

CPLO P

Cal Poly's Library of
Pyroprints

Senior Project
Guide

Kevin Webb , CPE Fall
2011

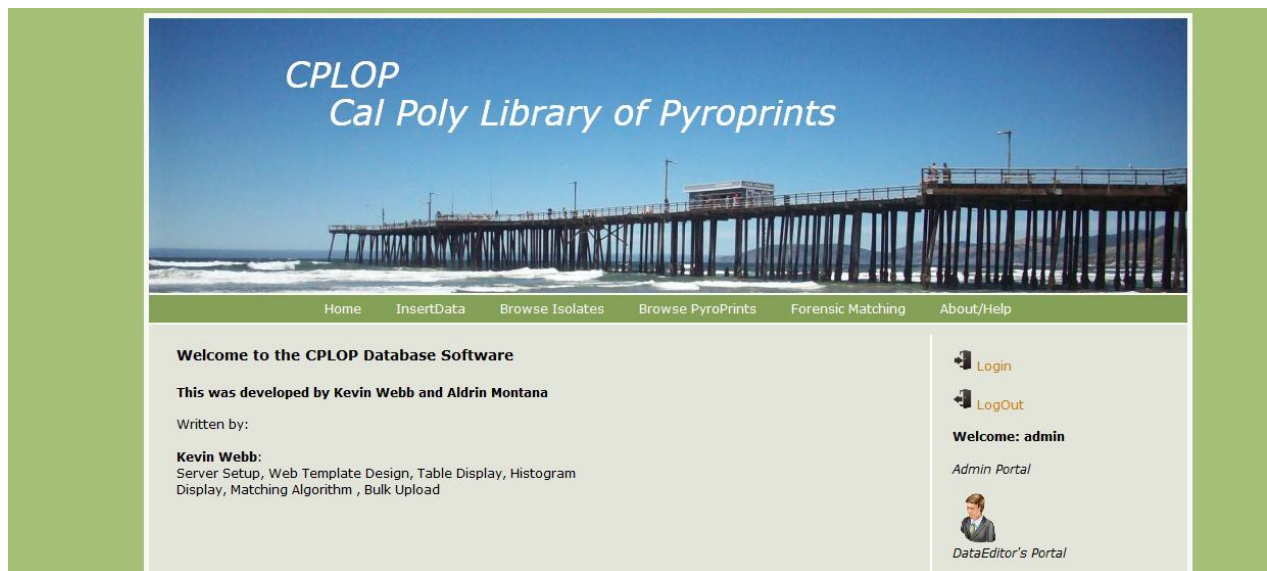
Edited by Jennifer VanderKelen
October, 2011

About this Guide:

This guide will be broken into two parts, one geared for a biology student or other end-user; and another for a php web programmer to add additional_functionality to this project in the near future. It will be important for the programmer to read both parts, as the database will be described abstractly in the beginning.

Thanks _____ and Credits:

Before we begin, I want to thank Alexander Dekhtyar for starting this project in CPE 366 and advising me throughout its progress. I want also thank Christopher Kitts and Michael Black (Biology Department) for their time answering questions, using laymen's terms, that brought this project from a concept into a production reality.



**Figure 1 Main Page of
CPLOP**

Part 1: An End-User's Guide

California Polytechnic Library of Pyroprints, CPLOP, is a web driven data-base application that stores data from the biology's departments *E. coli* Pyrosequencing project. Some of this data was stored in Excel datasheets, while data from the pyrosequencing machines was stored as just a list of random .xml files. There was no useful way to organize and store the massive amounts of data from multiple file sources in one location, nor to perform the complicated searches and comparisons that the project requires. CPLOP's primarily goal is to store such data in three organized tables that relate to one another. It was designed to organize *E. coli* strain isolation data, as well as the pyroprint data generated from analysis of these strains. The details of each table will be documented below.

WebSite Address: <http://cplop.cosam.calpoly.edu/>

The Actual Data:

Before we go into all of the web site's features, it is important to understand how this database is organized. The database is comprised of three main sections: a) the Isolates Table which includes information about each *E. coli* strain isolated for this project, b) the Pyroprint Identification Information Table which gives details about each pyrosequencing run including strains tested, well location, and analysis region (23-5 or 16-23 rDNA intergenic transcribed spacer (ITS)), and c) Pyroprint Data which includes the peak height data from the pyroruns represented as a histogram. The three sections depend on each other for a full understanding of how *E. coli* strains are related to each other based on the pyroprints of their 23-5 ITS and 16-23 ITS rDNA regions.

Section1: Isolates Table

Isolate ID	Common Name	Host ID	Poop Sample #	Freezer Location	Date Stored	Sampling Date	Site	Contributor	TA Notes
Ct-001	Cat	10	1	Rack 4	6/18/10	4/23/10	SLO	J. VanderKelen	

This table is also known as the freezer stock log. It is accessible using the "Browse Isolates" link. It contains information about each *E. coli* strain in the EBI collection. The *E. coli* strain is named based on the host from which it was isolated. The strain name is a two letter abbreviation of the name of the host. The Isolates table also includes the common name of the host species, an identifying name or number for each unique host, the poop sample number, where and when the strain is stored,

(and where, when and by whom the strain was isolated by). It should be noted that multiple strains may have been isolated from a single host, so the host ID will not necessarily be unique, but each *E. coli* isolate is distinct.

The Isolates Table is populated using two other tables called “Manage Host Data” and “Manage Samples Data” which are only accessible to those with Level 1 or 2 administrative accesses. These tables were required for the construction of the Isolates Table. The Host Data table gives information about the specific host; each host is uniquely identified by a species to which it belongs **and** by an integer number or name. If later other features of the host need to be collected and stored in the database (e.g. age of host), this information can be added to the Hosts Data table in a way that does not require a major database overhaul. The Samples Data table stores the list of all poops from which *E. coli* isolates were collected. A poop is uniquely identified by the host from which it came (which is the species and ordinal number described above) as well as the ordinal of the poop from that particular host. So, “Hu” “17” “03” means poop number 3 collected from Human 17. Having these two separate tables to populate the isolates table allows us to make changes about a specific host in one place that will then be properly updated throughout the Isolates table wherever needed.

Section 2: Pyroprint Identification Information Table

PyroID (Computer Generated)	IsolD	Amplified Region	Dispensatio n Name	File Name	Well ID	Tech	Pyroprint Date	Forward Primer	Reverse Primer	Seq Primer
1	Hu-001	23-5	AACACGC GA23(GAT C)GAA	082311 23-5 hu001 to hu024	A1	Vande rKelen	8/23/11	23-5 ITS-F	23-5 ITS-R- biotin	23-5- ITS-S

Once *E. coli* has been isolated from a fecal sample, it will be analyzed using the QIAGEN Pyromark Q24 Pyrosequencer. This machine can analyze up to 24 samples at a time in individual wells labeled A1-A8, B1-B8 and C1-C8. Prior to pyrosequencing, each isolate is subject to a PCR reaction with either 23-5 or 16-23 forward and biotin-labeled reverse primers. These PCR products will then be subject to the pyrosequencing reaction. Isolates may or may not be pyrosequenced many times.

The Pyroprint Identification Information Table lists all the information about the pyrosequencing reaction of a particular isolate including the name of the isolate, which region is being analyzed, the dispensation sequence, the name of the final .xml file, which well the isolate was in, who performed the analysis and on what date, and all

primers used for the reaction. The unique set of parameters for a pyrosequencing reaction is given a Pyro ID number which links the user to the actual pyroprint generated for the given isolate on a given day.

Section 3: Pyroprint Data

At the end of a pyrosequencing reaction, the pyrosequencer returns an xml file of peak heights which is the “pyroprint” for a particular isolate. The pyroprint is not an actual sequence, but rather the “fingerprint” of the mix of the seven rDNA sequences in an *E. coli* isolate. This fingerprint is represented as a pattern of peak heights, and in the database it is stored as a histogram. It is accessible by clicking on the Pyro ID link associated with the pyrosequence identification information on the isolate of interest. The reference data for the selected isolate will appear in the intervening window; subsequently selecting “View Histogram” connects the user to the histogram labeled with the peak height values obtained by the pyrosequencer.

“Behind the scenes” the xml file data has been separated into information about the peak height and shape for each dispensed nucleotide (between 95 and 105 nucleotides); this data has been organized into the histogram/compensated slope table. This table is not available to the user, but is necessary for generating the histogram. It also contains information from the xml file that may be useful to us in the future.

PyroID	Position	Nucleotide	Peak Height	Compensated	Peak Height	Peak Width	Baseline Offset	Signal Value
1	1	A	.67	1.0	2.4	4.5	5.3	1.1
1	2	A
1	...	C						
1	100	G		Data Differs	By	Each	Nucleotide	
1	200	A						

Other Tables

The following reference tables are only accessible to Level 1 and 2 users (Admin and Data Editor). Additions to these tables are individually added directly into the database using the online software.

Host Species Table

This auxiliary table includes the common name of a host from which a fecal sample might be collected, its Latin name, and a two letter code.

Common Name	Latin Name	Letter Code
Cat	<i>Felis</i>	Ct

Dispensation Table

This table shows the order in which nucleotides are dispensed during a pyrosequencing reaction for a given dispensation type.

Dispensation Name	Sequence
ModGATC-2c	AACACGCGAGATCGATCGATCGATCGATCGATCGATCGATC

Primer Table

This table shows the DNA sequence of the primers used for the PCR and pyrosequencing reactions.

Primer Name	Sequence
23-5ITS-F	ATGAACCGTGAGGCTTAACCTT

Users Table

This table contains the user names and security information for logging into the CPLOP system. Only an administrator (Level 1 access) can add or delete users to the CPLOP database. This table and related security tables are not shown in this guide.

First Pyrosequence Date Table

This table documents the first date upon which an *E. coli* isolate was pyrosequenced in either the 23-5 or 16-23 region. This is a bookkeeping table that allows users to determine if an *E. coli* isolate has a pyroprint for both regions. Data from this table is listed in “See Missing Pyroprint Data” available to the Data Editor.

INSERTING DATA

The file uploader is a feature of the website that allows lab data (typically Excel files) to be uploaded into the database. The InsertData Link on the home page allows Level 1 and 2 Users to add data to the CPLOP database. There is a three part uploading system: Part 1 Upload Isolate Data, Part 2 Upload Pyroprint Identification Information, and Part 3 Upload Pyroprint Data.

Uploading the Isolates Table (Part 1 of 3 of the Uploader)

The Isolates, Host Data, and Samples Data Tables are populated by one file, the Freezer Stock Master Log. Prior to uploading, this file must be saved as a .csv file.

Once the proper .csv file is selected, the user clicks on the “Submit” link, and the file is uploaded.

Uploading the Pyroprint Identification Information Table (Part 2 of 3 of the Uploader)

The Pyroprint Identification Information data comes from an Excel file called the Pyroprint Master Log. Prior to uploading, this file must be saved as a .csv file. Once the proper .csv file is selected, the user clicks on the “Submit” link, and the file is uploaded.

Uploading the Pyroprint Data (Part 3 of 3 of the Uploader)

The Pyroprint Data comes from individual xml files generated by the QIAGEN Pyromark Q24 Pyrosequencer. In order for the data to be recognized by the uploader, the data must be “analyzed” by the Pyromark software. This is a function that compensates the raw peak height data and tries to assign a nucleotide sequence to the generated data. It also warns the user of any potential problems in the run like low peak heights, or dispensation errors. Once the raw data in the xml file has been analyzed and selected in the uploader, the user clicks on the “Submit” link, and the file is uploaded.

Using CPLOP's System

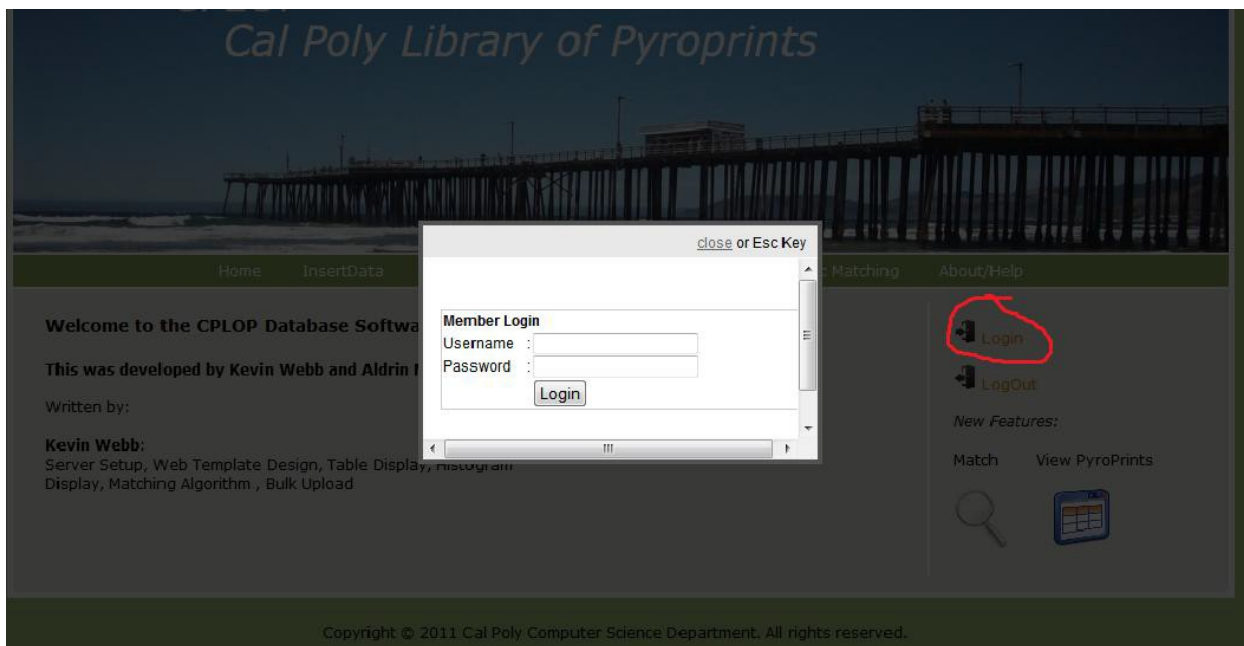
How to Log on

There are three types of users: guest (Level 3), data editor (Level 2), and administrator (Level1). A guest can view the Isolate and Pyroprint Identification Information tables, as well as link to Pyroprint data and the histograms through the PyroID. A data editor can view, edit, and add to all tables except the users table. The administrator can view, add, edit, and delete all tables. Both a data editor and administrator can use the extra features, such as viewing Host Data, Missing Pyroprint Data, etc.

A user can log in by accessing the main page using Internet Explorer or Firefox at

<http://cplop.cosam.calpoly.edu/>

The user will hit the “Log in Icon” and follow the steps shown in the pictures below.



**Figure 2 Step 1 of
Logging In**

Once logged in, a DataEntry user will see the DataEditor's portal button. An administrator will see both the DataEditor's and Admin Portals shown in the picture below.

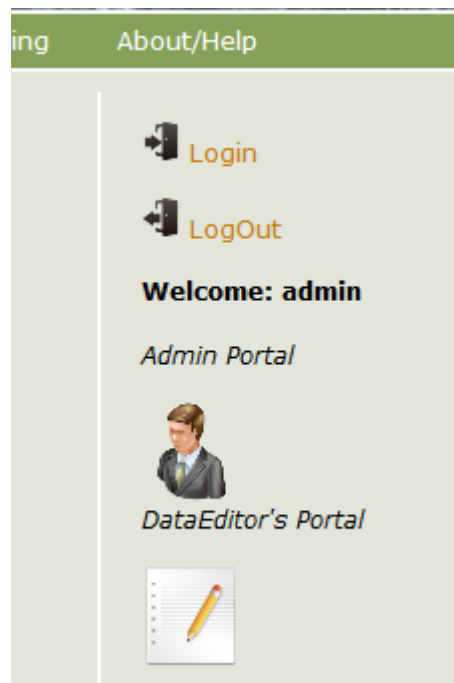


Figure 3 Levels of Login Access

Data Editor Portal

The DataEditor's portal provides access to the following:

- *Insert Data – Inserting Data for tables 1 – 6 thru the Bulk Upload Process.
- * Manage Host Species Data –View/ Add/Edit Data to the Specified Table
- * Manage Sample Data – View/Add/Edit Data to the Specified Table
- * Manage Primer Data –View/ Add/Edit Data to the Specified Table
- * Manage Sample Data – View/ Add/Edit Data to the Specified Table
- * Manage Dispensation Sequence Data – View/ Add/Edit Data to the Specified Table

Note that Administrators can also delete from the above tables.

- *Change My Password -- Changes the Password of the current user

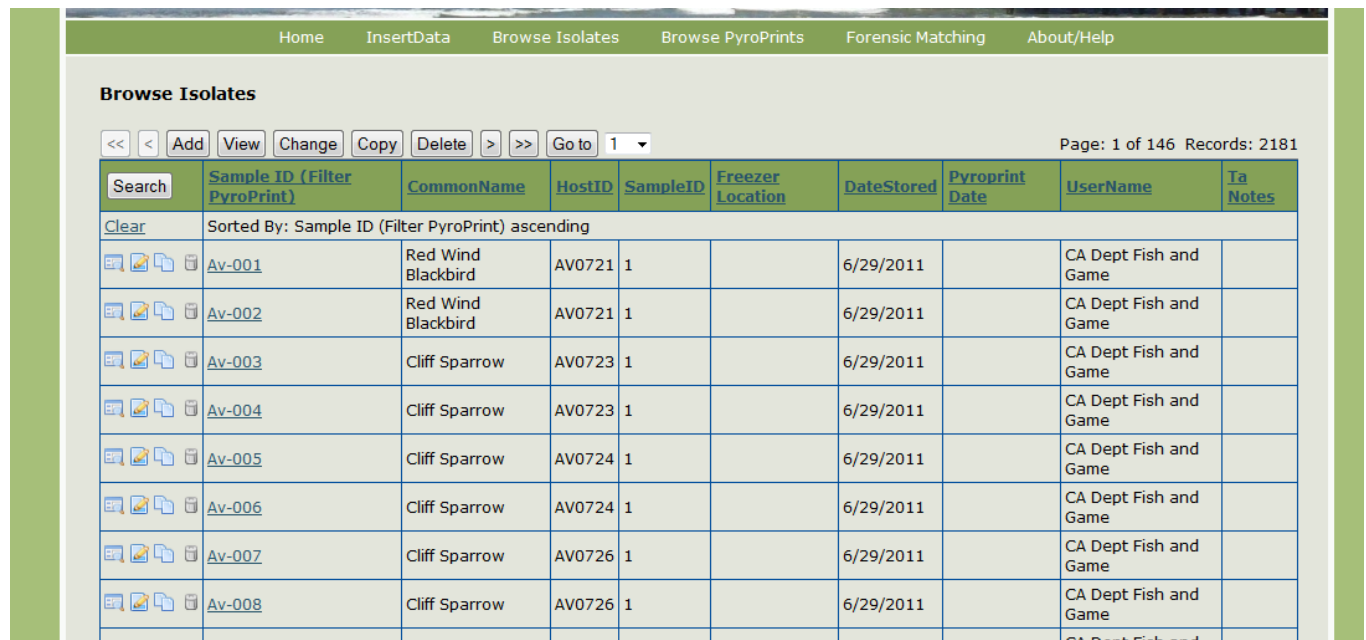
* Check Histogram State -- Tells the user how many entries are missing from the Histogram table.

Administrator Portal

Administrator users have access to modify (add/delete/view) from the user's table. They also can gain access to the backend database tools (phpMyAdmin).

Viewing CPLOP Data

Take for example the Isolate Table. A user may want to know what is in the freezer stock. They log in as a dataEntry user, hit the dataEditor portal button, and click on "manage Isolate table". The user will see the following interface.



The screenshot displays the 'Browse Isolates' interface. At the top is a navigation bar with links: Home, InsertData, Browse Isolates, Browse PyroPrints, Forensic Matching, and About/Help. Below the navigation bar, the title 'Browse Isolates' is followed by a set of action buttons: <<, <, Add, View, Change, Copy, Delete, >, >>, and a 'Go to' dropdown menu currently set to '1'. On the right, it says 'Page: 1 of 146 Records: 2181'. The main data is presented in a table with the following columns: Search, Sample ID (Filter PyroPrint), CommonName, HostID, SampleID, Freezer Location, DateStored, Pyroprint Date, UserName, and Ta Notes. The table is sorted by 'Sample ID (Filter PyroPrint) ascending'. The first eight rows of data are visible, showing isolates AV-001 through AV-008, all dated 6/29/2011, and associated with 'CA Dept Fish and Game'.

Search	Sample ID (Filter PyroPrint)	CommonName	HostID	SampleID	Freezer Location	DateStored	Pyroprint Date	UserName	Ta Notes
Clear	Sorted By: Sample ID (Filter PyroPrint) ascending								
	Av-001	Red Wind Blackbird	AV0721	1		6/29/2011		CA Dept Fish and Game	
	Av-002	Red Wind Blackbird	AV0721	1		6/29/2011		CA Dept Fish and Game	
	Av-003	Cliff Sparrow	AV0723	1		6/29/2011		CA Dept Fish and Game	
	Av-004	Cliff Sparrow	AV0723	1		6/29/2011		CA Dept Fish and Game	
	Av-005	Cliff Sparrow	AV0724	1		6/29/2011		CA Dept Fish and Game	
	Av-006	Cliff Sparrow	AV0724	1		6/29/2011		CA Dept Fish and Game	
	Av-007	Cliff Sparrow	AV0726	1		6/29/2011		CA Dept Fish and Game	
	Av-008	Cliff Sparrow	AV0726	1		6/29/2011		CA Dept Fish and Game	

Figure 4 Example of Viewing CPLOP's Table Data

A user can hit the search button, and enter into the text fields that discriminates entries based on that input. The following picture shows isolates being discriminated by the "common Name" and "date" columns.

Home
InsertData
Browse Isolates
Browse PyroPrints
Forensic Matching
About/Help

Browse Isolates

<<
<
Add
View
Change
Copy
Delete
>
>>
Go to
1

Page: 1 of 1 Records: 2

Hide	Sample ID (Filter PyroPrint)	CommonName	HostID	SampleID	Freezer Location	DateStored	Pyroprint Date	UserName	Ta Notes
Clear	Query	Blackbird				6/29/2011			
Sorted By: Sample ID (Filter PyroPrint) ascending									
Current Query: ('PMETable0'.commonName' LIKE "Blackbird") AND ('PMETable0'.dateStored' LIKE "6/29/2011")									
	Av-001	Red Wind Blackbird	AV0721	1		6/29/2011		CA Dept Fish and Game	
	Av-002	Red Wind Blackbird	AV0721	1		6/29/2011		CA Dept Fish and Game	

6.742 milliseconds

Copyright © 2011 Cal Poly Computer Science Department. All rights reserved.

Figure 5 A Search Example

The user can also click on the buttons on the lefts that edit/changes/modifies the individual entry.

View - Looks at a single entry horizontally.

Change - Allows user to edit any individual field of the data.

Delete - Allows user (admin) to delete an individual entry.

Home
InsertData
Browse Isolates
Browse PyroPrints
Forensic Matching
About/Help

Browse Isolates

Save
Apply
Cancel

Sample ID (Filter PyroPrint)	Av-001
CommonName	Red Wind Blackbird
HostID	AV0721
SampleID	1
Freezer Location	
DateStored	6/29/2011
Pyroprint Date	
UserName	CA Dept Fish and Game
Ta Notes	

1.995 milliseconds

Copyright © 2011 Cal Poly Computer Science Department. All rights reserved.

Figure 6 Editing an Example Isolate Entry

Special Table Features

All tables can use the same interface shown above to modify CPLOP's data. The isolate table has each Isolate ID as a hyper link to search the pyroprint table for all pyrprints that are linked with that particular Isolate ID. The pyroprint table's "pyroID" are hyperlinks to a gathering page that allows the user to perform special features (like displaying graph data) for that particular pyroID.

The following pictures show how the user can use the Isolate table to display all "Sg-027" isolates in the pyroprint table. We can also display Histogram data for all pyroprints linked to that isolate. This is shown in the pictures below.

Home InsertData Browse Isolates Browse PyroPrints Forensic Matching About/Help

Browse Isolates

<< < Add View Change Copy Delete > >> Go to 1 Page: 1 of 1 Records: 1

Hide	Sample ID (Filter PyroPrint)	CommonName	HostID	SampleID	Freezer Location	DateStored	Pyroprint Date	UserName	Ta Notes
Query	Sg-027								
Clear	Sorted By: Sample ID (Filter PyroPrint) ascending								
Clear	Current Query: ('PMetable0'. 'isoID' LIKE '*Sg-027*')								
	Sg-027	Seagull	6	1		8/12/2010		Vanessa Castillo	12/13/2010

4.211 milliseconds

Figure 7 Searching Isolates for SG-027 and Using Link Button

Home InsertData Browse Isolates Browse PyroPrints Forensic Matching About/Help

Browse Pyroprints

This table shows all Pyroprints linked to Isolate ID 'Sg-027'

<< < Add View Change Copy Delete > >> Go to 1 Page: 1 of 1 Records: 1

Search	PyroID (Options)	IsoID	AppliedRegion	DsName	ForPrimer	RevPrimer	SeqPrimer	FileName	WellID	PyroPrintedTech	PyroPrintedDate
Clear	Sorted By: PyroID (Options) ascending										
	240	Sg-027	23-5	ModGATC-2 contr	23-SITS-F	23-SITS- R-biotin	23-SITS-S	12/13/10 ModGATC-2 controls	B5	VanderKelen	12/13/2010

3.462 milliseconds

Copyright © 2011 Cal Poly Computer Science Department. All rights reserved.

Figure 8 Viewing All Pyroprints Linked by Isolate Sg-027 (Found PyroID 240)

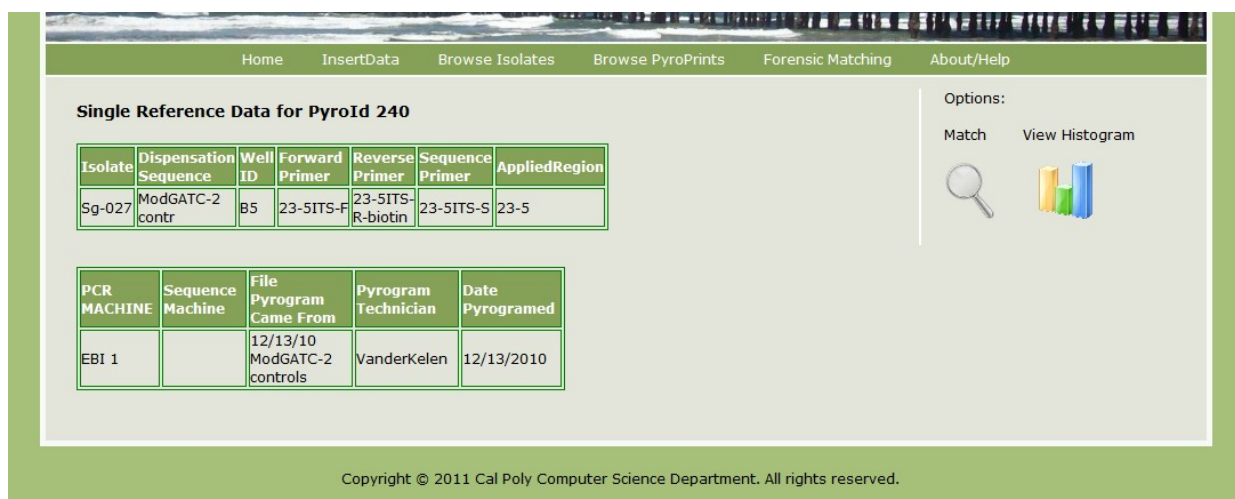
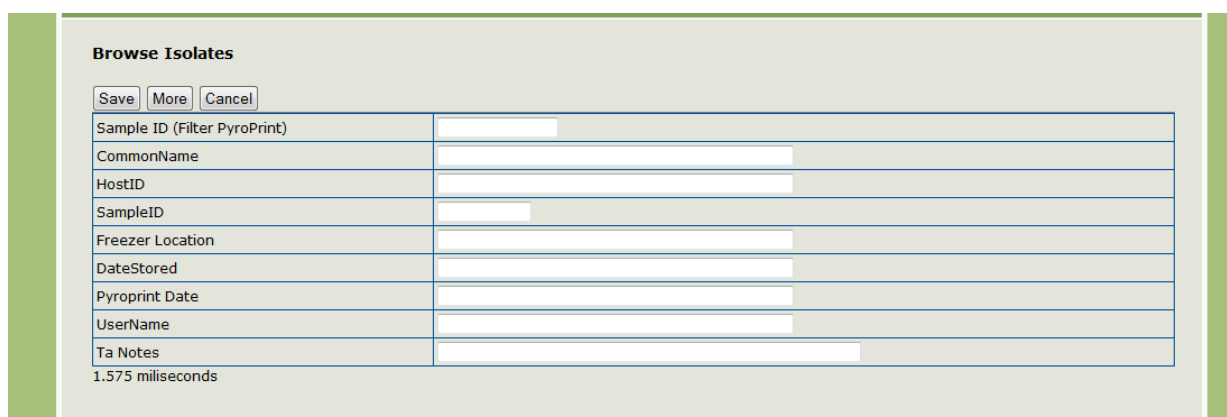


Figure 9 Pyroprint Features of PyroID 240

**Please note that the matching functionality will be completed by another student in the near future. My senior project focused on creating the database, upload functionality, and providing a base for future features. **

Uploading Data to CPLOP

Method 1) Manual Entry – Tables such as Primers or Users are not regularly added to. Therefore the user can use the datagrid viewing interface, shown above, to individually add an entry. This feature is shown below for the isolate table by clicking the “Add button” on the top of the table. By clicking the save button, the data entered will be saved to the database as another entry. If an entry is left blank, the table will show nothing in that field.



Browse Isolates

Save More Cancel

Sample ID (Filter PyroPrint)	
CommonName	
HostID	
SampleID	
Freezer Location	
DateStored	
Pyroprint Date	
UserName	
Ta Notes	

1.575 milliseconds

Method 2) Bulk Upload—The majority of the data will be held in the isolate, pyroprint, and histogram tables. Both the isolate and pyroprint tables are added via master excel sheets found on the lab computer which is discussed earlier in this document. The interface and web address for this feature is shown below.

<http://cplop.cosam.calpoly.edu/insertdata.php> (Login Required)

The following pictures serve as examples of the three step bulk uploading process:

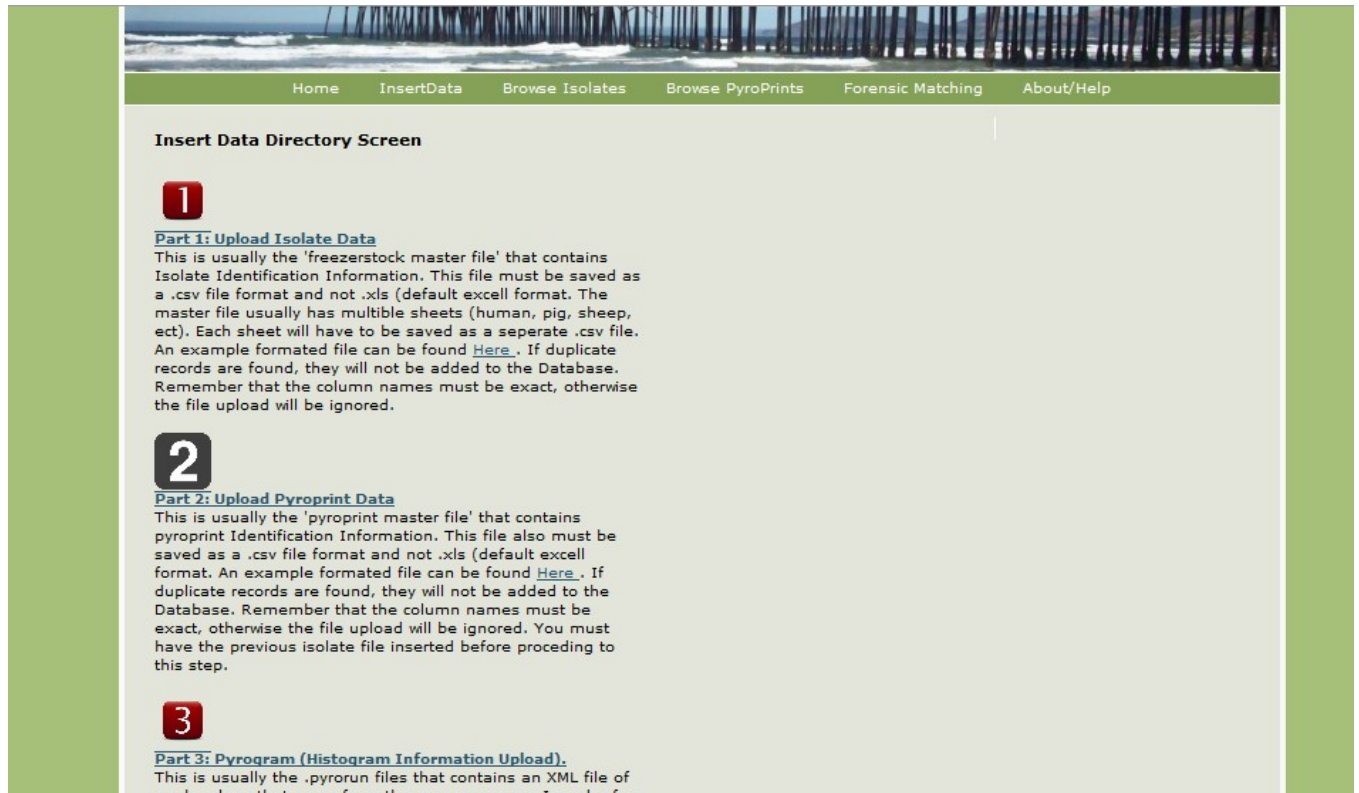


Figure 10 Main Screen of 3 Step Bulk Uploader Process

Upload Isolate Information (.csv)

Name

Your email address

Choose a file to upload

Browse...

Submit

Figure 11 Example Step One's File Uploader

Part 2: A Programmer's Guide

This part is geared toward a programmer that already has php, html, and mysql experience. This guide is to give a head start about the structure of the CPLOP system so a programmer could add the additional functionality. This guide also assumes the programmer read part one of this guide so he/she understands the database structure and how the tables are linked.

Overview of System

CPLOP is sitting on a virtual machine on the College of Math and Science Server. This is a Ubuntu server (Version 10) running php 5 and mysql. The php.ini file is located at /etc/apache2/ and the entire website is sitting at /var/cplop/. A person can "ssh" into the machine at cplop.cosam.calpoly.edu .

The server's administrator is Jon Sehmer who can be contacted at: jsehmer@calpoly.edu .

Each .php file represents a certain web page. Index.php is the page's home address. The upload folder is a temporary storage location for the uploaded file of the bulk uploading process. The file "functions.php" provide the programmer with functions that automatically connect to the server's mysql database. (The mysql username//pass are in this file). In addition, it provides tools such as "isAdmin() or isDataEditor()", when combined with the session_start() can tell if the user has logged in or not. A session lasts for 30 minutes. After that, a person is automatically logged out. All of the tables above are in the CPLOP database. The user table is in the User database. A user can be of three levels. Level 1 is admin. Level 2 is dataEditor (can view/edit/add data). Level 3 is guest - anybody who is not logged in.

Example of a Blank Page (Where to Start with Adding Functionality)

The following code segment represents a blank typical page of CPLOP, using the default html template. It also shows how the programmer can check the user's credentials and parse data from the mysql database. If I were to begin adding the matching functionality, I would start editing the match.php file with the following code.

//Example Match.php (Blank Template)

<?session_start(); //Start the Session ?>

<!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">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

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

<title>Cal Poly Library of Pyroprints</title>

</head>

<script type="text/javascript" src="/cplop/ThickBox/jquery-latest.js"></script>

<script type="text/javascript" src="/cplop/ThickBox/thickbox.js"></script>

<link rel="stylesheet" href="/cplop/ThickBox/thickbox.css" type="text/css"
media="screen" />

<body>

<div id="container">

<div id="header">

<div id="logo_w1">CPLOP</div>

<div id="logo_w2">Cal Poly Library of Pyroprints </div>

**<?php include("functions.php"); webSiteLinks(); //Prints out
theLinks ?>**

</div>

<div id="content">

<div id="left">

```
<?php
```

```
//SECTION TO DO YOUR WORK
```

```
$isAdmin = isAdmin(); //Checks if User is Admin (Called from Functions.php  
above)
```

```
if(!$isAdmin)
```

```
{
```

```
    echo("<H1>You do have a high enough access level to modify the CPLOP  
system or have not logged in. <br> Access Error 02 <H1><br>Log in: <a  
href='\"http://cplop.cosam.calpoly.edu/#TB_inline?height=155&width=300&inlineI  
d=hidden Content\"' class='\"thickbox\"'><p><img src='\"images/login.png\"'  
width='\"20\"' height='\"20\"' alt='\"Log In !\"' /></div>");
```

```
}
```

```
else
```

```
{
```

```
    //Left Section of the Page
```

```
    echo("<H1> Write on the Left Side of the Page</div>");
```

```
    //Right Section of the Page
```

```
    echo("<div id='\"right\"'> Write on Right Side of Page</div>");
```

```
}
```

```
?>
```

```
    <div id="footerline"></div>
```

```
</div> <!-- end content -->
```

```
    <div id="footer">Copyright &copy; 2011 Cal Poly Computer Science Department.  
All rights reserved.</div>
```

```
</div> <!-- end container -->
```

```
</body>
```

</html>

In the php work section, a user could use the following code to call a query to the MySql DataBase.

//Start Code

<?php

//Connect to CPLOP DB

mysqlConnect("CPLOP");

\$result = mysql_query("Select count(*) as COUNT from Pyroprints");

while(\$info = mysql_fetch_array(\$result))

{

 \$returnedNum =
 \$info['COUNT'];

}

?>

=====

Included Functions in the Functions.php (Common API)

function mysqlConnect(\$tableName)

Connects to User or CPLOP depending on \$tablename. This will allow all calls to mysql_query to work.

isDataEditor()

Checks User Table if user has level 2 or greater access. True if so, false

if not so. isAdmin()

Checks User Table if user has level 1 or greater access. True if so, false if not so.

Void function webSiteLinks()

Lists (in format) the header links of the website. If a change is needed, for example adding another button next to InsertData, a field could be added. This function is included in any page that uses CPLOP's standard template.

function connectToDataBaseTableSoftware()

CPLOP uses a php script to generate the table software used on all of its pages. This script requires an array to connect with a user Name and password. This script fills these arrays with the correct mysql login information.

Int function pearson(\$sequenceOne,
\$sequenceTwo)

CPLOP has a rudimentary match system. This feature needs to calculate the pearson's coefficient between two sets of similarly sized arrays. The return is between -1 and 1.

PME Table Package

A table script was used <http://www.phpmyedit.org/>. You can download its .php package, run it in a browser and it will generate all php code needed to display the tables live. The code it generates is the values the script needs to display one table. Examples can be found on browsisolates.php or browsePyroprints.php.

Bulk Uploader

Each step of the bulk uploader executes a page (upload_1.php, upload_2.php, or upload_3.php) respectfully.

Upload_1 Parses .CSV File and uploads data into Host, Samples, and Isolate Table

Upload_2 Parses .CSV File and uploads data into Pyroprint Table. This creates a unique pyroID for each entry.

Upload_3 Parses XML File (.PYRORUN) and dumps data into compensated_slope and Histogram table. Each entry is linked to a pyroprint by the well and filename.

This

function calls Aldrin Montana's xml parser package, xmlparser.php .

PHPMYAdmin (DataBase view of Database//Create Statements)

PHPMYAdmin has been installed for the convenience of the programmer. The standard mysql username and password will grant you access to cplop.cosam.calpoly.edu/phpMYAdmin. A link can also be found in the admin portal section. This tool allows the user to back up the databases, perform complicated sql queries, and check on the current state of the database.

The following includes the CREATE_TABLE statements of SQL needed to Create the CPLOP DataBase.

```
--
-- Database: `CPLOP`
--

-- -----

--
-- Table structure for table `compensation_slope`
--

CREATE TABLE IF NOT EXISTS `compensation_slope` (
  `pyrogram_num` decimal(50,0) NOT NULL,
  `position` decimal(50,0) NOT NULL,
  `level` decimal(10,0) NOT NULL,
  `drop_off_value` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----

--
-- Table structure for table `Dispensation`
--

CREATE TABLE IF NOT EXISTS `Dispensation` (
  `dsName` varchar(15) NOT NULL DEFAULT '',
  `dispSeq` varchar(200) DEFAULT NULL,
  PRIMARY KEY (`dsName`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----

--
-- Table structure for table `Histograms`
--

CREATE TABLE IF NOT EXISTS `Histograms` (
  `pyroID` int(11) NOT NULL DEFAULT '0',
  `position` int(11) NOT NULL DEFAULT '0',
  `pHeight` decimal(8,4) DEFAULT NULL,
  `cPeakHeight` decimal(8,4) DEFAULT NULL,
  `nucleotide` char(1) DEFAULT NULL,
  `PeakArea` float NOT NULL,
  `PeakWidth` float NOT NULL,
```

```

    `BaselineOffset` float NOT NULL,
    `SignalValue` float NOT NULL,
    PRIMARY KEY (`pyroID`, `position`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----

--
-- Table structure for table `Host`
--

CREATE TABLE IF NOT EXISTS `Host` (
  `commonName` varchar(50) NOT NULL,
  `hostID` varchar(50) NOT NULL DEFAULT '',
  PRIMARY KEY (`commonName`, `hostID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----

--
-- Table structure for table `HostSpecies`
--

CREATE TABLE IF NOT EXISTS `HostSpecies` (
  `latinName` varchar(50) DEFAULT NULL,
  `commonName` varchar(50) NOT NULL,
  `letterCode` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`commonName`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----

--
-- Table structure for table `Isolates`
--

CREATE TABLE IF NOT EXISTS `Isolates` (
  `isoID` varchar(15) NOT NULL DEFAULT '',
  `FreezerLocation` varchar(50) DEFAULT NULL,
  `TANOTES` varchar(300) DEFAULT NULL,
  `dateStored` varchar(15) DEFAULT NULL,
  `PyroPrintDate` varchar(50) DEFAULT NULL,
  `userName` varchar(50) DEFAULT NULL,
  `sampleID` int(11) DEFAULT NULL,
  `commonName` varchar(50) NOT NULL,
  `hostID` varchar(50) NOT NULL,
  PRIMARY KEY (`isoID`),
  KEY `commonName` (`commonName`),
  KEY `hostID` (`hostID`, `commonName`),
  KEY `sampleID` (`sampleID`, `commonName`, `hostID`),
  KEY `userName` (`userName`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

-- -----
--
-- Table structure for table `Primer`
--

CREATE TABLE IF NOT EXISTS `Primer` (
  `primerName` varchar(40) NOT NULL DEFAULT '',
  `sequence` varchar(40) DEFAULT NULL,
  PRIMARY KEY (`primerName`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----
--
-- Table structure for table `Pyroprints`
--

CREATE TABLE IF NOT EXISTS `Pyroprints` (
  `pyroID` int(11) NOT NULL DEFAULT '0',
  `fileName` varchar(100) DEFAULT NULL,
  `appliedRegion` varchar(20) DEFAULT NULL,
  `pcrDate` varchar(15) DEFAULT NULL,
  `wellID` varchar(5) DEFAULT NULL,
  `pcrMachine` varchar(20) DEFAULT NULL,
  `sequencerMachine` varchar(20) DEFAULT NULL,
  `dsName` varchar(15) DEFAULT NULL,
  `forPrimer` varchar(40) DEFAULT NULL,
  `revPrimer` varchar(40) DEFAULT NULL,
  `seqPrimer` varchar(40) DEFAULT NULL,
  `pyroPrintedTech` varchar(100) DEFAULT NULL,
  `pyroPrintedDate` varchar(15) DEFAULT NULL,
  `isoID` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`pyroID`),
  KEY `dsName` (`dsName`),
  KEY `forPrimer` (`forPrimer`),
  KEY `revPrimer` (`revPrimer`),
  KEY `seqPrimer` (`seqPrimer`),
  KEY `pyroPrintedTech` (`pyroPrintedTech`),
  KEY `isoID` (`isoID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```



```
-- -----  
  
--  
-- Table structure for table `Samples`  
--  
  
CREATE TABLE IF NOT EXISTS `Samples` (  
  `sampleID` int(11) NOT NULL DEFAULT '0',  
  `commonName` varchar(50) NOT NULL,  
  `hostID` varchar(50) NOT NULL,  
  `location` varchar(50) DEFAULT NULL,  
  `dateCollected` varchar(15) DEFAULT NULL,  
  PRIMARY KEY (`sampleID`,`commonName`,`hostID`),  
  KEY `hostID` (`hostID`,`commonName`),  
  KEY `commonName` (`commonName`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;  
  
-- -----
```

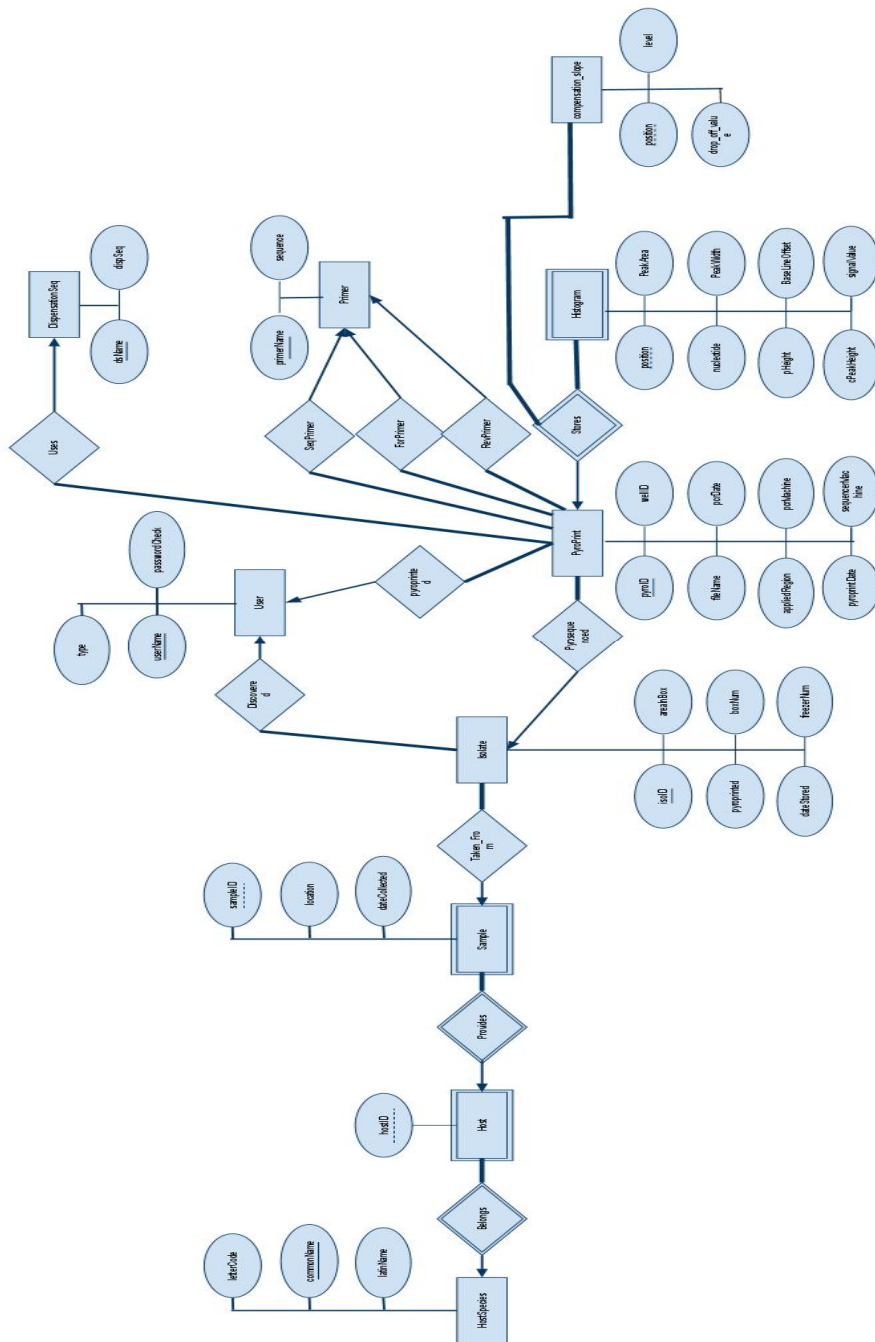
The following includes the CREATE_TABLE statements of SQL needed to Create the "Users"

DataBase.

```
--  
-- Database: `Users`  
--  
  
-- -----  
  
--  
-- Table structure for table `members`  
--  
  
CREATE TABLE IF NOT EXISTS `members` (  
  `id` int(4) NOT NULL AUTO_INCREMENT,  
  `username` varchar(65) NOT NULL DEFAULT '',  
  `password` varchar(65) NOT NULL DEFAULT '',  
  `type` int(4) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
```

The SQL Code has to be executed in the order above to perverse foreign and primary key constraints. Two DB's are needed, CPLOP and Users as shown above.

ER Diagram For CPLOP Database



Summary:

CPLOP is now at a functional state to upload, store, and display Pyroprint data to Cal Poly's Biology Department. Instead of using multiple excel files stored across many computers and folders, any student can access the data from any computer anywhere. My senior project's goals were to start the CPLOP system and provide a common base for future development. All of these goals have been met, and the result is a usable database content delivery system.

Analysis of Senior Project Design

Please provide the following information regarding your Senior Project and submit to your advisor along with your final report. Attach additional sheets for your responses to the questions below.

Project Title: Cal Poly Library of Pyroprints
Quarter / Year Submitted: Fall 2011

Student: (Print Name) Kevin Webb (Sign) _____

Advisor: (Print Name) Alex Dekhtyar (Initial) _____ Date: _____

• Summary of Functional Requirements

Describe the overall capabilities of functions of your project or design. Describe what your project does. (Do *not* describe how you designed it.)

Answer: CPLOP is a mysql/php web-driven database application that meets the needs of Cal Poly's Biology Pyrosequencing lab to store pyrosequencing DNA data. The system is a base to provide storage, searching, and matching needs for this data.

• Primary Constraints

Describe significant challenges or difficulties associated with your project or implementation. For example, what were limiting factors or other issues that impacted your approach? What made your project difficult? What parameters or specifications limited your options or directed your approach?

Answer: A major limiting component in the early design was access to a suitable Unix server to host the web application and MySQL database. The underlying operating system poses unique challenges to the possible scripts and extensions that could be added to such a web application. Early servers were limited by the amount of "burst ram" creating unnecessary bottlenecks for end users as they were launched on a virtual machine. Other difficulties included limited Summer contact with group members on both the CSC and Biology side.

• Economic

- Original estimated cost of component parts (as of the start of your project)

Answer: No estimate was needed as this was a "software" code deliverable that started out of CPE366 (Database Modeling and Design).

- Actual final cost of component parts (at the end of your project)

Answer: In reality, CPLOP is currently running on a College of Science and Math Server. All bandwidth and server costs associated to this project are limited to those arenas. Server budgeting information was not provided to me.

- *Attach a final bill of materials for all components*
- Additional equipment costs (any equipment needed for development?)

Answer: N/A See Above

- Original estimated development time (as of the start of your project)
The project was broken into CPE 366 class time, Summer 2011, and Fall 2011 extra time. I estimated between 200-300 hours were spent to bring the project to version 1.0.

- Actual development time (at the end of your project)
Near 300 hours over the course of three quarters.

• **If manufactured on a commercial basis:**

- Estimated number of devices to be sold per year
- Estimated manufacturing cost for each device
- Estimated purchase price for each device
- Estimated profit per year
- Estimated cost for user to operate device, per unit time (specify time interval)

Answer: The code is limited to a single server for non-commercial yet academic use at Cal Poly. Cal Poly does not need to purchase any license from me to use this software. This software is very specific to our internal lab and can not be resold for other use. Therefore, the product has no real market value.

• **Environmental**

Describe any environmental impact associated with manufacturing or use.

Answer: The additional impact should be negligible. The server the web application is running on has many additional applications it supports to students and faculty. My added application may run additional cycles on the CPU, but that system was already turned on in the first place. The environmental cost will be limited to amount of “current” the server draws for this one application and any additional hardware changes in the future that is needed.

• **Manufacturability**

Describe any issues or challenges associated with manufacturing.

Answer: Manufacturing was limited to software development time. As software is “non material”, it is hard to describe a manufacturing process. However, CPLOP was designed in steps. Step one was gathering the requirements from the Biology department. Step two was modeling an E-R database diagram around the needs of the user. Step three was selecting a proper server for this application, and then actually implementing a web application around the database. The last step involved testing.

• **Sustainability**

- Describe any issues or challenges associated with maintaining the completed device or system.

Answer: The database software can be added to and will require the server administrator to grant access to other programmers in the near future. Also programmer intervention will be required to completely backup or restore the database in case of failure.

- Describe how the project impacts the sustainable use of resources.

Answer: Any environmental impact of this web application was described above.

- Describe any upgrades that would improve the design of the project.

Answer: Ideally this web-application would include full matching functionality at the beginning of the production cycle. I would have bought/implemented automatic backup tools on the database for data recovery failures.

- Describe any issues or challenges associated with upgrading the design.

Answer: An upgrade, like adding matching capabilities, would involve getting another php programmer up to date with the database schema in addition to the code structure of the website. The database scheme will be hard to change with production data already in use. Development will have to be limited to the first VM server (cslvm96.csc.calpoly.edu).

• **Ethical**

Describe ethical implications relating to the design, manufacture, use or misuse of the project.

Answer: End-users will have the ability to edit certain sections of data in the database. If used maliciously, the data could be corrupted and become useless. It will be the job of system wide administrators to grant access (levels 1 – 3) to select users. Ethically the data is used to determine unknown bacteria in beach samples. Injected data could tamper those results.

• **Health and Safety**

Answer: End user's will have to consult their PC manufactures for health concerns using their monitors when viewing a web browser.

• **Social and Political**

Describe any social and political concerns associated with design, manufacture or use.

Answer: This project is limited to Cal Poly's academic community. The matching results could be used to determine unknown contamination on beaches. Any public impact, such as a beach closure, could be used from the conclusion

• **Development**

Describe any Development Tools Used

Answer: This project uses MySQL as a relational database using SQL to trigger commands. The web server renders html pages using apache. Each page is dynamically controlled and written by PHP 5.0 with a host of add on extensions.