taxonID**

The request envelope shows that the call has been made with the taxon id 2 to the method getNameByTaxonId.

### Response

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<soapenv:Body>  
<getNameByTaxonIdResponse xmlns="http://DefaultNamespace">  
<getNameByTaxonIdReturn>bacteria</getNameByTaxonIdReturn>  
</getNameByTaxonIdResponse>  
</soapenv:Body>  
</soapenv:Envelope>
```

**Listing 11: SOAP response for name by taxonID**

The response envelope gets us the name of the node by the taxon id, here the name we got is 'bacteria' which is the name for node 1, whose id was 1684 and taxon id was 2.

This was a simple example that explained how a web service could be used for manipulating data. Now the advantages for researchers are a lot as they can get statistics of genomes, proteins, complete hierarchy and a lot more information in a standard form. So they can now easily use that information and implement their techniques on the data to process further. The method of screen scrapping would not be required once the web service is available.

## Critical Evaluation

This project is mostly based on data visualization and hence the best way to judge whether the visualization is better or not was to ask people. I carried out this on a small group of people, they used the application and they were asked to answer a couple of questions based on the visualization. They were first asked to view the data using the current visualizations and/or raw data, and then asked to view the same in the visualization developed in this project. They were to judge whether the data that is being displayed is meaningful, and shows the information they wish to see. They were asked to judge based on the information they see in raw data and/or in previous visualization.

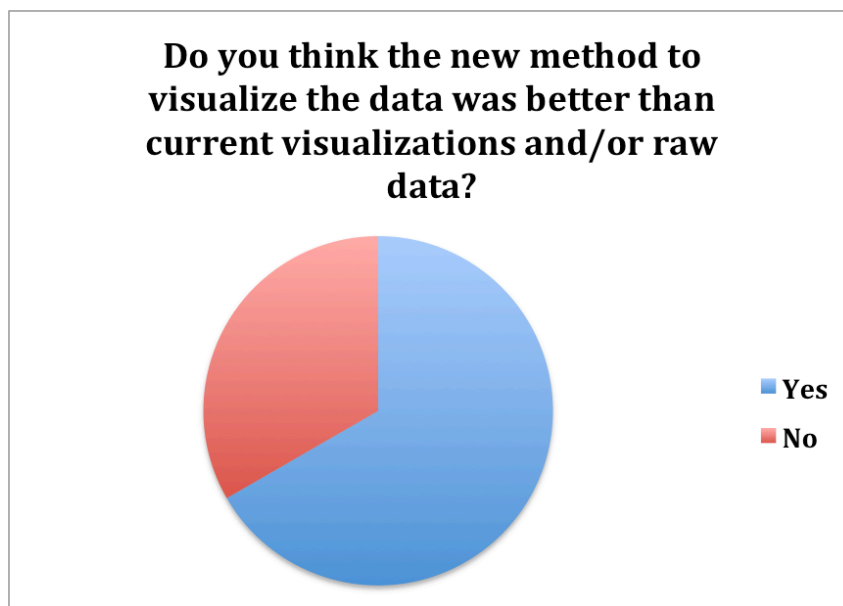


Figure 33: Evaluation Results 1

The result of the first question was satisfactory as most of the people thought the visualization was better than the viewing raw data or the current method of visualization. Only small portion of the people said that this was not better they were asked to comment on it. So I could get a better idea about why they did not prefer this visualization.

### If not please comment

Most of the comments on this were regarding the fact that they are fine with current visualizations and would not like them to change. Some comments show that they did not want any kind of change to visualizations. As some people are not open to changes this can be reason for this as well.

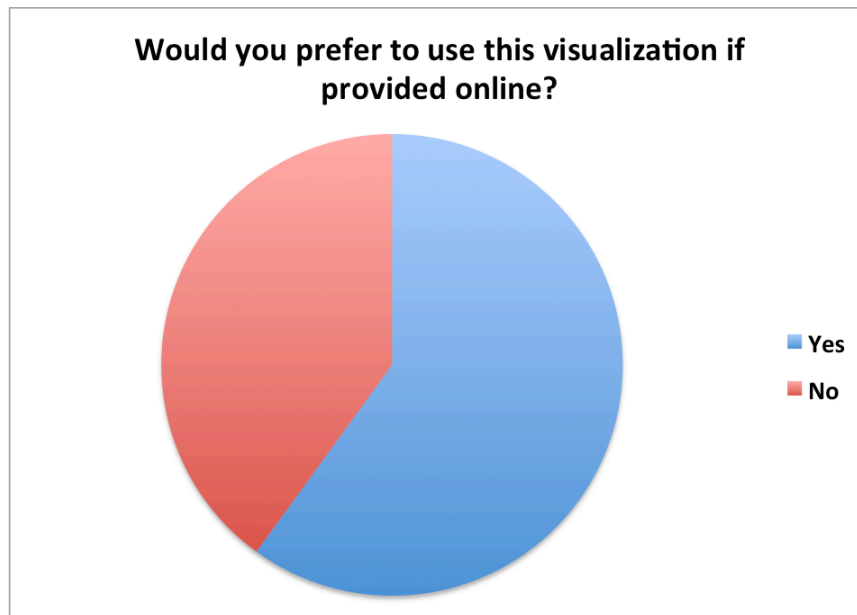


Figure 34: Evaluation results 2

Again the results were satisfactory as most of the people answered that they would use the visualization if provided online whereas a comparatively smaller portion said they would not prefer using this visualization if it was provided online.

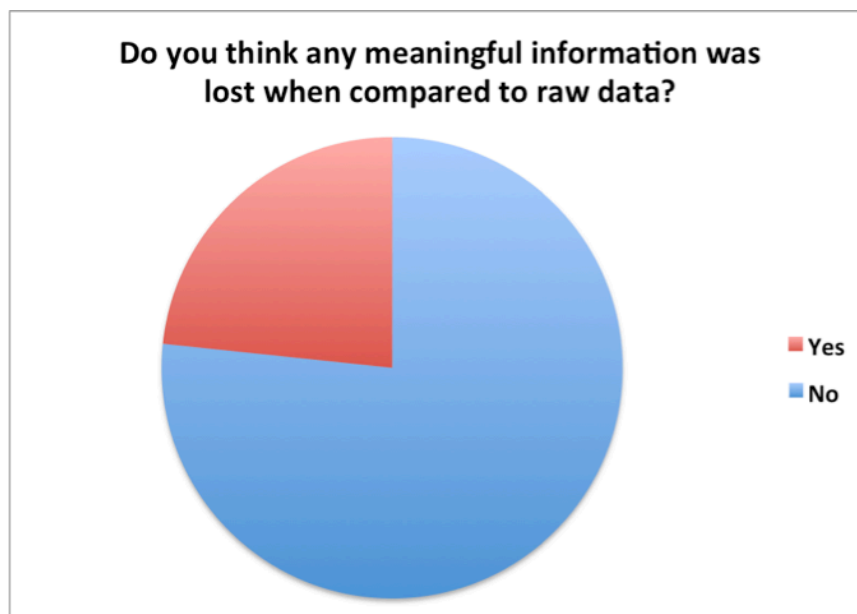


Figure 35: Evaluation Results 3

The most important question was to know whether the graph shows what needs to be shown. Students of computer science were satisfied with the information shown and felt nothing important was lost if compared to raw

data. Making this overall a better approach than currently available approaches.

After looking at all the results together, I can conclude that people do prefer this system, but still there is room for improvement as some answered they would not prefer to use this visualization that may be because they are satisfied with the current method. As the visualization is very different from currently implemented visualization, this can be a reason, as people generally do not like change. Implementing the system using the current visualization methods and introducing these features in those visualizations could result in a better system.

### **Problems that caused the limitation**

The visualizations developed in the project are limited but can be expanded using the technique defined above. Visualizations developed are limited because at first the requirements given were ambiguous; with a complex database that was so problem specific that it required the advisors domain knowledge for any interpretation. After getting to know the database and understanding what was required. I started with designing and developing the system. After the system was designed and ready to implement I was faced with more questions requiring domain knowledge of the advisor to understand the taxonomies and types of genomes that would be searched.

I now have the domain knowledge in the web service structure; now that the web services have been created the future implementation of the visualizations would be simplified using this system.

### **Conclusion**

The project started off with an objective at hand and an idea that needed to be implemented. The objective to visualize the data in a manner that is informative and yet easy to use for users and that the researchers who are willing to perform further work using the data in the SUPERFAMILY database are able to do so easily, both of these objectives were met.

Moreover, The objectives of visualization in bioinformatics were also met. As the system developed enables user to view large amount of information in a clear and effective manner and allows patterns and trends to be recognized, it even supports multiple visualization options to aid in pattern recognition.

The system developed can be improved as the evaluation results suggest that improvements can be made and hence there is scope for future work in this project.

## Future Work

Data visualization has a vast scope; it is not limited to the data considered in this project or even the technology. Data can be represented in a many ways. The way it has been represented in this project is meaningful for this type of data but there are many possibilities. Imagination is the limit to data visualization. If you can imagine the data meaningfully you can try to visualize the data in the same manner. It might not be possible now, due to limitation of the technology but it will be some day. [29]

The implementation in this project has been focused on the data in the superfamily database. Further improvements can be made, as from the evaluation we get a lot more insight on what could be a better system. As results indicate that some people would prefer old visualization rather than new. The system can maintain both the visualizations to rectify this problem or the old visualization can be improved to provide similar functionalities. Most of the people were satisfied with the fact that no information was lost in the visualization but to improve 3D based visualization could be used to provide more insight to the data.

## Glossary

Word	Definition
Superfamily Database	Database with the required information.
Biologist	A person looking to view the data on the webpage.
Researcher	A person who wants to get the data from the database and process it further.
Web Service	A service provided to give the researcher data in a standard format.
N-tier Architecture	A tiered approach to developing the system, keeping in mind that the system might be extended.
IV	Information Visualization
Rendering	Showing the data on the screen
Data Mapping	To match the data with visualization tools/shapes
Data acquisition	To get the data requested by the user
Bioinformatics	Combination of biology and computer science
DAO	Data Access Object, used to get information from the database.
BL	Business Logic, an important part of software architecture.
User	Biologist and/or Researcher



## References

1. Fu, Chen, Ryder, Barbara. G., Milanova, Ana, & Wonnacott, David (2004). 'Testing of java web services for robustness'. *Proceedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis - ISSTA '04*.
2. Lord, Philip, Bechhofer, Sean, Wilkinson, Mark D., Schiltz, Gary, Gessler, Damian, Hull, Duncan, Goble, Carole, Stein, Lincoln (2004) 'Applying Semantic Web Services to Bioinformatics: Experiences Gained, Lessons Learnt.' From ISWC 2004, *Bioinformatics*, 350-364.
3. Boehm, Boehm. W. (1988). 'A spiral model of software development and enhancement'. *IEEE Computer Journal*, volume 21, 61-72.
4. Londoninternational.ac.uk 'Software Engineering: Principles and Models', Available from [http://www.londoninternational.ac.uk/current\\_students/programme\\_resources/cis/pdfs/subject\\_guides/level\\_2/cis210\\_vol1/software\\_eng\\_chp3.pdf](http://www.londoninternational.ac.uk/current_students/programme_resources/cis/pdfs/subject_guides/level_2/cis210_vol1/software_eng_chp3.pdf) [Accessed 3rd July 2011]
5. Bhalla, Nishchal, & Kazerooni, Sahba (2007). 'Web Services Vulnerabilities' a White Paper by Security Compass
6. Gregory Murray and Jennifer Ball, Oracle.com 'Including Ajax Functionality in a Custom JavaServer Faces Component', Available from <http://www.oracle.com/technetwork/java/javaee/tutorial-jsp-140089.html> [Accessed 5<sup>th</sup> July 2011]
7. Lori MacVittie (2007) 'The Impact of AJAX on the Network'. A White Paper by F5 Networks, Inc.
8. Todd Anglin, 'The AJAX Papers Part III: Why, When, and What' by Telerik.
9. Brennan Spies, Ajaxonomy.com (2008) 'Web Services, Part 1: SOAP vs. REST', Available from <http://www.ajaxonomy.com/2008/xml/web-services-part-1-soap-vs-rest> [Accessed 1<sup>st</sup> July 2011]
10. Julian Gough, (2002) 'The SUPERFAMILY database in structural genomics' *Biological Crystallography*, ISSN 0907-4449
11. Supfam.org (2008) 'Superfamily', Available from <http://www.supfam.org> [Accessed 3<sup>rd</sup> July May 2011]
12. Nicolas Garcia Belmonte, thejit.org 'JavaScript InfoVis Toolkit', Available from <http://www.thejit.org> [Accessed 1<sup>st</sup> July 2011]

13. W3C, W3.org 'Simple Object Access Protocol (SOAP)', Available from <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/> [Accessed 15<sup>th</sup> August 2011]
14. W3C, W3.org 'Web Services Description Language (WSDL)', Available from <http://www.w3.org/TR/2001/NOTE-wsdl-20010315> [Accessed 15<sup>th</sup> August 2011]
15. Supfam.org (2008) 'SUPERFAMILY Web Services', Available from [http://supfam.org/SUPERFAMILY/web\\_services.html](http://supfam.org/SUPERFAMILY/web_services.html) [Accessed 19<sup>th</sup> August 2011]
16. Pressman R. S., (2000). 'Software Engineering - A Practitioner's Approach', 5th Edition, European Adaptation, McGraw Hill, 26-40.
17. Eclipse, eclipse.org (2011) 'Eclipse Downloads', Available from <http://eclipse.org/downloads/> [Accessed 15<sup>th</sup> August 2011]
18. JavaScript InfoVis Toolkit, simplecomplexity.net (2009) 'Interactive Data Visualizations for the Web', Available from <http://simplecomplexity.net/javascript-infovis-toolkit/> [Accessed 19<sup>th</sup> August 2011]
19. Raccoon, L. B. S. (1997). Fifty years of progress in software engineering. ACM SIGSOFT Software Engineering Notes, 22(1), 88-104.
20. Yu, Q., Liu, X., Bouguettaya, A., & Medjahed, B. (2006). Deploying and managing Web services: issues, solutions, and directions. The VLDB Journal, 17(3), 537-572.
21. Pautasso, C., & Leymann, F. (2008). RESTful Web Services vs . "Big" Web Services : Making the Right Architectural Decision Categories and Subject Descriptors. WWW '08, Proceeding of the 17th international conference on World Wide Web.
22. Protovis, Stanford.edu 'a graphical approach to visualization', Available from <http://vis.stanford.edu/protovis/> [Accessed 19<sup>th</sup> August 2011]
23. Miriah Meyer (March,2001), TEDx Waterloo, Available from <http://www.experiencefestival.com/wp/videos/tedxwaterloo--miriah-meyer--information-visualization-for-scientific-discovery/Sua0xDCf8MA> [Accessed 5<sup>th</sup> September 2011]
24. Ying Tao, Yang Liu, Carol Friedman, and Yves A. Lussier (2004). "Information Visualization Techniques in Bioinformatics during the Postgenomic Era". Drug Discov Today Biosilico. 2004 November; 2(6): 237-245.
25. Erin Cavanaugh. "Web services: Benefits, challenges, and a unique, visual development solution", Altova, Inc [Accessed 19<sup>th</sup> August 2011]
26. Joseph Miller , "Data Visualization Tool: Information as Fast as the Imagination". Available from <http://www.onlinesoftwareguide.com/0605datavisual.html> [Accessed 8<sup>th</sup> September 2011]

27. Stephen Few (2007), "Data Visualization Past, Present, And Future", Available from [http://www.perceptualedge.com/articles/Whitepapers/Data\\_Visualization.pdf](http://www.perceptualedge.com/articles/Whitepapers/Data_Visualization.pdf) [Accessed 9<sup>th</sup> September 2011]
28. Yen, G. G., & Wu, Z. (2008). Ranked centroid projection: a data visualization approach with self-organizing maps. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 19(2), 245-59. doi: 10.1109/TNN.2007.905858.
29. Lang U, et al. Visualization-related metadata. Data Issues for Data Visualization IEEE Visualization '95 Workshop; pp. 26–34.
30. All-about-psychology.com, Gestalt psychology Article: "Patterns" The New Psychology, Available from <http://www.all-about-psychology.com/gestalt-psychology.html> [Accessed 4<sup>th</sup> September 2011]
31. Andreas Prlić, Thomas A Down, Eugene Kulesha, Robert D Finn, Andreas Kähäri and Tim JP Hubbard (2007), Integrating sequence and structural biology with DAS, *BMC Bioinformatics* Available from <http://www.biomedcentral.com/1471-2105/8/333> [Accessed 7<sup>th</sup> September 2011]
32. Madera, M., Vogel, C., Kummerfeld, S. K., Chothia, C., & Gough, J. (2004). The SUPERFAMILY database in 2004: additions and improvements. *Nucleic acids research*, 32(Database issue), D235-9.
33. DePual University, Gestalt Principles, Available from [http://facweb.cs.depaul.edu/sgrais/gestalt\\_principles.htm](http://facweb.cs.depaul.edu/sgrais/gestalt_principles.htm) [Accessed 5<sup>th</sup> September 2011]
34. Lisa Graham (2008), Gestalt Theory in Interactive Media Design, *Journal of humanities and Social Sciences* Volume 2, Issue 1, 2008
35. GMOD, GBrowse, Available from <http://gmod.org/wiki/GBrowse> [Accessed 18<sup>th</sup> September 2011]
36. Hogeweg, P. (2011). The roots of bioinformatics in theoretical biology. *PLoS computational biology*, 7(3), e1002021. doi: 10.1371/journal.pcbi.1002021.
37. Cha, S.-jun, Hwang, Y.-young, Chang, Y.-seop, Kim, K.-ok, & Lee, K.-chul. (2007). The Performance Evaluations and Enhancements of GIS Web Services Variants for GIS Web Services. *Evaluation*.
38. Taverna, myGrid Project Available from <http://www.taverna.org.uk/> [Accessed 20<sup>th</sup> September 2011]

### Image Sources:

1. <http://supfam.cs.bris.ac.uk/SUPERFAMILY/cgi-bin/taxviz.cgi>
2. <http://www.pantherdb.org/chart/summary/pantherChart.jsp?filterLevel=1&chartType=1&listType=1&type=4&species=Homo%20sapiens>
3. <http://www.pantherdb.org/chart/summary/pantherChart.jsp?filterLevel=1&chartType=1&listType=1&type=5&section=PC00204&species=Homo%20sapiens&level2=PC00191&level1=PC00095>
4. [http://supfam.cs.bris.ac.uk/SUPERFAMILY/pics/database\\_schema\\_1\\_75.png](http://supfam.cs.bris.ac.uk/SUPERFAMILY/pics/database_schema_1_75.png)
5. [http://supfam.cs.bris.ac.uk/SUPERFAMILY/cgi-bin/allcombs.cgi?comb=54452,48726,\\_gap\\_](http://supfam.cs.bris.ac.uk/SUPERFAMILY/cgi-bin/allcombs.cgi?comb=54452,48726,_gap_)
6. <http://www.oracle.com/technetwork/java/javaee/tutorial-jsp-140089.html>
7. <http://www.experiencefestival.com/wp/videos/tedxwaterloo--miriah-meyer--information-visualization-for-scientific-discovery/Sua0xDCf8MA>
8. <http://www.thejit.org>
9. <http://vis.stanford.edu/protovis/>

## Appendix

### Functionality available in Web Service

The available functionalities within the web service are listed below.

#### **getPorteinBySeqId**

```
public int getPorteinBySeqId(java.lang.String seqId)
```

---

#### **getAssByProteinId**

```
public java.lang.String[] getAssByProteinId(int proteinId)
```

---

#### **getCombinationByTaxonomyAndProteinId**

```
public java.lang.String[] getCombinationByTaxonomyAndProteinId(int  
proteinId,  
java.lang.String taxonomy)
```

---

#### **getCombinationByProteinId**

```
public java.lang.String[] getCombinationByProteinId(int proteinId)
```

---

#### **getOrderByProteinId**

```
public java.lang.String[] getOrderByProteinId(java.lang.String taxonomy)
```

---

#### **getNameByTaxonId**

```
public java.lang.String getNameByTaxonId(int taxonId)
```

---

#### **getTaxonIdByLeftId**

```
public int getTaxonIdByLeftId(int leftId)
```

---

#### **getCombinationByComidandTaxonomy**

```
public java.lang.String[] getCombinationByComidandTaxonomy(int  
combId,  
java.lang.String taxonomy)
```

---

#### **getCombinationByCombId**

```
public java.lang.String[] getCombinationByCombId(int combId)
```

**getChildrenById**

public java.lang.Integer[] **getChildrenById**(int NodeId)

---

**getRightIdByLeftId**

public int **getRightIdByLeftId**(int NodeId)

---

**getTaxonIdById**

public int **getTaxonIdById**(int NodeId)

---

**getLeftIdByRightId**

public int **getLeftIdByRightId**(int NodeId)

These show the functionality available in the web service and the parameter they take as input and even the data type of the value returned.

## User Manual

This manual will allow researchers to connect to the web service, in this manual I will discuss how web service can be used through java. It can be used on other platforms as well but this manual will focus on using it with java.

1. First step is to create a java application
2. Import the Axis jar files available from Internet.
3. Now to call the web service, write the code below in your method.

```
try
{

String endpoint ="URL";

Service service = new Service();
Call call = (Call) service.createCall();

call.setTargetEndpointAddress( new java.net.URL(endpoint) );
call.setOperationName(new QName("http://soapinterop.org/", "Method
Name"));
String ret = (String) call.invoke( new Object[] { Paramter 1, Parameter 2 } );

System.out.println("Reply was '" + ret + "'");
}
catch (Exception e)
{
System.err.println(e.toString());
}
```

Replace URL with the URL of the service.

Replace Method Name with the name of the method you wish to call

Replace Parameter 1, Parameter 2 with the values that match the method being called.

Ret will hold the result.

This is a basic manual to enable the user to use the web service.