

Abstract	3
Introduction.....	3
The Client.....	3
Communication with our Client.....	5
First Steps	6
Context Drawing.....	6
Input and Output Detail	7
Normalisation	8
Entity Relationship Diagram	9
Designing the database.....	10
Databases Available.....	10
Database Decision	12
Database Rows and Columns.....	12
Bits and Bytes	13
Management of Images.....	14
SQL	14
Testing Server	15
Environmental Variables	16
WAMP Server.....	17
Tomcat	18
Using Command Prompt to Build mySql Database.....	18
phpMyAdmin	19
The first servlet.....	20
Java Virtual Machine	21
Text Editors	21
XML.....	22
Tomcat - testing the first servlet locally	23
Live Web Hosting Server Space	24
Communication with the Server Provider	24
DNS and Nameservers	25
Administration.....	26
FTP.....	27
Terrapin ftp	28
Uploading the first servlet.....	28
Building the Database Tables on the Server	29
Setting up the MySql Database.....	29
phpMyAdmin	29
Standardisation and Naming	30
Relational Database.....	31
Consolidating Where We Are	31
HTML for the Structure.....	32
Human and Computer Interaction	34
Sound and Voice Matching	36
Cascading Style Sheets for the Style	36
Fixed.....	37
Liquid.....	37
Elastic.....	37
JDBC	39
Outstanding Servlet Issue	41
Java Servlets.....	42
Two-tier servlet.....	42

Implementation	43
Introduction.....	43
ServletUserEnquiryForm1.java – design detail.....	43
Associated file: entry.html	43
Testing	44
Areas of concern	44
Tasks that were time-consuming.....	45
DisplayServlet.java – design detail.....	45
Associated files: admin.html, csEntryFormReport.html.....	45
Areas of concern	46
UpdatePageContent.java – design detail	48
Associated file: updatePagesForm.html.....	48
Testing	48
Areas of concern	48
Tasks that were time-consuming.....	49
JSP.....	50
MVC	50
Outstanding	50
Sessions (Authentication and Authorisation).....	50
Forthcoming Events	51
Prototype	51
Javascript	51
Java Applets.....	51
Soap and php	51
Tools Used	52
Photoshop	52
Evaluation.....	52
Conclusion.....	53
Abbreviations:.....	54
Reference:	54
Books	54
Web sites visited	55
Learning Sources	55

See Appendix (a) - Emails

See Appendix (b) - Questionnaire

See Appendix (c) - Minutes of Meeting

See Appendix (d) - Progress Report

Appendix (e) – Input/Output Schedule

Appendix (f) – Input/Output Schedule showing normalisation

Appendix (g) – print of the database tables

Code Appendix A – HTML files

Code Appendix B – CSS files

Code Appendix C – Java files

Abstract

Design and build of a website incorporating HTML, CSS, JSP, XML, Javascript, Java Servlets, JDBC, SQL, mySql database and SOAP, deploying the files to a web server creating a live website with some content management.

Introduction

This report is about Java Servlets, which are like little servants that run around on the web server getting things done. But to get a servlet to work requires a whole host of other things to be in place. Yes, a person can use Notepad, write a few lines of Java code that will be a servlet, but then, what do they do with it? Raw Java code is no good to anyone other than someone learning how to write Java. It needs to be compiled, which means written into a code that a computer machine will understand. The next step is to get all the things needed to compile it. But then, what to do with it, once it's compiled? To use it on a website needs some web server space, and also a database to hold all the information. The server space is purchased, and the database built and the code compiled. It doesn't work! What then? This report details achieving getting it all to work as well as other items related to the servlets working, being the project itself and the website implementation.

The Client

Yes, there is a real client, the Durrington Festival Committee, a Registered Charity, operated by local residents and people from the education, business and religious communities. Figure (i) shows their structure.

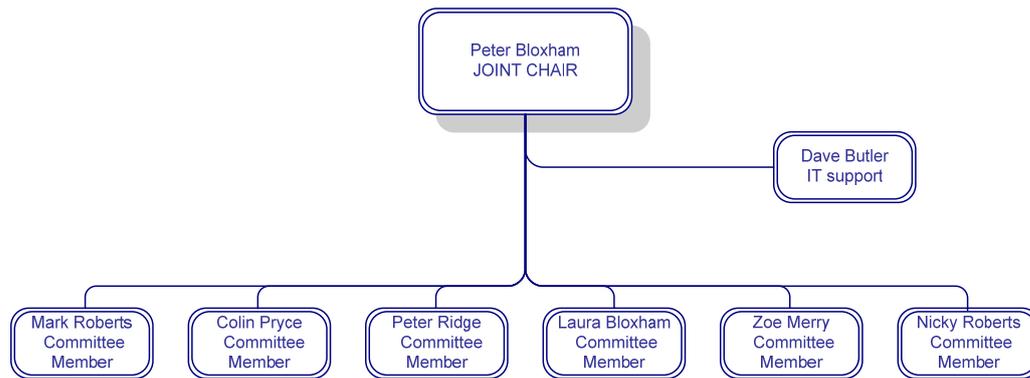
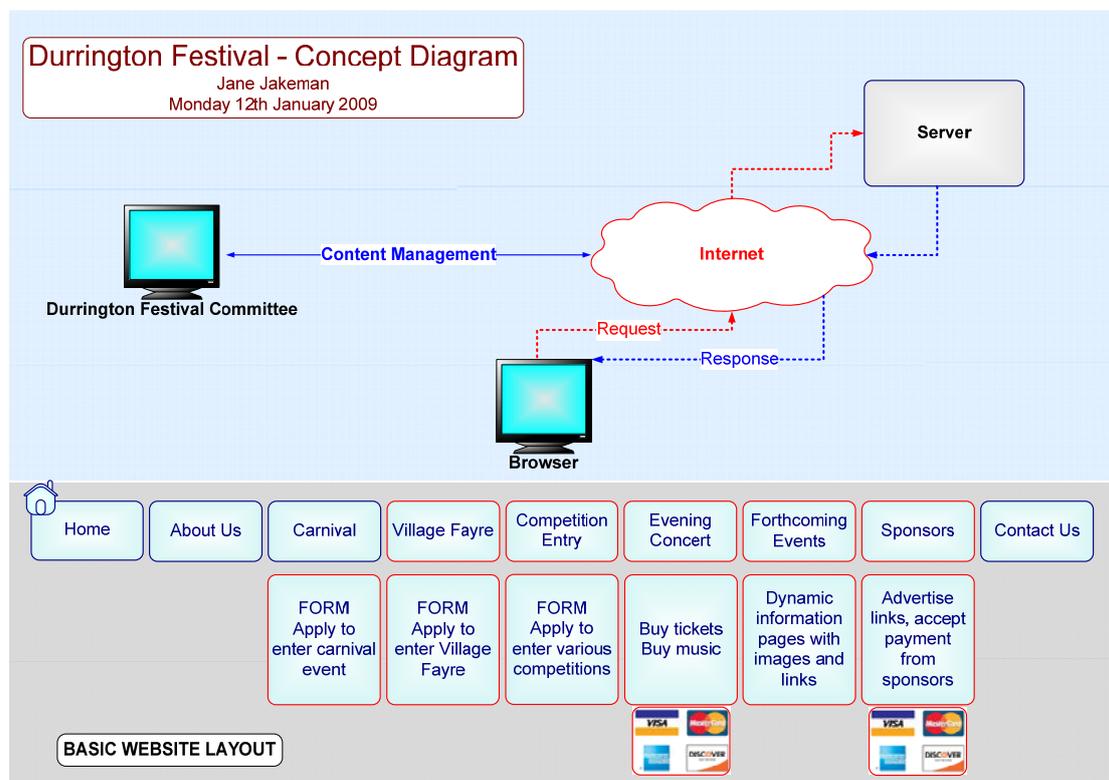


Figure (i) – Durrington Festival Committee Structure

The Committee organise a large-scale Festival in West Sussex each year and require a website to support this event. A concept diagram is shown below in Figure (ii)



Figure(ii) – Concept drawing of the proposed website

The main thing to see with this drawing is that the Committee may update the website from their computer, anywhere in the world. A user, for example, a member of the public may browse the site and apply online to enter various elements of the Festival.

Apart from displaying text and image detail on their website, their requirements are:-

- online forms
- some content management facility
- a counter of visitors

The Committee also have a wish list of items including:

- Photo Gallery
- Choice of music to listen to whilst browsing the site
- Start of a video collection from June 2009 to be displayed on the site
- A feedback page, where local residents can upload comments.
- A Message from the Chair, which could be updated on a regular basis, or as a video.

Apart from the above requirements, it is an open brief.

It is a big leap from having a virtually static site, to a complete system design that the Committee can update themselves as a website with total content management functionality. Committee are likely to benefit by having some content management facility; after becoming familiar with that facility more advanced functionality could be introduced.

The Committee decided later on in the project that they did not want a payment facility on their site for reasons of security and management of financial transactions and are happy with their current system that works for them.

Communication with our Client

This was by phone, email, questionnaire and a meeting, providing a copy of the Project Progress Report and receiving copy of the Committee's Annual Report.

See Appendix (a) - Emails

See Appendix (b) - Questionnaire

See Appendix (c) - Minutes of Meeting

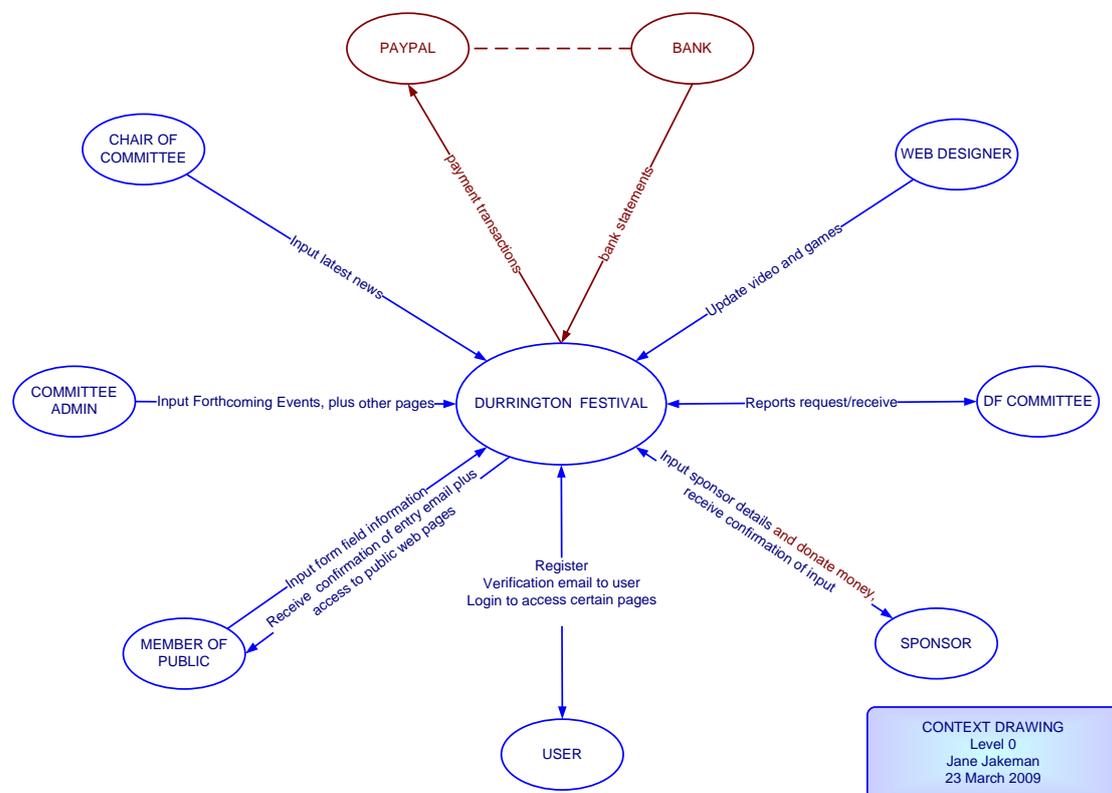
See Appendix (d) - Progress Report

First Steps

Prior to this report, first a Project Proposal Report was written and then in January 2009 a Progress Report were written as attached in Appendix (d).

Context Drawing

The first thing to do was to create a Context Drawing, Level 0, as below:



Figure(iii) – Context Drawing – Level 0 - proposed website

This shows in the briefest form, the input and output planned, what the input should or could be and what output there could be. It also shows who would be inputting information and who would be receiving information. Shown in a different colour is the banking element, which is not being designed in at this time. The SPONSOR details will now be input by the Committee Administrator.

At some stage in the future the Committee could request the site be used for fundraising, attracting online donations from members of the public, business, and payments for goods and services and the design is intended to accommodate for growth in the use of the website.

However, the main point of this drawing is that it helps to establish what is required. The next step is to create an Input/Output Schedule.

Here is a screenshot of part of the Input/Output designed for the website. A full print of the Input/Output Schedule is attached (*See Appendix (e)*).

Input and Output Detail

INPUT						OUTPUT	
By who ...	Page Title	Detail	Data Type	Other functions	Category	Detail	Who for
Peter Bloxham	Latest News	Date	Date	Search by:	Text	Date	Member of public
		News Title	VARCHAR	Order by:	Date	News Title	
		News Item	TEXT	Admin Panel		News Item	
		Image	MEDIUMBLOB	Authentication		Image	
		Author	VARCHAR			Author	
Mark Roberts	Forthcoming Events	Event Title	VARCHAR	Search by:	Date	Event Title	Member of public
		Date	DATE	Admin Panel	Category	Date	
		Time	TIME	Authentication		Time	
		Venue	VARCHAR			Venue	
		Image	MEDIUMBLOB			Image	
		Other information	TEXT			Other information	
		Link	VARCHAR			Link	
Mark Roberts	Photo Gallery	Photo	MEDIUMBLOB	Search by:	Category	Image	Member of public
		Photo text	TEXT	Order by:	Date	Image text	
		Category	VARCHAR	Admin Panel		Category	
		Date photo taken	DATE	Authentication		Date image taken	
Member of Public	Entry Form	Category	VARCHAR			Automated confirmation email	Member of public
		Person type	VARCHAR			Automated email to DF email	Mark Roberts
		First name	VARCHAR				
		Second name	VARCHAR				
		House Number	VARCHAR				
		Streetname	VARCHAR				
		Town	VARCHAR				
		Postcode	VARCHAR				
		Email address	VARCHAR				
		Number of people	INT				
		Vehicle type	VARCHAR				
		Phone number	VARCHAR				
		Previous participant	BOOL				
Sponsor	Sponsor	Name	VARCHAR			Output on Sponsors Web Page	Member of public
		Organisation name	VARCHAR			Organisation Name	
		House Number	VARCHAR			Logo	

(Figure(iv) – Input/Output Details – see Appendix(e) for full print of Input/Output Details)

By categorising the data inputs and data outputs and identifying who inputs which data and what sort of information it is, and who wants what out, helps to

clarify the database requirement in a logical manner. This Input/Output table determines the Attributes and Entities needed for the database design.

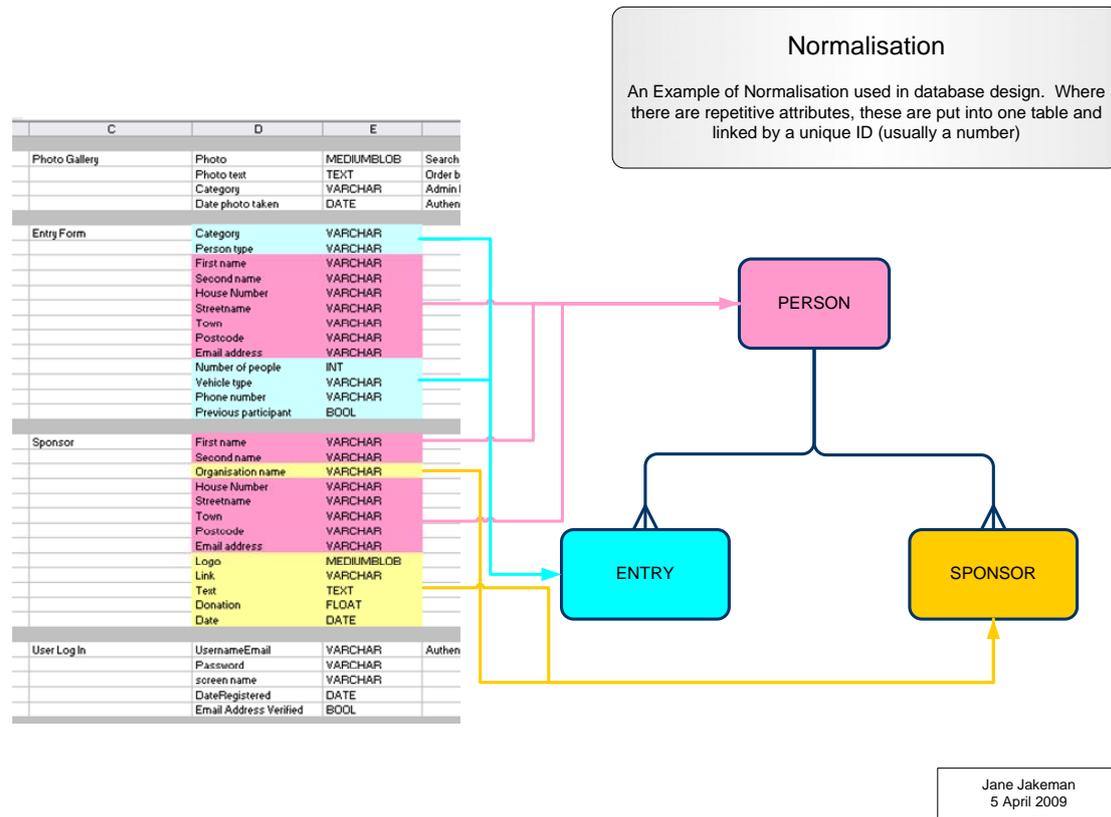
Once this is detailed, normalisation may begin.

Normalisation

Normalisation is about creating an efficient data store. The main aim is to try not to have the same element more than once in the whole database. For example, if there is a competition entrant with a 'Surname' and there is a sponsor with a 'Surname', instead of creating a table for competition entrant, and another for sponsor, one table would be created and named say, Person. In the Person table, there would be a field for 'Surname' and a unique Id number attached to that 'Surname' field. Then in the Sponsors table, creating a new sponsor would create a unique Sponsor Id, and then the Person Id from the Person table could be placed in the Sponsor table. This means at any time, if the Person details are required for the sponsor, they can be retrieved by linking the Person id in the Person table to the Person id in the Sponsors table to retrieve the Surname. This is the main structure of the relational database.

Normalisation is about making sure that inefficiencies do not occur in your database.

E. F. Codd designed the relational database and also established normalisation. There are a series of forms of normalisation, ranging from one to five, with one being the most strict. The form of normalisation generally used is Third Normal Form, and it is this Third Normal form that is intended for this design. Next, the normalisation is applied to the Input/Output details. Figure (v) below is taken from our Input/Output details. Normalisation has been applied and coloured so it is clear to see. *See Appendix (f)*



Figure(v) – Normalisation

Entity Relationship Diagram

This then leads to an Entity Relationship Diagram which is shown below:

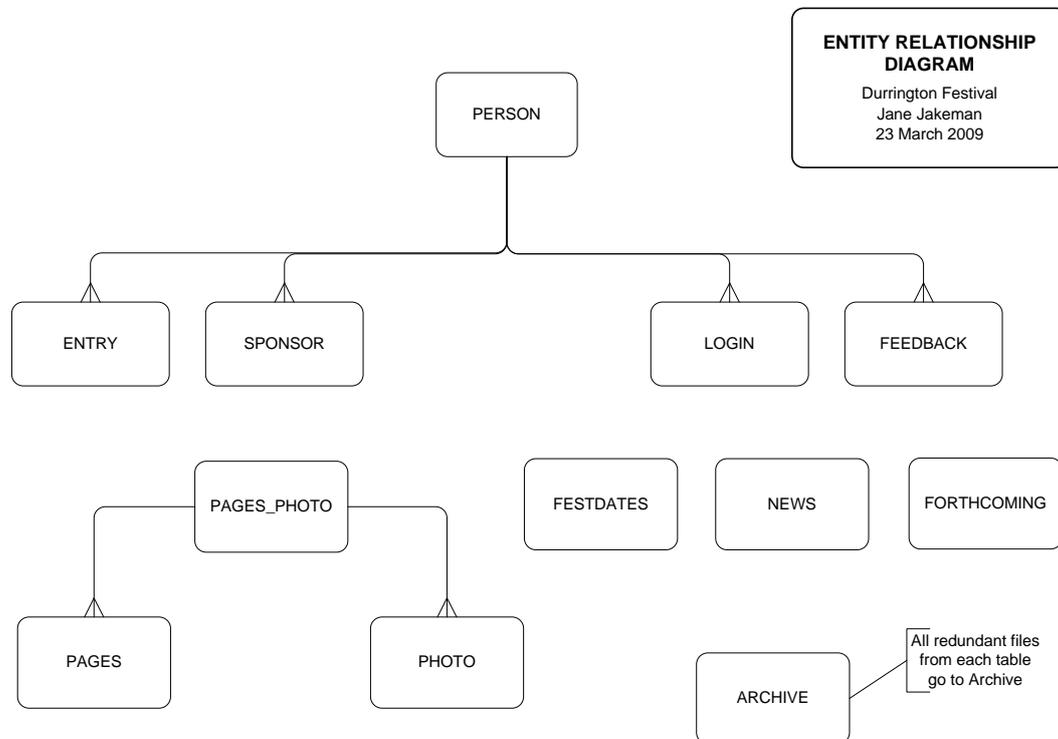


Figure (vi) Entity Relationship Diagram

This is the method used to design the database. Above it shows that the Person Entity is linked by a one-to-many relationship to the following entities:

- Entry
- Sponsor
- Login
- Feedback

For example:

- one person can enter many times
- one person can sponsor many times
- one person can login in many times
- one person may add feedback many times

With Pages and Photo, there are many pages and many photos. That would create a muddle in the database, so it is separated with a Pages_Photo entity that will act as a go-between.

The other entities sit separately in the database.

Designing the database

The next thing to do was to design and build the database and all the database tables with the database table structure that was defined. But first, a decision is needed over which brand of database to use.

Databases Available

Here are three of the most well-known and most popular databases:-

- mySql Database
- Oracle Database
- Microsoft Access Database

Each of these databases recognises SQL statements.

Previous learning about Microsoft Access showed that it supports relatively small systems, for example, up to approximately twelve users comfortably. The language behind Microsoft Access is Visual Basic. It provides wizards to make designing a database quite simple for a user, is generally considered suitable for small applications.

To use a database, there needs to be a connection to it. Our specification is to use Java Database Connectivity which is known as JDBC. Microsoft Access Database has its own 'database connector' ODBC and there is a bridge available from Java, however Sun Microsystems, the owners of Java themselves recommend that this is not the best option to use. Therefore this is a negative point in the decision-making process, reducing further Microsoft Access as first preferred choice in this instance.

Oracle Database has a good reputation, is possibly one the fastest database for web services and within recent years it has become more freely available by the pricing structure changing and becoming more open source. Their Oracle 10g Express Edition database is free, however, Oracle sell add-ons to this and a decision was made not to use Oracle at this time, based on the possibility of unforeseen costs occurring.

The third database considered is MySQL. This is an open source database that is widely used on the internet and has proved itself to be reliable and capable of handling web data transactions. It is continually being improved with each new version. MySQL is Open Source software which means there is no licence fee to pay. There may be a liability to pay a licence fee if it is used for business. As it is being used for a not-for-profit Registered Charity this caveat does not apply in this instance.

MySQL database usually works hand-in-hand with php which is a scripting language that may be inserted to HTML pages and is used for web development. Incidentally, php also provide an inbuilt database called SQLite, and possibly as each version of SQLite is improved upon, this may prove as popular as MySQL database.

Database Decision

The decision made is to use mySql, based on its proven track record of supporting thousands and possibly millions of websites, its reputation and popularity among developers and its costs, of which there are none.

Database Rows and Columns

Databases work by rows and columns (*see figure (vii)*), so for example, a person has an Id number, then the next item along the row is another piece of information, ie the person's name, then the next example is their address, and so on.

	Column 1	Column 2	Column 3
	Person_Id	Person_Name	Person_Address
Row 1	1	Jane	Smith Street
Row 2	2	Bob	Harrow Road
Row 3	3	Jim	Greenford Gardens

Figure (vii) – Database rows and columns

In Column 1, which is Person_Id in this instance, the database would probably be designed so that this column:

- assigns a Primary Key
- makes its data type an Integer
- sets the Integer to auto_increment every time a record is added

In the example above, the data type selected was an Integer, other data types for mySql database are contained in from the following list sourced from a website (*see source Reference 1-List of MySql data types*):-

Data Type	Storage Required
CHAR(<i>M</i>)	<i>M</i> × <i>w</i> bytes, 0 ≤ <i>M</i> ≤ 255, where <i>w</i> is the number of bytes required for the maximum-length character in the character set

BINARY (<i>M</i>)	<i>M</i> bytes, $0 \leq M \leq 255$
VARCHAR (<i>M</i>), VARBINARY (<i>M</i>)	<i>L</i> + 1 bytes if column values require 0 – 255 bytes, <i>L</i> + 2 bytes if values may require more than 255 bytes
TINYBLOB , TINYTEXT	<i>L</i> + 1 bytes, where $L < 2^8$
BLOB , TEXT	<i>L</i> + 2 bytes, where $L < 2^{16}$
MEDIUMBLOB , MEDIUMTEXT	<i>L</i> + 3 bytes, where $L < 2^{24}$
LOBLOB , LONGTEXT	<i>L</i> + 4 bytes, where $L < 2^{32}$
ENUM ('value1', 'value2', ...)	1 or 2 bytes, depending on the number of enumeration values (65,535 values maximum)
SET ('value1', 'value2', ...)	1, 2, 3, 4, or 8 bytes, depending on the number of set members (64 members maximum)

Figure viii – List of mySql data values

Variable-length string types are stored using a length prefix plus data. The length prefix requires from one to four bytes depending on the data type, and the value of the prefix is *L* (the byte length of the string). For example, storage for a **MEDIUMTEXT** value requires *L* bytes to store the value plus three bytes to store the length of the value. (Reference 1).

Source: <http://dev.mysql.com/doc/refman/5.0/en/storage-requirements.html>

Bits and Bytes

Here is an example of ASCII character encoding and bits and bytes. ASCII stands for American Standard Code for Information Interchange. As computers only work in 1s and 0s, the ASCII code is a numerical representation of the alphabet and other special characters such as '/' or '!'.

A bit is a 1 or a 0, a byte is a group of 8 bits. Working from right to left, the value increases in an 8 bit set (1Byte) so that if there was a '1' in each of the 8 values from 1 to 128, the total would be 255, or 256 if 0 is counted. Therefore, 1 Byte can have 256 different values. It works on base 2 mathematics for example $2^3 = 8$. However, if you look at the position of 8, it is the 4th value from the left. But if you count 0,1,2,3 then it is correct. So working out the bit position is different to working out the value of the bit.

Here is my name, in ASCII encoding.

Letter	ASCII No	128	64	32	16	8	4	2	1
J	74	0	1	0	0	1	0	1	0
a	97	0	1	1	0	0	0	0	1
n	110	0	1	1	0	1	1	1	0
e	101	0	1	1	0	0	1	0	1

Management of Images

As the database is to hold images, it is essential to manage those file sizes so that the size of the store of bytes in the database is not detrimental to the performance of the database.

Most photographs or images when taken with an ordinary digital camera measure approximately 1 or 2 Megabytes in size. There are 1024 bytes in a Kilobyte, and there are 1024 Kilobytes in a Megabyte. 1MB equates to 1,048,576 bytes.

Byte size can be reduced using compression, the ideal compression is JPEG which will create file sizes of around 100 KB per image without losing too much quality and is sufficient for the website being developed.

SQL

There is a complete difference between SQL and mySql and sometimes people mistakenly consider them to be the same.

SQL stands for Structured Query Language. It is a way of writing a statement that most databases will understand. A typical simple SQL statement would be:-

```
SELECT * FROM TABLENAME
```

The asterisk (*) means ALL, therefore this statement will provide all the rows of data in the table called TABLENAME.

This is how SQL would be written to create a table called TBLFORTHCOMING in the database called compnqmt_DURRFEST.

```
CREATE TABLE `compnqmt_DURRFEST`.`TBLFORTHCOMING` (  
  `EventId` INT( 11 ) NOT NULL ,  
  `EventTitle` VARCHAR( 255 ) NOT NULL ,  
  `EventDate` DATE NOT NULL ,  
  `EventTime` TIME NOT NULL ,
```

```
`EventVenue` VARCHAR( 255 ) NOT NULL ,  
`EventInfo` TEXT NULL ,  
`EventImage` MEDIUMBLOB NULL ,  
`EventLink` VARCHAR( 255 ) NULL ,  
PRIMARY KEY ( `EventId` )  
)
```

To join the data between two tables to become relational, there needs to be a similar value in each table, usually on an Id number. Listed below is detail of how each join works.

- **JOIN:** Return rows when there is at least one match in both tables 1
- **LEFT JOIN:** Return all rows from the left table, even if there are no matches in the right table
- **RIGHT JOIN:** Return all rows from the right table, even if there are no matches in the left table
- **FULL JOIN:** Return rows when there is a match in one of the tables

(Reference 2)

If the join is not determined correctly, then the data returned could be incomplete.

We could create a table ID so that when we want to select from any table we may write the statement.

```
SELECT * FROM table1, table2;  
  
SELECT * FROM table1 INNER JOIN table2 ON table1.id=table2.id;  
  
SELECT * FROM table1 LEFT JOIN table2 ON table1.id=table2.id;  
  
SELECT * FROM table1 LEFT JOIN table2 USING (id);  
  
SELECT * FROM table1 LEFT JOIN table2 ON table1.id=table2.id  
LEFT JOIN table3 ON table2.id=table3.id;
```

(Reference 3)

Testing Server

A testing server is needed to test files that are being developed. Both Tomcat and WAMP are open source software that is freely downloadable and may be installed to a computer. Tomcat is typically used by developers for JSP and

Java Servlets and mySql and WAMP is typically used by developers using php and mySql.

WAMP may be downloaded from:

<http://www.wampserver.com/en/download.php>

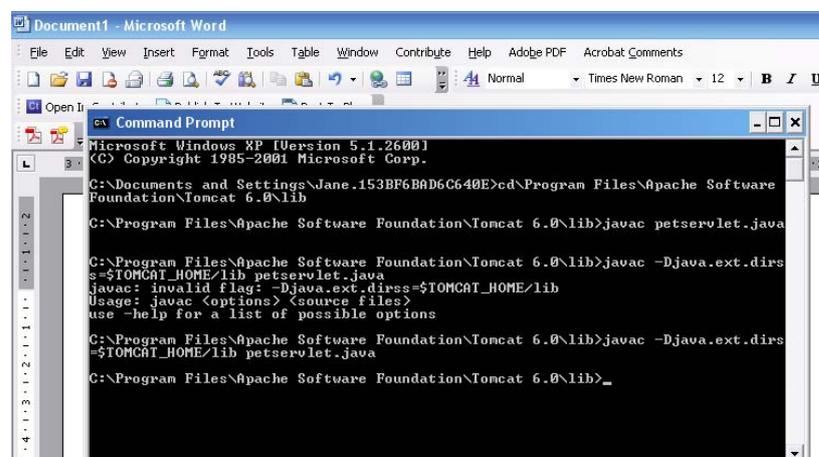
Tomcat may be downloaded from: <http://tomcat.apache.org>

Setting up a testing server involves organising a complete development and deployment environment on both your local pc and the server directories.

Environmental Variables

For using Java and in particular Java Servlets, the Java Development Kit (JDK) needs to be downloaded; also required is the Java Runtime Environment (JRE). Then classpaths need to be set in the Environmental Variables within the System Properties of the pc as well as the classpaths for Tomcat.

Below in figure (vii) is a screenshot showing trying to compile petservlet.java. It did not compile correctly because one of the environmental variables was set incorrectly. The issue was the JAVA_HOME Environmental Variable was not set properly



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Jane.153BF6BA06C640E>cd\Program Files\Apache Software
Foundation\Tomcat 6.0\lib
C:\Program Files\Apache Software Foundation\Tomcat 6.0\lib>javac petservlet.java

C:\Program Files\Apache Software Foundation\Tomcat 6.0\lib>javac -Djava.ext.dirs
s=$TOMCAT_HOME/lib petservlet.java
javac: invalid flag: -Djava.ext.dirs=$TOMCAT_HOME/lib
Usage: javac <options> <source files>
use -help for a list of possible options

C:\Program Files\Apache Software Foundation\Tomcat 6.0\lib>javac -Djava.ext.dirs
-$TOMCAT_HOME/lib petservlet.java

C:\Program Files\Apache Software Foundation\Tomcat 6.0\lib>_
```

For example, the environmental variables set up for this task are:

ANT_HOME C:\Ant\apache-ant-1.7.1

```
CATALINA_HOME C:\Program Files\Apache Software Foundation\Tomcat
6.0
JAVA_HOME C:\Program Files\Java\jre6 (this was set to jdk before)
PATH C:\Program Files\Java\jdk1.6.0_12\bin;C:\Ant\apache-ant-1.7.1\bin
TEMP %USERPROFILE%\LocalSettings\Temp
TMP %USERPROFILE%\LocalSettings\Temp
TOMCAT_HOME C:\Program Files\Apache Software Foundation\Tomcat 6.0
```

Code for compiling the servlet in the command prompt

```
JAVAC -Djava.ext.dirs=$TOMCAT_HOME/lib petservlet.java
```

When the command prompt returns the file path, as it has in the screenshot above, then it means that the file has compiled correctly, and there will be a new file created, a class file and that will be in the directory as designated in the command line, in this instance, it is in c://Program Files/Apache Software Foundation/Tomcat 6.0/lib/petservlet.class.

Therefore \$TOMCAT_HOME in the command prompt points to c://Program Files/Apache Software Foundation/Tomcat 6.0

Therefore the line of code means:

- JAVAC
 - call for a compile
- Djava.ext.dirs=\$TOMCAT_HOME/lib
 - means direct the file to an external directory which is c://Program Files/Apache Software Foundation/Tomcat 6.0/lib
- petservlet.java
 - is the file to compile

This will create petservlet.class in c://Program Files/Apache Software Foundation/Tomcat 6.0/lib

WAMP Server

Having already a WAMP server installed on this pc and set to operate local host at Port 80, to display a file in the browser, first of all:

- Save myFile .php in the www directory of WAMP file system

Then, start the server if it's not already started, or stop and start it if it's already running:

- this is done by selecting the server>Start All Services
- type into the browser address bar: <http://localhost> will result in the browser displaying the home page of the server
- <http://localhost/myFile.php> will result in the file being displayed, unless:

- there are any errors in the file, then the errors will be detailed on the browser

Tomcat

Installing Tomcat is similar, however its default Port is 8080, therefore, to display the similar php file, first you need to save the file in:

- Tomcat Directory/webapps

You then need to start Tomcat, for example:

- start from the command prompt
- access the Tomcat/bin directory and find the startup file in there and operate the start from there
- type into the browser address bar <http://localhost:8080/myFile.php>

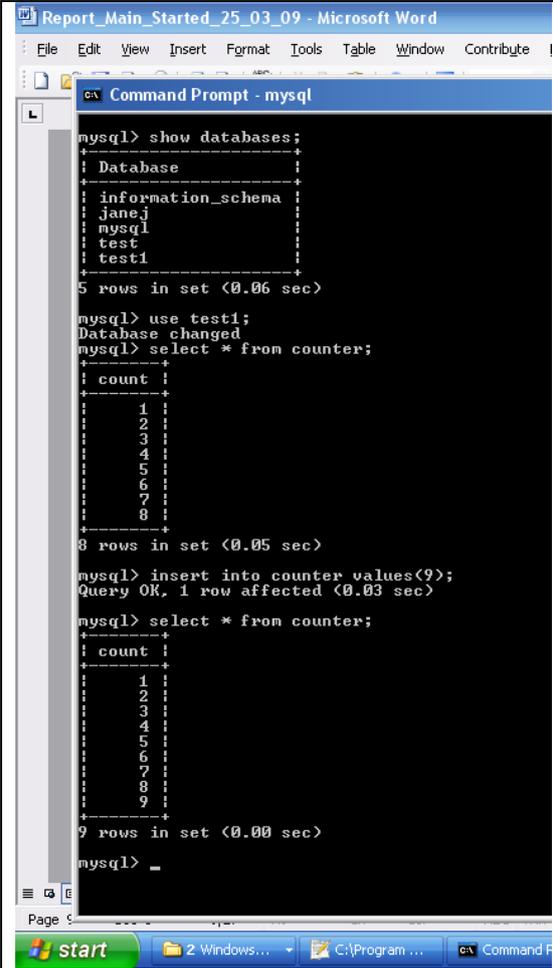
The reason that the port is not needed in the WAMP folder is that the default Port number for displaying webpages is 80 and therefore as WAMP Port number is set to 80 by default, unless it has been changed, it does not need to be entered into the address bar for that reason. When installing Tomcat, there is an option to change the Port number if required.

Using Command Prompt to Build mySql Database

Using the command prompt to build a mySql is straightforward and basic, with no frills and creates a sense of technical ability. Once the SQL syntax and the MySql terminology is learned, it is responsive and quick.

The command prompt may be accessed from the Windows Start/All Programs/Accessories/Command Prompt, or may be accessed by Start/run and then type cmd into the message box. Once the command prompt appears on the screen, type mysql and this will put a user in the mySql environment. From this point a user will be able to create databases, add, update, delete, view, drop tables plus many more commands.

Here is a screenshot of database test1 and table Counter being accessed. Table Counter has just one attribute (or column), Count.



The screenshot shows a Windows Command Prompt window titled 'Command Prompt - mysql'. The user has executed the following commands and received the following output:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| janej     |
| mysql    |
| test     |
| test1    |
+-----+
5 rows in set (0.06 sec)

mysql> use test1;
Database changed
mysql> select * from counter;
+-----+
| count |
+-----+
| 1     |
| 2     |
| 3     |
| 4     |
| 5     |
| 6     |
| 7     |
| 8     |
+-----+
8 rows in set (0.05 sec)

mysql> insert into counter values(9);
Query OK, 1 row affected (0.03 sec)

mysql> select * from counter;
+-----+
| count |
+-----+
| 1     |
| 2     |
| 3     |
| 4     |
| 5     |
| 6     |
| 7     |
| 8     |
| 9     |
+-----+
9 rows in set (0.00 sec)

mysql> _
```

One example here is:

- show databases;
- use test1;
- select * from counter; (1-8)
- insert into test1 values(9);
- select all from counter; (1-9)

The items shown in red are the results of the query, SELECT * FROM COUNTER which is a table in database test1;

Figure (ix) – Screenshot of mysql in the command prompt

However, although using mySql in the command prompt is easy to do, there is another tool to use, phpMyAdmin.

phpMyAdmin

As this site will be live, the plan is to develop and implement the database live on the web hosting server.

An alternative to using the command prompt to carry out MySQL database tasks is to use phpMyAdmin. Mysql could have been written in the command prompt and packaged and sent to the web host. However, it is far more time efficient to build directly on the web host space and this is the reason why phpMyAdmin was developed, for that purpose. It is an interface that is

provided with a server that supports mySql database. Therefore, phpMyAdmin is the interface, mySql is the database, and SQL is the language used to access the database. The procedure is start building the database is to access the internet, and access the website control panel, in this instance the address in the address bar is <http://www.completewebsites.biz/cpanel> which is the live server space being used for live testing. Then, when authorisation is passed, in cpanel:-

- access mySql database area
 - create a database
 - create a user for that database
 - create a password for that database
 - set the privileges for that user

- access phpMyAdmin
 - create the tables for the database

However, before we do that, we need a test some files locally using a testing server, Tomcat.

The first servlet

The first servlet can now be written. This comprises of:

- an HTML file written in Notepad
- a Java file written in Notepad
- an XML file to provide the mapping of the servlet to the server

The HTML form has a method 'post' and this was set to the path of the servlet, as the testing server will understand this path definition with the aid of an xml file.

The Java file was then compiled by saving it in the Tomcat/lib directory, then opening the command prompt and navigating to the lib directory, and then typing: `javac myFile.java`. This is another way of compiling a file. This then created a `myFile.class` in the Tomcat/lib directory. This class file was cut and

pasted into a new folder under webapps, the folder must be named classes. This is because the server knows when a servlet has been asked for, and looks for it in the classes folder. The web.xml file directs the server to the servlet.

Java Virtual Machine

The Java Virtual machine links to the operating system of a computer and runs on a separate part of the disc or memory, ie a little operating system all of its own. Any pc that has a JVM installed, which is freely available, may display Java compiled files. This is how Java is platform independent. Most computers throughout the world have the JVM installed on them.

Text Editors

Notepad is a text editor. When a person writes text into the 'page' in Notepad, saves the file as myDocument.txt it generally saves at 1 byte per character typed. This is quite a low bit rate, because Notepad is a little like a word processing package such as Microsoft Word, but without all the extras that Microsoft Word has.

During this project, all files apart from the database files have been written in Notepad++ and saved as their file type, for example .html, .css, .java, .jsp.

Notepad++ is a luxury version of Notepad, as it provides colours for the text and is a great help for identifying errors immediately they are made as the colour flow of the 'script' changes if it's not correct and this has been used. However it is the most basic tool.

There are many other text editors with more functionality than Notepad including Eclipse and Ant. These have not been used for this project although Ant was tested.

Netbeans IDE

This is an interface that does a lot of the work for the developer and has not been used at all as this would detract from the learning from using basic tools.

XML

The web.xml file was created in Notepad which creates the mapping, typical example follows:

```

18
19 <web-app xmlns="http://java.sun.com/xml/ns/javaee"
20     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
21     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
22     version="2.5">
23
24 <servlet>
25 <servlet-name>myServletMadeUpName</servlet-name>
26 <servlet-class>MyActualClassName</servlet-class>
27 </servlet>
28 <servlet-mapping>
29 <servlet-name>myServletMadeUpName</servlet-name>
30 <url-pattern>/myFirstServlet</url-pattern>
31 </servlet-mapping>
32 </web-app>
33

```

Figure (x) typical xml mapping

The figure below shows how the server knows from the Html form action what it is looking for

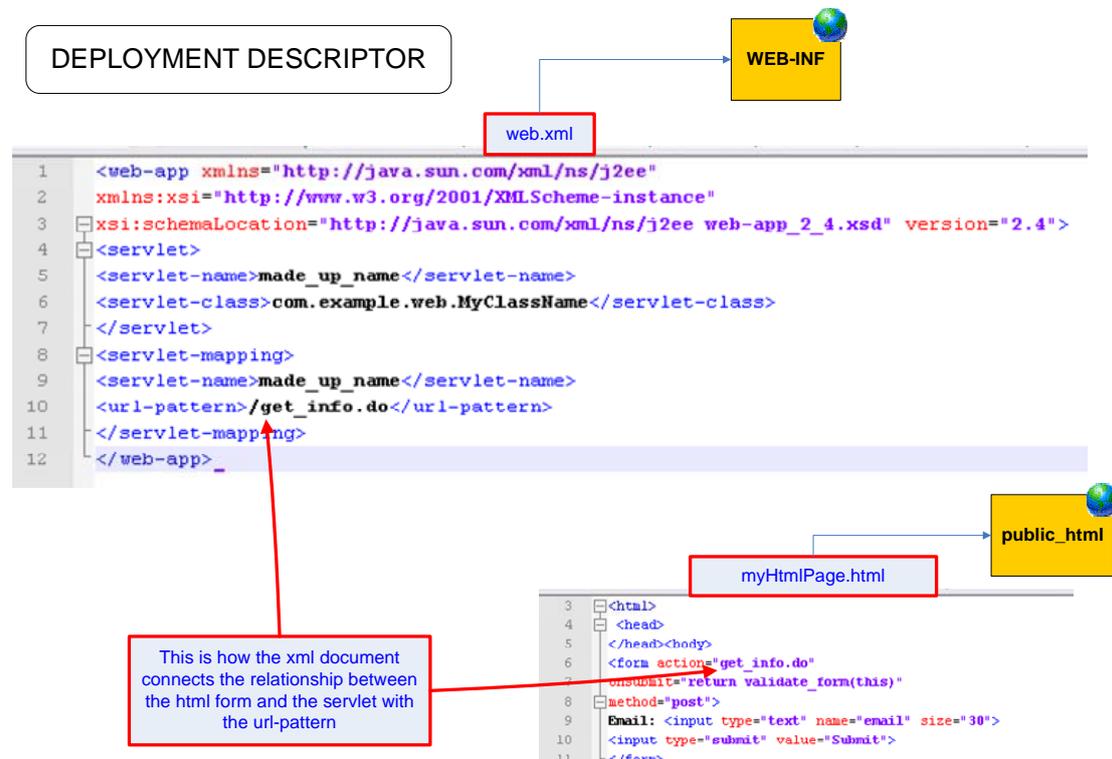


Figure (xi) – XML mapping to form action

Tomcat - testing the first servlet locally

The first servlet tested working and correctly shows the html form, and the information is collected and displayed according to the Java code in `petservlet.java` and did what it was supposed to do.

It was compiled in the command prompt using

Compile the servlet using:

```
javac -Djava.ext.dirs=$TOMCAT_HOME/lib petservlet.java
```

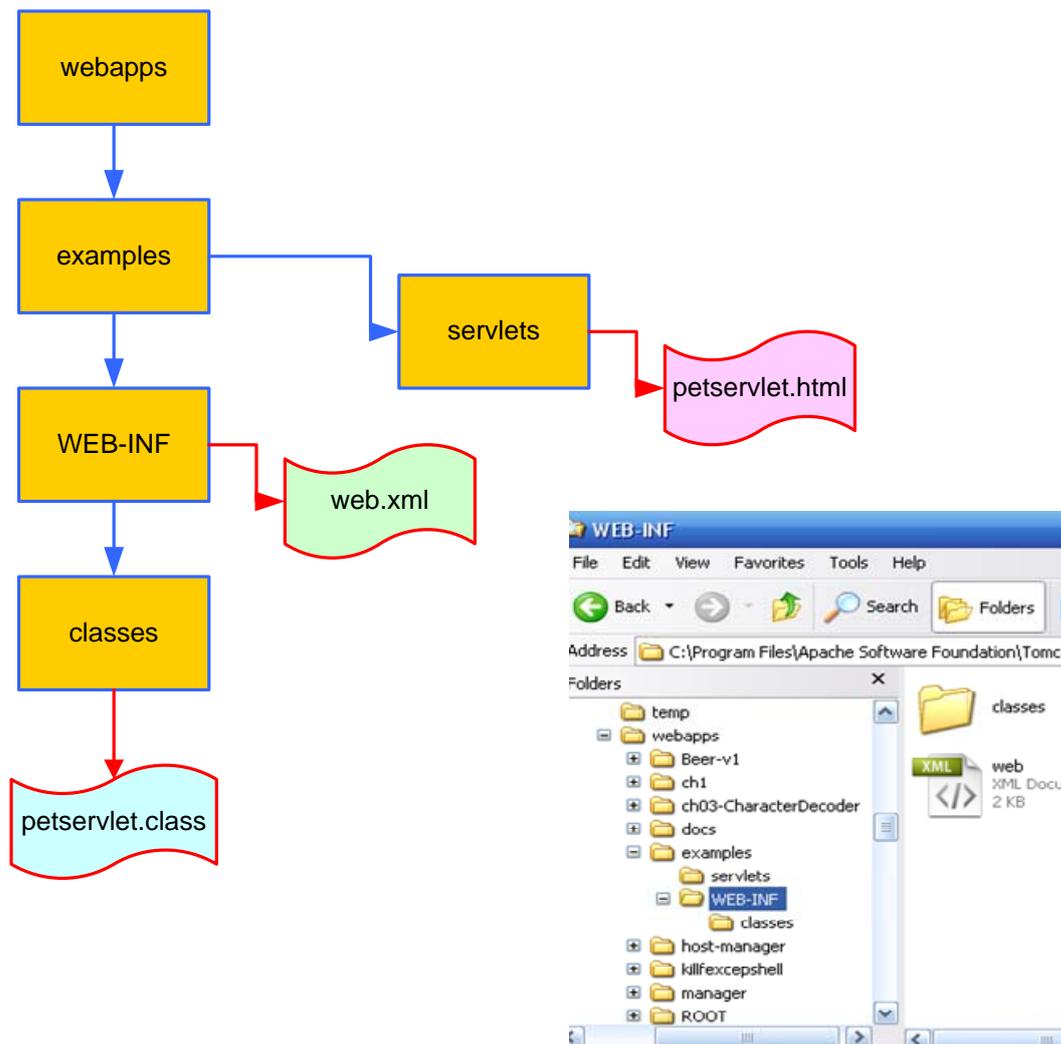


Figure (xii) – File/folder/directory configuration

The figure above shows the file/folder/directory configuration.

The classes folder must be contained in the WEB-INF folder, and the WEB-INF folder must be under webapps folder. The path to type into the browser is <http://localhost/examples/servlets/petservlet.html> to view the servlet.

Now that servlets are working locally, the next step is to purchase the webhosting server space and get them working live on the website.

Live Web Hosting Server Space

Server space generally operates one of two systems, Microsoft and Linux. But it does not mean that if a Microsoft Operating System is being used on a pc, that a Linux server cannot be used, and vice-versa. Carrying out research on server space showed that quite a few servers would support mySql but not Java Servlets. The general price seemed to be around £200 for an annual fee. However, Hostroute were offering a package for around £80 which they state is a reseller package, with reseller rights, up to 5 domain names, and it supports Java Servlets and JSP. This research was carried out early on in the project and the web host sourced, as if had not been financially feasible for the Durrington Festival Committee, Java Servlets would not have been used in the design process.

Communication with the Server Provider

The company chosen was based on two things, quality of service and price. Previously Hostroute have provided a cheap 10MB webhosting and domain name facility for an inclusive price of around £10 to £15 depending on the domain name type and this has proven to be a good quality service in the past.

The server space was therefore purchased and the username and password provided.

The domain name www.completewebsites.biz will be used to host the testing pages. Once the design is working complete it will be relatively simple to set it up for the www.durringtonfestival.org.uk site. It will mean either doing a

redirect, which could make the website slow, or to create an account for www.durringtonfestival.org.uk and transfer the files over. This means:

- rebuilding the class files
- renaming the database connection
- renaming the database (as the system uses the username as a leading database name)

DNS and Nameservers

This will need to be investigated more nearer the time, it may simply be a simple DNS transfer, pointing the nameservers to these new nameservers, using the admin panel of oneandone.co.uk which is the domain name provider of www.durringtonfestival.org.uk. By co-incidence, www.completewebsites.biz was purchased through oneandone.co.uk.

From an administrative point of view it is efficient to keep a log of all names that will need changing, their file names and their location.

The figure below shows a simplified structure of the live webhost and its folder, and the two testing servers on the local machine

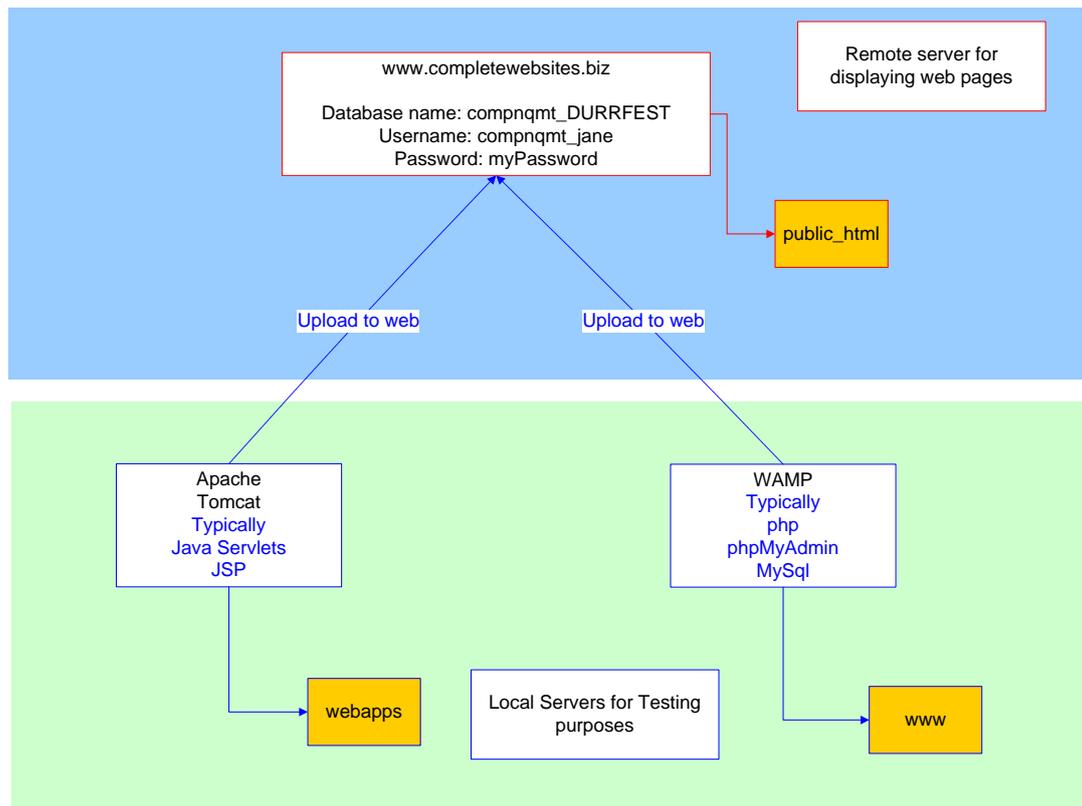


Figure (xiii) – Servers for Testing and Live Web Server

Administration

Username and passwords need to be logged somewhere safe. This is because there become so many of them. Here is an example of when building this site, although these are not correct names for security reasons:

1. Username: cmcncmcn
2. Password: a1b23Cz4
3. Nameservers ns52.abcdefg.co.uk
4. Nameservers ns53.abcdefg.co.uk
5. Account: www.mydomainname/cpanel
6. Manual: cpanel.net/docs
7. Email: mail.mydomainname.com(without the www)
8. Outgoing: Use dialup provider or isp
9. Support: <http://www.ustroute.com/supportcentral>
10. Support login: 12345
11. Support password: abcdefg

12. Helpdesk username: me@myemailaddress.com
13. Helpdesk password: 98765432
14. Upload: <ftp://mydomainnamewithoutwww.com>
15. MySql Database name: cmcncmcn_DBNAME
16. MySql Username: cmcncmcn_jane
17. MySql Password: zxzazxzazx
18. Admin Username: xxxxxxxx (for updating the website content)
19. Admin Password: xxxxxxxx (for updating the website content)

These are just initial passwords for webhosting and making a database. For example, point numbered 16. makes a username with a leading username. This could become confusing to a designer. It is clear when it is listed in a logical order.

One way of dealing with this is to make a spreadsheet containing all of the passwords, connections and other useful information and print it off and place it in a safe file.

Whilst working on the website development and implementation, this paper is a useful tool and may be updated and reprinted every so often as and when more usernames and passwords are created and/or changed.

FTP

Uploading to server space is usually done by FTP which is File Transfer Protocol. Tried using Internet Explorer to ftp but the browser would not allow it

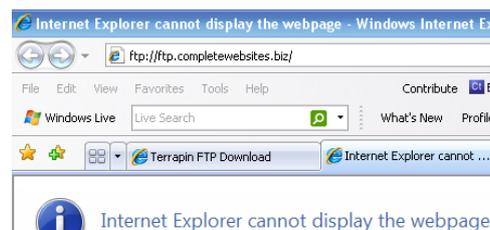


Figure (xiv) Internet Explorer

Possibly this is a security issue as am using the Harrow Hall internet connection and maybe ftp is not allowed.

Terrapin ftp

Chose Terrapint ftp and downloaded to use to ftp the files over to the server. Uploaded the files by connecting to the server and dragging the files over to public_html folder on the server.

Uploading the first servlet

The first servlet would not work once it was uploaded. Initially checked the code, it worked locally, checked the file configuration, all was correct. Changed the path address in the html document. Changed the servlet mapping in the xml document. Tried with different names. Made new folders on the server. Made new folders on local pc. Tried all that could possibly be tried. Changed the localhost to the full website name. It would not work.

The next step was to contact the web hosting company. This was the beginning of a series of polite and friendly emails backwards and forwards to Hostroute Support Systems Engineer.

On 18th March Hostroute confirmed that Servlets had been enabled on the account. Despite trying every possible way and alternative they would not work. The host stated that either the url was wrong, or that the Html page was not correct. A series of friendly emails continued to travel back and forth, over 20 emails in all, over the next 3 weeks.

Whilst working on the server/servlet issue and trying to resolve it, using the time passing before resolving this issue, the next task to carry out what will be the actual building of the database tables on the website was started.

Building the Database Tables on the Server

Setting up the MySql Database

We are going to be using phpMyAdmin to build the database tables on the live webhost. In order to create a database in phpMyAdmin, the first task is to login to the website control panel (www.mydomainname.com/cpanel) and then go into the mySQL Database area and create:

- database name
- username
- password

This information is added to the increasingly long list of important data of names, usernames and passwords for future reference. Once the database is defined there is an option to 'grant privileges'. If, for example, a user is only allowed to read the database, then the 'write' privilege would not be granted to that user. In designing and developing and implementing this database, all privileges have been granted.

phpMyAdmin

Making the tables - Here is a screenshot of one table, TBLFORTHCOMING created in phpMyAdmin.

There are 11 tables. Screenshots of the completed tables are shown in the Appendices (See Appendix (g))

The screenshot shows the phpMyAdmin interface for a MySQL database. The browser address bar indicates the server is localhost and the database is compnqmt_DURRFEST. The table being viewed is TBLFORTHCOMING. The interface includes a navigation menu on the left with 11 tables listed, and a main table structure view on the right. The table structure view shows the following fields:

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> eventId	int(11)			No		auto_increment	[Edit] [Delete] [Refresh]
<input type="checkbox"/> eventTitle	varchar(255)	latin1_swedish_ci		No			[Edit] [Delete] [Refresh]
<input type="checkbox"/> eventDate	date			No	0000-00-00		[Edit] [Delete] [Refresh]
<input type="checkbox"/> eventTime	time			No	00:00:00		[Edit] [Delete] [Refresh]
<input type="checkbox"/> eventVenue	varchar(255)	latin1_swedish_ci		No			[Edit] [Delete] [Refresh]
<input type="checkbox"/> eventImage	mediumblob		BINARY	No			[Edit] [Delete] [Refresh]
<input type="checkbox"/> eventLink	varchar(255)	latin1_swedish_ci		No			[Edit] [Delete] [Refresh]
<input type="checkbox"/> eventOtherInfo	text	latin1_swedish_ci		No			[Edit] [Delete] [Refresh]

At the bottom of the table structure view, there are options to 'Check All / Uncheck All With selected' and a 'Propose table structure' button.

(Figure xv – TBLFORTHCOMING as stored in mySql database compnqmt_DURRFEST)

Figure xv shows:

- eventId which is an integer, with an auto-increment. This means that each time a new event is added to the table, the eventId will be numbered with the next consecutive number. The data restriction on the integer is 11 digits, which will allow up to 999,999,999 entries.
- eventTitle - varchar allows 256 (0-255) characters and is the same as 'tinytext' so tinytext is not used
- eventDate - is in its default format of YYYY-MM-DD and the developer will need to apply some programming to the date so that it will display on a browser in a different way, for example Tuesday, 21st April 2009
- eventTime, which is in its database format of 00:00:00
- medium blob – to hold binary image files
- eventOtherInfo, which will allow 65535 characters

There are many options within phyMyAdmin, including:

- to browse the tables
- edit tables
- add, update, insert, select data – used for testing purposes
- drop a table

Standardisation and Naming

One of the most important items when creating anything more than the tiniest project is to pay particular attention to naming and file management.

Throughout this project we have used:

- lower case to start the name of a variable, or a field name, followed by uppercase for the first letter of any following word
- clear naming of tables, ie TBLENTY, it will be clear when reading code that it is referring to a table

- page naming of the website – where possible using the first five letters of the page title
- using meaningful names
- Accessibility pages, use the same name as the main site, with an added 'a'
- Using leading capital letter for Classnames

Relational Database

Now all the tables are built, we need to check that we have the ability to create the correct relational links, or joins between the tables. For example personId is listed in TBLPERSON, it is also listed in TBLENTY, because a 'person' will enter a Festival event, so instead of storing their name and address twice, it will be stored in TBLPERSON, and a matching personId in TBLPERSON will match with the personId in TBLENTY, when an SQL INSERT query is written, it will insert the relevant records relating to that person in TBLPERSON and in TBLENTY without duplicating or storing unnecessary data but simply using the personId to link the data. This refers back to the normalisation process, ie trying not to repeat the same information in different tables.

Consolidating Where We Are

We now have:

- a working java servlet development environment with all environmental variables set correctly, using:
 - Notepad++ to write the Java Class file in Java
- Tomcat/lib directory
 - to store the Java file that will become the Class file when it is compiled
- Command prompt with correct environmental variables set to work
 - to compile the file
- Tomcat/webapps/WEB-INF/classes folder
 - to place the compiled file in

- Tomcat/webapps/examples/servlets folder
 - to hold the HTML file
- Tomcat/webapps/WEB-INF
 - to hold the web.xml file
 - that maps the servlet to the server so that it will display on the browser
- Live webhost
 - hostroute – package providing facility for JSP and servlets
 - issues with servlets not displaying on the server, although
 - hosting site is displaying a working JSP Test page
- MySql Database Tables
 - series of database tables
 - designed and built
 - sitting on the server awaiting data

What is needed now is for the servlet to work on the live web host and the rest of the website may be implemented.

Meanwhile, friendly emails are travelling back and forth to and from the webhost trying to resolve this issue.

Again, whilst waiting, the site needs its HTML pages and its linked CSS pages, so that is the next task to be undertaken.

HTML for the Structure

HTML stands for Hypertext Markup Language. Its parent is SGML which is a system for defining 'mark-up' languages. HTML has been defined by SGML and is set out in the Document Type Definition which is listed at the top of HTML pages as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Usually if browsers do not pick up this line of script, they default to this Document Type Definition.

This line of code which requires stricter HTML code, ie all opening tags must be closed, has been used for all html pages on the site:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```

The alternative to writing HTML in a text editor is to use an interface, ie a tool such as Adobe Dreamweaver, which is a 'What You See Is What You Get' (WYSIWYG) interface. There are a wide variety of tools such as Adobe Dreamweaver on the market. However part of the learning process of this project is to work with Notepad on coding, scripting and writing HTML mark-up language. Also Adobe Dreamweaver adds lots of tags and unnecessary information to the files. Now becoming used to writing in Notepad, and appreciating the simplicity and the 'clean' code, there is no inclination to use the WYSIWYG tools.

This is a typical HTML page

```
<html>
  <head>
    <title>Title here</title>
  </head>
  <body>
    Main body of content here
  </body>
</html>
```

The general rule is that each opening tag must have its closing tag according to how the Document Type Definition defines its layout.

There are a lot of other tags to use, for example `<style>` new style here `</style>` will allow styles to be added to the script or, `<script>` some Javascript here `</script>` will allow Javascript to be used in an HTML page.

The HTML pages are now built, and a navigation bar created and sample text (Lorem Ipsum) put into the pages.

Notepad was used to write all the HTML pages and they contain only structure with absolutely no additional tags or formatting. The code appears tidy and neat and easy to read.

The HTML files for the website are contained in the Code Appendices (See *Code Appendix (A)*).

The next step is to style the pages, before doing that though, consideration needs to be given to Human and Computer Interaction principles (HCI)

Human and Computer Interaction

The concept is to create a user-centred design. This means applying principles such as Gestalt, understanding perception of colour, Jakob Nielsens theory of the ten important concepts to apply when designing a user interface.

The question is, who are the users? They are certainly the people that browse the website for information, the people that complete forms online. Some of these users may be:

- not very familiar with computers/internet
 - solution – straight-forward design and clear consistent layout
 - more than one way to achieve an interaction
 - use of existing symbol knowledge – ie green for go
- have old machines
 - allow the browser display to shrink to size without losing information

- have new machines
 - create a horizontal split in the text to achieve shorter reading span over longer line length
- have accessibility issues
 - text only pages in black font on yellow background as is the most suitable colour for people with vision defects
 - ability to change the text size
 - keyboard shortcut codes in text pages
 - tab order to be set to minimal tabbing
 - Use of ems and percentages as a measurement in HTML as opposed to pixels where possible
 - navigation horizontally on text pages so users don't need to scroll

The other users are of course the users are the Durrington Festival Committee Members themselves, in the sense that they will be providing an Administrative role in updating the website themselves.

The colours selected for use in the design are in the palette shown in Figure (xvi) below. The numbers are their hexadecimal values.



Figure(xvi) – hexadecimal colour values

The code for the keyboard shortcut is:

```
<li><a href="indexa.html" ACCESSKEY="T">Text Pages Home</a></li>
```

The link to the speech browser also opens in a new window, so that the user does not 'lose' the site when downloading the speech browser by coding:

```
target="_blank">
```

Sound and Voice Matching

Addressing Accessibility, one good idea would be to be able to implement word matching, so that a person may say 'next' to move to the next page, or 'back' to move back. Some simple voice commands would open up a whole world for some people who can speak but who cannot operate a mouse or a keyboard.

It would seem relatively straight-forward to be able to match a sound pattern to a library pattern. Some detailed research for this project has been carried out during the project period and it mentioned within the Progress Report. It will be implemented and tested outside of this project period.

Cascading Style Sheets for the Style

Cascading Style Sheets are known as CSS. The point of CSS is for the style sheet to control the style of the website. The reasons for this are:

- ease of changing information
 - website wide
 - by page
 - by element
- assists with Accessibility
 - not so much code for a speech browser to read
 - ability to change the look of the webpage 'on the fly' i.e. whilst browsing on the site

The current policy for designers is to create a design where the structure of the page is written in HTML .

Structure means for example:

- the way that the page displays, ie the HTML for buttons, option lists, paragraphs, sentences, titles

Then, the current policy for designers is to use Cascading Style Sheets to 'style' the site. Style means, for example:

- the colour of the text, the layout of the text, the width of the page content

There are four types of design layout in CSS:

- static
- fixed
- liquid
- elastic

Static is not used as fixed is virtually identical.

Fixed

- is for example drawings, or any other display where the content is designed to remain the same size whatever the size of the browser window. This means that if the browser window is small, a person may have to scroll down, and/or scroll across to read the page content. If the browser screen is large, then the page may look strange only taking up a small portion of the browser space.

Liquid

- is when all the items are floated and they arrive on the page in their floated order. If someone has a really wide screen, this can make the web page display in a strange manner and certainly will not display the way the web designer intended it to.

Elastic

- is a mixture of fixed and liquid, together with adding 'stretch' to items like images and fonts, using percentages and ems as a measurement so that for example a user may increase the size of a page and the page contents will all increase in unison.

There are various measurements to consider when designing the pages, in particular:

- pixels
 - will be used for fixed positioning

- percentages
 - will be used for page widths
- ems
 - will be used for sizing text

To link to an HTML page to a style sheet, at the top of the HTML page, type:

```
<link rel="stylesheet" href="myStyles.css" type="text/css">
```

Then create a stylesheet in notepad and save it as myStyles.css. Here is some sample CSS. The general rule is either a class or id is created in the style sheet. The tag name may be used which will automatically apply the style, or a class or an id may be written and 'called' within the html tag and applied to that area.

HTML	CSS	TYPICAL USE
<code><div id = "myName">content here</div></code>	<code>#myName { border-style: dotted;}</code>	For only one item of that name on a page.
<code><div class ="myName2">content here </div></code>	<code>.myName2{ border-style: solid;}</code>	Border around items Can use more than once on a page
<code><a href etc>content here</code>	<code>ul.myName3 a{ display:block;}</code>	Targets the 'a' and 'ul' in the elements, usually for vertical navigation buttons
<code>Home</code>	<code>a:hover{color:F9999 99}</code>	Will change the colour of the text on mouse over to F999999
<code><div id = "mainNav"> Home</code>	<code>#mainNav a:hover{color:F9933</code>	Will target a elements only with

	99)	the <div> and </div> tag
--	-----	-----------------------------

Figure (xvii) – some CSS to demonstrate how it works

CSS 3 will have attribute selectors including the ability to select attributes that start with a certain value (like http://) or end with a certain value (like .jpg or pdf).

Testing of CSS on a 16:9 screen showed some drastic errors in display, so these were corrected by adjusting the style sheet.

CSS has the ability to float content, ie right or left, so can use floated content with padding around it to create text to float around an image, for example.

CSS style sheets are attached in the Code Appendices (See Code Appendix (B)).

Now we have the HTML structure and the CSS styled, the tested pages were uploaded to the live host and tested as working correctly.

The servlet issue has still not been resolved although emails are still travelling backwards and forwards, and further testing of servlets is being carried out on a daily basis.

Beginning to consider buying alternative web hosting space, or using php and MySql instead of Java servlets. Have offered to pay the web host to sort out the servlet issue although this has not been taken up. The next step is to work out how to connect to the MySql database which is sitting on the live hosting space, that is assuming servlets will be able to run.

JDBC

JDBC is a API that will allow SQL statements to operate. As Java is multi-platform, this means that using JDBC to connect to a database creates a way

of connecting to many databases that operate SQL whilst Java itself is platform independent.

Java designers wanted to provide a one-stop solution to connecting to databases. However each database vendor is different and uses different systems and terminology and it soon became clear to the Java Development Team that this is a virtually impossible task. So each database vendor makes a driver that will connect to the JDBC driver. The diagram below shows how the physical connection takes place.

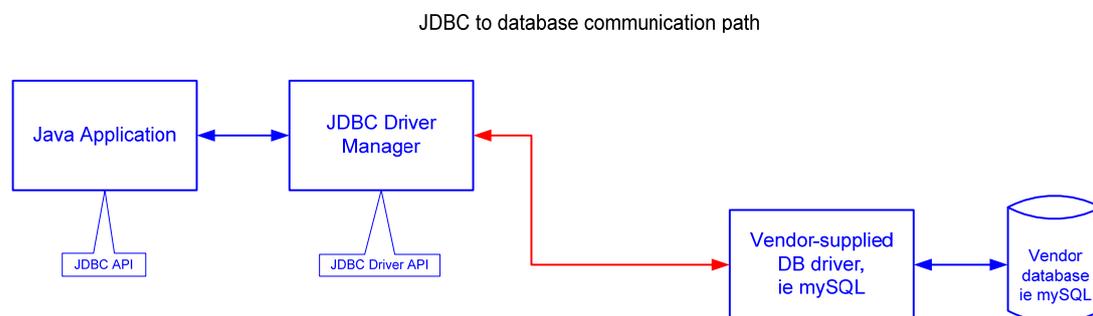


Figure (xviii) JDBC to database communication path

The programming code needs to create a situation where the two drivers connect. The JDBC driver needs to be told the name of the driver to connect to. And the database url to connect to, in this instance the website address and database name.

Establishing the connection details was not straight-forward. Contacted the web host and asked for the driver name and they provided the wrong name of the driver. Searched on the internet and found a variety of driver names, but the two that most stood out were:

`org.gjt.mm.mysql.Driver`

and

`com.mysql.jdbc.Driver`

`jdbc.drivers=org.gjt.mm.mysql.Driver or com.mysql.jdbc.Driver`

- to connect the driver to the database driver

```
jdbc.url=jdbc:mysql://localhost/databasename
```

- to connect to the database

```
Class.forName("").newInstance();
```

This seemed to be from research in books and from searching the internet, the most important parts of the code to connect.

Outstanding Servlet Issue

Now virtually everything is in place except for the servlet issue. Testing the servlet could still not resolve the issue.

Everything had been tried. Clearing out the files and reinstalling them. Checking the code. Testing again in Tomcat. Recompiling the class files. Checking the paths, changing the paths, testing again. Deleting all and starting again. It was impossible. Located a private tutor, but he did not understand server side, and was only prepared to work locally to get a servlet to work, which had already been achieved several times.

A final email was sent to Hostroute on 6th April 2009 in the early hours of the morning, after exhausting every single possible avenue, stating that despite trying and trying every possible way of getting a servlet to work, it was impossible and had now given up on using Java Servlets as running out of time to implement the servlets.

Took all of the class files, the test files, the jsp file and the html files in relation to the servlets and the web.xml off the website, cleaned up the Directory ready for working with php and MySql.

Work started with php and mySql, a connection was created to the database and the website design was proceeding immediately.

Then received a reply from Hostroute later in the day on 6th April to say that they had made some changes to the server.

Reluctantly uploaded a servlet class file expecting to see the same error messages that had been displayed hundreds of times during this process only to see with absolute total surprise, the servlet was working. Loaded the next servlet. That worked too. And the next. It was seamless and simple.

For the future, will have a simple self-contained servlet and when purchasing webspace, will run the servlet to test it. If the servlet does not run, then the likelihood is that the server is not configured correctly as is possibly what happened in this case.

Just as a footnote, the support service from Hostroute was first class. And the price is incredibly low.

Java Servlets

Now that we have the ability to use servlets, we need to write the code.

A standard servlet has a post and get method, each with a request and a response.

Two-tier servlet

Quite an interesting concept, the servlet's 'Get' method is called first. Here html may be written to create a form with some input field names and a submit button. The URI is picked up and put into the form post method, the form then displays, already that means that the 'get' method has run, the post of the form calls the 'post' method where the next 'do something' occurs, in this case accessing the database for data to be updated. It seems that servlets can be designed where a user may be hopping between one or another servlet, or the same servlet using HTML.

Or if just the post method is to be used, then the get method can call the post method, and it should work vice-versa.

Implementation

Introduction

The starting point for the implementation is the Entry Form. This is because the public will be using the form soon to enter the Festival which starts in June. Even if the rest of the website is static, ie with HTML pages only, the Entry Form may be used for the purpose it is designed for. Once the Entry Form is designed, then we will work on the admin section, to view the Entry Form details. Once this is working, we will work on an interface for downloading text content from the website to a form field, making changes to that field and then uploading it again. Once this is working we will then feed that content into a jsp page. The first class file written is ServletUserEnquiryForm1.java. (See Code Appendix (C) for java files)

ServletUserEnquiryForm1.java – design detail

Associated file: entry.html

This file is compiled and tested working correctly. The design procedure is detailed below:

1. An HTML form, entry.html is designed using html and css to style to form.
2. The form action calls /servlet/ServletUserEnquiryForm.
3. Javascript validates the form
4. The class file collects the parameters from entry.html
5. Sets up the connection parameters for JDBC
6. Loads the mySql database driver
7. Uses a PreparedStatement to insert data values to TBLPERSON
8. Uses createStatement to get the last personId from TBLPERSON
9. Uses Prepares Statement to insert data values to TBLENTY, including the last personId from TBLPERSON
10. Checks the number of rows changed
11. Prints message to user
12. Closes all the resources

Testing

Iterative Testing carried out during build, further testing listed below.

ServletUserEnquiryForm1.java		
Test	Fault	Remedy
Data input to form	error on number of people if no data entered	Set default to 1
Data input to form	error on radio button	Set default value
Data input to form	Unable to read any data in	Javascript validation written to 3 fields
ID's match	No error	No error
All data stored in database	No error	No error
Data displays correctly	Yes, however there is no date.	Put timestamp into database and code and recompile (Outstanding)

Areas of concern

Issue: synchronization.

If there is any way that another servlet could access the database at the time when it this servlet is carrying out a transaction, it may corrupt the data and this needs to be tested and investigated, to see whether this is possible, or whether synchronization is needed to be added to this code to stop the possibility of this occurring.

Issue: database behaviour

If the connection is open, need to identify if another connection may be open at the same time. If not, then consider it is virtually impossible for the data to corrupt due to lack of synchronization.

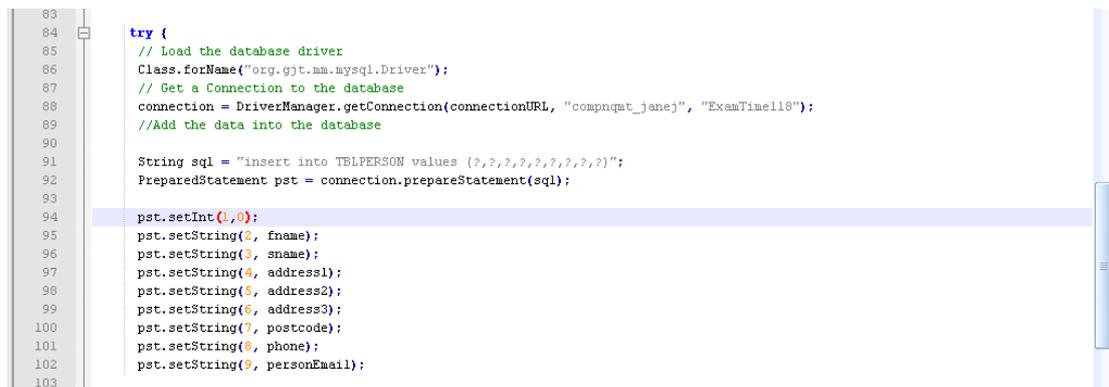
Issue: clean data

Other languages have various functions for 'cleaning' data input to stop people writing code to the database, for example programming type symbols are converted. Sourcing information about Java, and because it is compiled, the code is apparently cleansed automatically, for example a + is put in for spaces, however need to test that that is correct information and that this is working correctly by testing programming symbols and checking the raw database data.

Tasks that were time-consuming

Auto_increment is a feature that automatically increments the personId by one iteration each time a new row or new record is inserted to the mySql database. However, when coding and using a prepared statement to query the database, it was difficult to get the code to work with the auto_increment. Searched the forums and the internet for the answer, and could not seem to find the answer. Created own solution by trying all sorts of data test, and found that setting the value to 0 forced the database to auto_increment. Tests and works correctly.

Below is figure (xiv) showing a screenshot of the code in Notepad++. where it says `pst.setInt(1,0)` refers to the prepared statement, set Integer in column 1 to 0.



```

83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
try {
    // Load the database driver
    Class.forName("org.gjt.mm.mysql.Driver");
    // Get a Connection to the database
    connection = DriverManager.getConnection(connectionURL, "compnqat_janej", "ExamTime118");
    //Add the data into the database

    String sql = "insert into TBLPERSON values (?, ?, ?, ?, ?, ?, ?, ?, ?)";
    PreparedStatement pst = connection.prepareStatement(sql);

    pst.setInt(1,0);
    pst.setString(2, fname);
    pst.setString(3, sname);
    pst.setString(4, address1);
    pst.setString(5, address2);
    pst.setString(6, address3);
    pst.setString(7, postcode);
    pst.setString(8, phone);
    pst.setString(9, personEmail);

```

Figure (xiv) – part of the code for ServletUserEnquiryForm.java

The next step is to create a system that allows the Administrator to view the data collected from the Entry Form.

DisplayServlet.java – design detail

Associated files: admin.html, csEntryFormReport.html

This file is compiled and tested working correctly. The design procedure is detailed below:

An HTML form, csEntryFormReport..html is designed using html and css to style to form.

1. The form action calls /servlet/DisplayServlet.

DisplayServlet.java		
Test	Fault	Remedy
Display Information	No errors	
ID's match	No errors	
Time	No errors	
Display readable	Not clear	Text added to html so data becomes more comprehensible
All correct data retrieved from database	No error	No error
Data displays correctly	Yes, however there is no date.	Put timestamp into database and code and recompile (Outstanding)

2. doGet is called to display an html format and submit button.
3. The form action calls the DisplayServlet doPost method.
4. doPost prints out the html.
5. JDBC Connection parameters are created.
6. SQL statement created to select from TBLPERSON and TBLENTY with inner join on personId.
7. Query is executed.
8. The details are placed in a recordset.
9. The recordset array is looped through.
10. out.print displays the data.
11. Some minor formatting and text titles input to html to be output by the servlet so that the data is easily readable.
12. Closes all the resources.

Areas of concern

Issue: Data archiving

The data should be archived at some stage.

Issue: Security

These details are people's personal details and need to be managed correctly. The data needs to be kept in a secure area of the server and further investigation on security needs to be carried out.

Authentication is currently being carried out by the server default authentication creating a password to access the area of the files. Added authentication using more security if possible could provide further security benefit.

Issue: Back up of information

An assessment will need to be made by the Committee how problematic it would be if this data was lost. It is probable that this data would need to have a back-up stored somewhere away from the server or on another server.

Issue: notification to Administrator

An email automatically generated to both entrant and Administrator would aid communication in alerting Administrator to new entrants applying online.

Before continuing, it might be useful to look at the working drawing in Figure (xx).

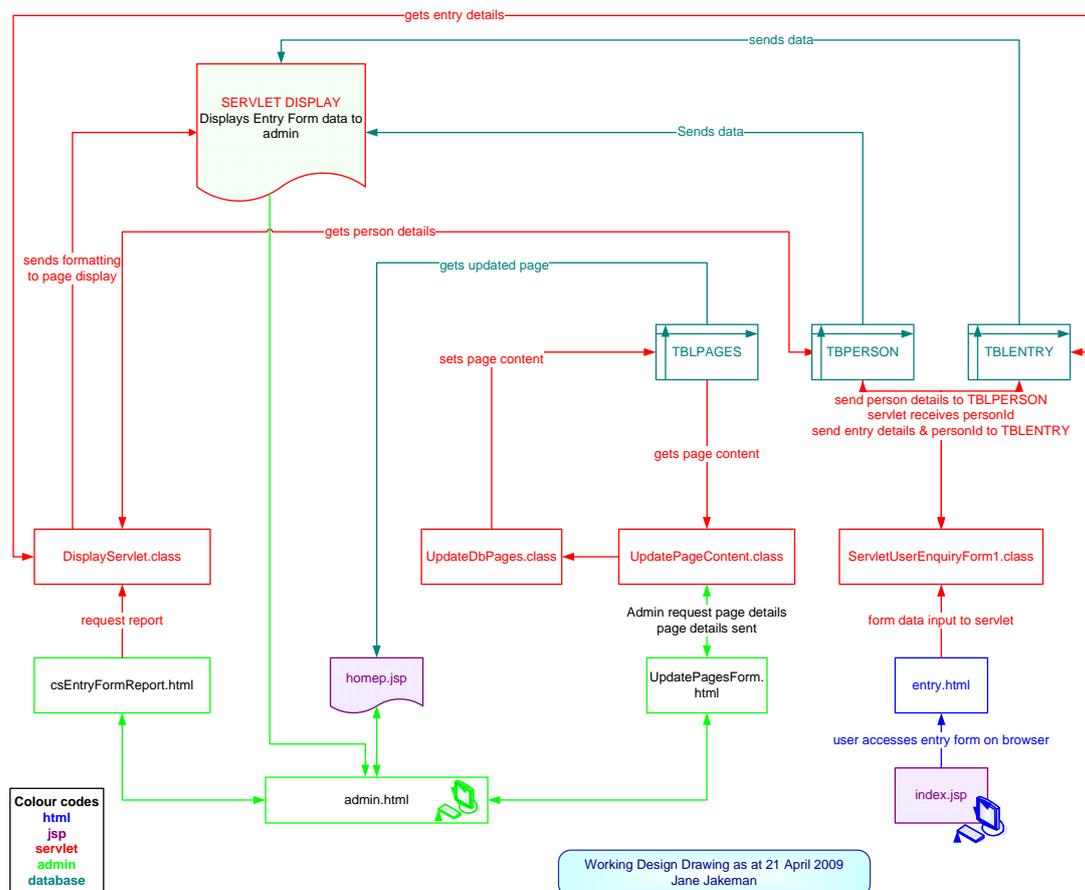


Figure (xx) – Working drawing

This drawing was borne of the need to understand the data flows, the naming of pages and how they were communicating with each other so it was designed as a management tool for implementation.

Anyhow, back to the DisplayServlet.class, it creates a display that displays the relevant information to the Administrator.

The site needs User Authentication design being Session Management, however, until this code is written the Server provides a facility for a protected area and all Administrative functions are in the protected area for the time being.

UpdatePageContent.java – design detail

Associated file: updatePagesForm.html

This file is compiled and tested working correctly. The design procedure is detailed below:

1. An HTML form, updatePages.html is designed using html and css to style to form.
2. The form action calls /servlet/UpdatePageContent
3. The doGet method refers to the doPost method
4. Collects text value “formPageCat” and creates String fPageCat
5. JDBC connection to database created
6. creates select statement using String fPCat and order by pageDate to get the last entry
7. Loop through the recordset and put the column data into variables
8. Display the data in html format
9. Close all resources.

This then leaves the data onscreen for Admin to update at their leisure. When they have finished updating the text fields, they press ‘submit’ and it calls the next servlet (UpdateDBPages).

Testing

Iterative Testing carried out during build, further testing listed below.

ServletUserEnquiryForm1.java		
Test	Fault	Remedy
Data input to form	No fault, data accepted	
Change data	Change accepted, no error	Set default value
Submit works?	Submits properly to database, calls servlet correctly, no error	

Areas of concern

Issue: text field size

Need to change the text field to a text area and use slightly different code.

Not certain that a servlet will pick up the parameter, need to test it. textarea in html doesn't seem to have a value as an attribute, although have seen it working elsewhere, just need to identify the correct id procedure.

Issue: If admin want previous data

Need to write in a facility to provide for this.

Tasks that were time-consuming

The code for this was quite time consuming. There is probably a simpler way of updating information but the way it is designed to work is that the data is fed from the database to a page the admin person may view. The database connection closes. The admin person does whatever work on the data they wish. When they are ready, they submit the new data, or updated data and it is entered to the database, order by date, using a new servlet. Whenever data is required, it is retrieved on last date, so the most recent data will be displayed.

1. UpdateDbPages.java
2. doPost creates the connection
3. Statement prepared, data to insert to TBLPAGES.
4. Connection and all resources are closed

index.jsp

Two variables have been placed in index.jsp to pick up:

1. Page Title
2. Page Content

UpdateDbPages.java

This servlet is operational once the admin person has carried out all the amendments to the text. It sends the updated text to the table. Really it is not updated text, but new text and is timestamped. From the point of view of the admin person, their concept will be that they have simply changed the text, in reality there will be a new record, with new text.

This will apply to all text in all pages in the website that the Committee wish to be able to update when they want.

Testing needs to be carried out on a larger scale to see if there are any major delays in displaying the text. If so, there needs to be a different solution and possibly xml may provide that, however this is just an idea and needs further investigation.

Testing:

The first time the data is 'displayed' by the server, the display is slow, as it must be compiling the Java code, but once it is compiled it becomes fast for the next and all following displays. It is likely to only be updated on an occasional basis, however the fundamental element of page content management is there with quite a flexible facility.

Issues: one issue would be if the .jsp had a problem displaying and it would be good to write in some code for it to display a default html page if the jsp element didn't work for some reason, just so that there was a web presence.

JSP

The variable produced by the class file above, needs to be inserted to the jsp page. We had a page called index.html, we have added some Java code to this and now call this index.jsp. It is uploaded, tested, the Java has compiled and the server has found the file, all is tested working.

MVC

Using Servlets is using the Model/View/Controller design. The model is the collection of classes, the view is Java Server Page and the Controller is the Http element.

Outstanding

Sessions (Authentication and Authorisation)

Session management needs to be created and added to the website to allow login by members of the public as well as the Administrator.

Forthcoming Events

The Java file needs to be written for this page.

Prototype

The work carried out to date can be viewed as the prototype for the actual website. The technology has proved itself, there is some more code to write and then the prototype may be applied.

Javascript

Was used to create validation on the form. Validation is used only on essential fields, first name, second name and email address. The email address is validated when an @ is found in the address and when the punctuation mark, full-stop, is found in the address. Also there is a test on the position of these. The script was sourced from w3schools site and is referenced in the code. Script for email input validation was also sourced from w3schools site, and the function call was rewritten to accommodate two methods of validation, and the function is called on the 'on submit' button of the Entry Form. Uploaded to the web, tested working on all three fields.

Java Applets

Java Applets and Java servlets may work together. A servlet may work for an Applet, ie connecting to a database server on another machine and passing the information to the Applet. Applets run on the browser, client side, whilst servlets run on the server, server side.

Soap and php

A simple example of Soap Server using php has been included to demonstrate the use of Soap which was written in another coursework. It is a way of contacting a remote server and obtaining data. The key to soap services is to use the information contained in the .wdsI file sitting on the remote server. For example, the one used here provides current currency statistics. The information is requested using the parameters set out in the remote .wdsI file and the response is displayed in the php file, which when scripted within HTML displays the result.

Tools Used

The complete tool used for this project was only one. Notepad++. All of this has been achieved using the simplest text editor.

The only other software used, apart from using phyMyAdmin to create the database, was Adobe Photoshop.

Photoshop

The image at the top of the website was created with a new file of 700 pixels x 120 pixels. Then created a new layer, opened a photo, copied and pasted, used transform and then a gradient. Checked the banner and decided to float it 'right' so cropped the logo off the banner, reduced the size to 584 pixels wide x 200 pixels high and then created the logo separately and used HTML and CSS to structure and display the image.

Evaluation

This has been a long and complex project with a steep learning curve for the author. First of all, the challenge was to write with Notepad only. So all of the tutorials that I read, the books that I got, and the information I received, I did not find one single author from any media that used just Notepad. For example, writing a WAR file (a Web Archive File) or an EAR file or a JAR file, the instructions and tutorials all relate to certain IDE's such as Netbeans which would automatically prepare the files, and it was difficult to get information. The internet is and was the most valuable source, because if one search didn't work, another could be tried, different words, different angles, kept chipping away at the issues that needed resolving and approached them from all different angles. Now in retrospect it all seems so easy, but it is always easy when you know how.

During the research and working with the servlets it seemed as if I was constantly stumbling upon new technology, new ways of carrying out tasks, and one technology would always lead to another, for example, Java Beans, they would have been my next step to study, and tag libraries.

I think that XML needs exploring for efficient data transmission for this website. For example, instead of retrieving data from a database, maybe the data could be sent to an xml file and fed through to the html file and sit on that html file.

I am sure there is that technological ability out there, someone somewhere will have written about it and it will be on a website, or in a book for me to find. But at the moment I do not know where it is or how to approach it. This is how I was at the start of this project. Although I feel I have learned a lot, it has also made me realise how little I know and how much more there is to learn about programming for websites.

As for the website, it functions, it has been tested and works, however it needs strenuous testing. The best testing will be to monitor it live and pay close attention to what it is doing. The plan is to continue to build the website after this project period and launch it as soon as possible.

The website needs a lot of repetitive work to become fully functional.

Conclusion

On my next venture, I will try and source as much information as possible before embarking on a project such as this, plan the files and the layout of the directories. Collect as much information as possible. Liaise with the client as much as possible. And I won't underestimate the amount of sheer repetitive work that goes into a website.

All in all, the website looks professional, the HTML is just the structure, the CSS provides the style, which is a text-book design concept, it meets some

Accessibility issues, there is much more to add to the site, the entry form works perfectly and appears professional

Abbreviations:

API stands for Application Programming Interface

JVM Java Virtual Machine

HTML Hypertext Mark-up Language

CSS Cascading Style Sheets

JDK Java Development Kit

J2EE Java 2 Enterprise Edition

JSP Java Server Pages

Reference:

1. List of MySql data types sourced from:
<http://dev.mysql.com/doc/refman/5.0/en/storage-requirements.html>
[18/4/09](#)
2. Source http://www.w3schools.com/Sql/sql._join.asp

Books

Bakharia, Aneesha. *Java Server Pages* [2001].

Coleman, John. *Introducing speech and language process* [2005].

Jim Farley and William Crawford with Prakash Malani, John G Norman and Justin Genthland. *Java Enterprise In a Nutshell* [2005] Third Edition, O'Reilly.

Cay S. Horstmann, Gary Cornell . *Core Java Volume II Advanced Features* [April 2008] Eighth Edition. Sun Microsystems Press.

Englander, Robert. *Java and SOAP* [2002]

Doug Lowe and Barry Burd. *Java For Dummies* [2007] Second Edition. Wiley Publishing Inc .

David Sawyer McFarland. *CSS The Missing Manual* [August 2006] First Edition Pogue Press O'Reilly.

Michael Moncur, *JavaScript in 24 Hours*, Sams [July 2006] Fourth Edition.

Norman, Donald A, *The Design of Everyday Things* [2002]

Bruce W. Perry, *Java Servlet & JSP Cookbook* [January 2004] First Edition

Reinheimer, Paul Michael, *Professional Web APIs with PHP* [2006]

Hugh E. Williams & David Lane. *Web Database Applications with PHP and MySQL*

[2004] Second Edition, O'Reilly.

Web sites visited

<http://www.stardeveloper.com>

<http://www.roseindia.net/servlets>

<http://dev.mysql.com>

<http://java.sun.com/j2se/1.4.2/docs/api/java/>

<http://www.w3schools.com>

<http://www.stardeveloper.com>

<http://java.sun.com/developer/onlineTraining/Servlets/Fundamentals/servlets.html>

<http://www.wikipedia.com>

www.youtube.com

Learning Sources

Learning and tuition was sourced from:

- internet
 - websites
 - discussion forums
 - video tutorials
- lecture notes from
 - Human and Computer Interaction (HCI)
 - Web Applications Development
 - Java
 - Digital Video

- library
 - see Reference>Books in this report for a Bibliograph
- private tuition
 - two-hours private tuition in mySql, Java, JSP, XML and file management