



Field Session 2005 - Final Report

Jonathan Nahil
Jonathan Petty
Jonathon Van Cleve

June 22, 2005

Abstract

The Colorado Performance Academy is developing a website to assist personal trainers with managing all aspects of trainees' regiments. The current prototype site has minimum functionality and needs major graphical modifications as well as a new database schema before release. The site must be capable of tracking an individual's workout components and status, and adjusting workouts according to trainees' performance. Administrative functionality must be built in to allow trainers to manage exercise lists, exercise areas, and other attributes.

The prototype site is currently using PHP and MySQL, and we will continue development using these technologies. In order to avoid excessive difficulty, the project's scope will be limited to a single platform rather than trying to construct a universal cross-platform solution. The final project will be a fully functional and installable website, and will be delivered on June 24.

Contents

1.0	Project Requirements	4
1.0.1	Non-Functional Requirements	4
1.0.2	Functional Requirements	4
1.0.3	Programming Language Requirements	5
1.0.4	Software Requirements	5
2.0	Design	6
2.0.1	Security	6
2.0.2	Webpage Use Cases	7
2.0.3	Database	14
2.0.4	Common Function Libraries	15
2.0.5	Third Party Software	15
2.0.6	Project Development Flow	16
2.0.7	Estimated Project Time	17
3.0	Implementation	19
3.0.1	PHP Pages	19
3.0.2	Database	20
4.0	Debug and Testing	21
5.0	Delivery	21
6.0	Project Conclusion	21
7.0	Appendix	22

List of Tables

2.1	Security Groups and Their Associated Web Pages	6
2.2	Estimated Webpage Development Time	17
2.3	Estimated Database Development Time	17
2.4	Estimated Common Function Library Development Time	18
2.5	Estimated Total Project Development Time	18
3.6	Physical Webpages	19
3.7	Database Tables	20

List of Figures

2.1	Project Design Flow	16
7.2	Database Schema	23

1.0 Project Requirements

Our client, Jerry Howard at the Colorado Performance Academy, has started coding a website that is designed to be used as a tool by personal fitness trainers to assign workouts to their clients. He has asked us to complete and add functionality to his existing project. This site must track the client's progress in completing a workout, and must scale the workout intensity automatically as the personal trainers see fit.

Jerry Howard's current webpage prototype uses PHP to change and read information in a MySQL database, then display the data in an HTML web format. This prototype will be heavily modified and expanded during the course of the project.

The database schema must be redesigned to incorporate new features such as team lists, workout history, and site security. Administration of body parts, exercises, workouts, clients, and team rosters must be developed and access to these pages must be restricted to trainers and site administrators. Trainers must be able to add and edit clients, assign those clients to teams, and manage workouts for all their clients. Clients must be able to view their past and existing workouts, and update their current workout results. Workout intensity must be adjusted automatically depending on consecutive successes or failures of clients to meet workout goals.

The following lists summarize the functional and non-functional requirements of the project:

1.0.1 Non-Functional Requirements

- Security must be added to the entire site, with a login page before any other pages can be viewed
- Must allow personal trainers to define clients, teams, body parts, exercises, and workouts
- Must allow personal trainers to assign workouts to their individual clients or teams of clients
- Must keep a history of exercises completed by the clients, and adjust workouts according to performance
- Must store personal, medical and emergency contact information for clients

1.0.2 Functional Requirements

- Web site must interact with MySQL to insert, modify, delete, and view data

- Web site must format data and display it in an html format
- Web site must track data posted by any forms and update the database accordingly
- Database schema must be modified to accommodate all new site functionality
- Security model must include three main groups: site administrators, trainers, and clients (descending access order)
- Members of a security group must have access to all functionality allowed at their group level and below
- Administration screens must be security-based and built to add/edit/delete trainers, clients, teams, body parts, exercises, and workouts
- Trainers must be able to assign workouts to individual clients or teams. Assigning a workout to a team adds that workout to each client on the team
- Clients must have access to all workout history, including inactive workouts
- Exercises must have configurable parameters to determine how that individual exercise will react to client performance
- Upon entering workout results, the site must calculate a new workout intensity based on consecutive completions or failures of the workout

The following languages and technologies will be used to complete this project:

1.0.3 Programming Language Requirements

- html
- PHP
- CSS
- JavaScript

1.0.4 Software Requirements

- MySQL
- Apache

2.0 Design

2.0.1 Security

The entire website security will be based around three supergroups that have access to webpages on their level and below. The lowest level will be clients, who can only access their personal information and workouts. Coaches will be one level higher, having access to all of the individual client pages and additional pages that allow them to design workouts and team rosters. The highest security level will be Administrators, who will have access to all site management functionality and all coach and client pages. In addition to sites granted to a user on their security level and below, everyone will be able to open the Login/Home Page and Help Pages. The security groups and their individual pages are shown in table 2.1 below.

Table 2.1: Security Groups and Their Associated Web Pages

Security Group Name	Webpages On Security Level
Administration	Create/Modify/Deactivate User Page* Reactivate User Page Create/Modify/Delete Body Category Page Create/Modify/Delete Exercise Page Create/Modify/Delete Exercise Rule Create/Modify/Delete Exercise Surface Create/Modify/Delete Client Attribute Create/Modify/Delete Health Attribute Page Reset Password Page Site Management Page
Coach	Create/Modify/Delete Workout Page Assign Workout Page Create/Modify/Delete Team Roster Page Create/Modify/Delete Client Page* Body Composition Analysis Page Flexibility and Postural Analysis Page
Client	View/Change Personal Information Page* View Printable Workout Page Enter Workout Result Page Workout History Page Change Password Page
Public	Login/Home Page Help Pages

* See Create/Modify/Deactivate User Page. Same physical page, but security level determines available functionality

2.0.2 Webpage Use Cases

Create/Modify/Deactivate User Page

- Precondition: User has selected create new user
- Precondition: User is an administrator or a coach
- Precondition: User has security clearance to create new specified user
 - A form requiring personal information is displayed
 - Precondition: User has entered valid information
 - * A form requiring health information is displayed
 - * Precondition: User has entered valid information
 - New user is added to the database
 - * Precondition: User has not entered valid information
 - Health information form is redisplayed
 - Precondition: User has not entered valid information
 - * Personal information form is redisplayed
- Precondition: User has selected edit user
- Precondition: User has security clearance to edit the specified user
 - Precondition: User has selected 'Edit Personal'
 - * A pre-populated form is displayed for the user to edit personal information
 - * Precondition: User has posted valid changes
 - The database is updated with the changes
 - Precondition: User has selected 'Edit Health'
 - * A pre-populated form is displayed for the user to edit health information
 - * Precondition: User has posted valid changes
 - The database is updated with the changes
 - Precondition: User has selected 'Edit Attributes'
 - * A pre-populated form is displayed for the user to edit attribute information
 - * Precondition: User has posted valid changes
 - The database is updated with the changes
 - Precondition: User has selected 'Deactivate User'
 - * Selected user is marked inactive in the USERS table

Reactivate User Page

- Precondition: User is an administrator
 - A list of all inactive users are displayed in a list
 - The selected user is marked active in the USERS table

Create/Modify/Delete Body Category Page

- Precondition: User is an administrator
 - A list of all body categories is displayed with options to modify, delete, or create new body category
 - Precondition: User chooses 'Create New Body Category'
 - * A form requiring body categories information is displayed
 - * New body categories information is stored in the BODY_CATEGORIES table
 - Precondition: User chooses 'Modify Body Category'
 - * A pre-populated form displaying the body categories information is displayed
 - * Changes are stored in BODY_CATEGORIES table
 - Precondition: User chooses 'Delete Body Category'
 - * Selected body category is deleted if not used anywhere

Create/Modify/Delete Exercise Page

- Precondition: User is an administrator
 - A list of all exercises is displayed with options to modify, delete, or create new exercise
 - Precondition: User chooses 'Create New Exercise'
 - * A form requiring exercise information is displayed
 - * New exercise information is stored in the EXERCISES table
 - Precondition: User chooses 'Modify Exercise'
 - * A pre-populated form displaying the exercise information is displayed
 - * Changes are stored in EXERCISES table
 - Precondition: User chooses 'Delete Exercise'
 - * Selected exercise is deleted if not used by any workouts

Create/Modify/Delete Exercise Rule Page

- Precondition: User is an administrator
 - A list of all exercise rules is displayed with options to modify, delete, or create new exercise rule
 - Precondition: User chooses 'Create New Exercise Rule'
 - * A form requiring exercise rule information is displayed
 - * New exercise rule information is stored in the EXERCISE_RULES table
 - Precondition: User chooses 'Modify Exercise Rule'
 - * A pre-populated form displaying the exercise rule information is displayed
 - * Changes are stored in EXERCISE_RULES table
 - Precondition: User chooses 'Delete Exercise Rule'
 - * Selected exercise rule is deleted if not used anywhere

Create/Modify/Delete Exercise Surface Page

- Precondition: User is an administrator
 - A list of all exercise surfaces is displayed with options to modify, delete, or create new exercise surface
 - Precondition: User chooses 'Create New Exercise Surface'
 - * A form requiring exercise surface information is displayed
 - * New exercise surface information is stored in the EXERCISE_SURFACES table
 - Precondition: User chooses 'Modify Exercise Surface'
 - * A pre-populated form displaying the exercise surface information is displayed
 - * Changes are stored in EXERCISE_SURFACES table
 - Precondition: User chooses 'Delete Exercise Surface'
 - * Selected exercise surface is deleted if not used anywhere

Create/Modify/Delete Client Attribute Page

- Precondition: User is an administrator
 - A list of all client attributes is displayed with options to modify, delete, or create new client attributes
 - Precondition: User chooses 'Create New Client Attribute'
 - * A form requiring client attributes information is displayed

- * New client attributes information is stored in the CLIENT_ATTRIBUTES table
- Precondition: User chooses 'Modify Client Attribute'
 - * A pre-populated form displaying the client attributes information is displayed
 - * Changes are stored in CLIENT_ATTRIBUTES table
- Precondition: User chooses 'Delete Client Attribute'
 - * Selected client attribute is deleted if not used anywhere
 - * Selected client attribute is marked inactive if used somewhere
- Precondition: User chooses 'Reactivate Client Attribute'
 - * Selected client attribute is marked active in the CLIENT_ATTRIBUTES table

Create/Modify/Delete Health Attribute Page

- Precondition: User is an administrator
 - A list of all health attributes is displayed with options to modify, delete, or create new health attributes
 - Precondition: User chooses 'Create New Health Attribute'
 - * A form requiring health attributes information is displayed
 - * New health attributes information is stored in the HEALTH_ATTRIBUTES table
 - Precondition: User chooses 'Modify Health Attribute'
 - * A pre-populated form displaying the health attributes information is displayed
 - * Changes are stored in HEALTH_ATTRIBUTES table
 - Precondition: User chooses 'Delete Health Attribute'
 - * Selected health attribute is deleted if not used anywhere
 - * Selected health attribute is marked inactive if used somewhere
 - Precondition: User chooses 'Reactivate Health Attribute'
 - * Selected health attribute is marked active in the HEALTH_ATTRIBUTES table

Reset Password Page

- Precondition: User is an administrator
 - A list of all users is displayed in a form
 - The selected user's password is changed to some default value

Site Management Page

- Precondition: User is an administrator
 - A form with inputs to change the site logo and settings related to recalculating workout intensity is displayed
 - Note: This page will be used to change additional site parameters in the future if the need arises

Create/Modify/Delete Workout Page

- Precondition: User is a coach or an administrator
- Precondition: User has selected 'Create Workout'
 - A form requiring the workout name and exercises in the workout is displayed
 - The workout form can be blank or populated based on another active workout
 - The entered workout named is recorded in the WORKOUT table
 - The exercises in the workout are recorded in the WORKOUT_EXERCISES table
- Precondition: User is a coach or an administrator
- Precondition: User has selected 'Modify Workout'
 - A form displaying a list of all exercises in the workout is generated
 - An option to modify the workout for a specific client or all clients is given
 - Precondition: User chooses to modify all clients
 - * The workout is deleted from the USER_WORKOUT_EXERCISES table
 - * A new workout with the changed information is saved to the USER_WORKOUT_EXERCISES table
 - * The new workout is reassigned to all clients who had the original workout
 - Precondition: User chooses to modify one client
 - * Workout changes are saved to USER_WORKOUT_EXERCISES table
- Precondition: User is a coach or an administrator
- Precondition: User has selected 'Delete Workout'
 - A form displaying a list of active workouts tied to the user is generated

- An option to delete the workout from a specific client or all clients is given
- Precondition: User chooses to delete from a specific client
 - * Workout is marked inactive in USER_WORKOUT
- Precondition: User chooses to delete from all clients
 - * Workout is marked inactive in WORKOUT table
 - * Workout is marked inactive for all clients in USER_WORKOUT table is they are assigned to the workout

Assign Workout Page

- Precondition: User is a coach or an administrator
 - A form listing all athletes and rosters available to the user is listed
 - The workout is assigned to the selected athlete/roster by storing workout details in the USER_WORKOUT and USER_WORKOUT_EXERCISES tables

Create/Modify/Delete Team Roster Page

- Precondition: User is a coach or an administrator
 - A list of all available rosters to the user is displayed with options to modify, delete, or create new roster
 - Precondition: User chooses 'Create New Roster'
 - * A form displaying all clients available to the user is generated
 - * Selected clients are added to the roster and saved in the ROSTERS and ROSTERS_USERS tables
 - Precondition: User chooses 'Modify Roster'
 - * A pre-populated form displaying clients in the roster is generated
 - * A form displaying all clients available to the user is generated
 - * Roster changes are saved to the ROSTERS and ROSTERS_USERS tables
 - Precondition: User chooses 'Delete Roster'
 - * Roster is removed from ROSTERS and ROSTERS_USERS tables

Body Composition Analysis Page

- Precondition: User is a coach or an administrator
 - Input form for entry of body composition elements is displayed
 - Entered data is saved to BODY_COMPOSITION_ANALYSIS table

Flexibility Analysis Page

- Precondition: User is a coach or an administrator
 - Input form for entry of flexibility and posture elements are displayed
 - Entered data is saved to FLEXIBILITY_ANALYSIS table

View Printable Workout Page

- Precondition: User is a client, coach, or administrator
- Precondition: User has at least one active workout
 - The clients current active workouts are displayed on the page without any extraneous page features such as a navigation bar to allow for easy printing

Enter Workout Results Page

- Precondition: User is a client, coach, or administrator
- Precondition: User has at least one active workout
 - A form with fields for all the workout information is displayed
 - Entered results are used to change the workout intensity based on consecutive failures or completions. The increase/decrease percentage and number of consecutive results are configurable through the Site Management page
 - Entered results are saved to the USER_WORKOUT_RESULTS table

Workout History Page

- Precondition: User is an client, coach, or administrator
 - A list of all the user's active and inactive workouts are displayed on the page
 - The user can click on each workout to see a trending graph and total detailed history of the workout results

Change Password Page

- Precondition: User is logged in
 - A form requiring a new password is displayed
 - New password is encrypted and saved to USERS table

Login/Home Page

- Precondition: User is logged in
 - Main navigation bar and site logo is displayed
- Precondition: User is not logged in
 - A login screen with the site logo is displayed
 - User input is parsed and validated against USERS table

Help Pages

- Pre-Condition: None
 - Various help pages are displayed

2.0.3 Database

A database schema will be designed to accommodate all of the additional site functionality requested by Jerry Howard. The database tables will be grouped into three main categories: base system data, user data, and workout data. Base system tables will store site configuration and "building block" data used in the construction of users and workouts. User data will be inputted as new accounts are created, and will store a collection of personal, health, and security information about the user. Workout data includes workout templates, customized user workouts, workout assignments, and workout history.

The database will be in 3rd normal form, which means it will meet the following criteria:

- All redundant columns will be eliminated from each table. Each piece of information should be represented only once within a table
- All tables will have primary keys assigned to uniquely identify each row within the table
- All fields not directly related to a primary key in a table will be removed and placed in a separate table. Each piece of information will appear only once in the entire database after this step
- All tables will have constraints or restrictions enforced through foreign keys if necessary. If a table references data in another table, it will have a foreign key relationship with that table

2.0.4 Common Function Libraries

A library of common function calls available to every page will be developed to avoid code redundancy and increase efficiency. These functions will fall into the following categories.

- Form Input Verifications
- Database Functionality
- HTML Output
- User Verification and Security
- Navigation Bar Generation

Due to the large amount of functions needed, three different library files will be created. These files being Library.php which will include generic functions needed for every page, dbLibrary.php which will include all functions related to the database, and clientNavBar.php which will include all functions that generate the navigation bar.

2.0.5 Third Party Software

This project will use free open source and open license code from two third parties to better the sites functionality, and allow more time to be spent on other areas of development. Code from Matt Kruse at <http://www.mattkruse.com/> will be used to create expandable HTML trees and select box functionality. The code consists of some JavaScript and a set of Cascading Style Sheets (CSS) definitions. See documentation on rosters.php, workoutHistory.php, workout.php, and manageUsers.php for specific implementation details.

Code from an open source project at <http://www.sourceforge.net/projects/phplot/> will also be used to render graphs for the workout history page. The code consists of PHP using the gd library which is an optional graphics library built into PHP. See documentation on workoutHistory.php for specific implementation details.

2.0.6 Project Development Flow

In the project development figure 2.1 below, any webpages that write to the database are considered Administrative. Pages in this category are listed under Administrative Pages in the appendix. Any pages that read from the database and display information are considered Display/Reporting pages. Pages in this category are listed under Display/Reporting Pages in the appendix

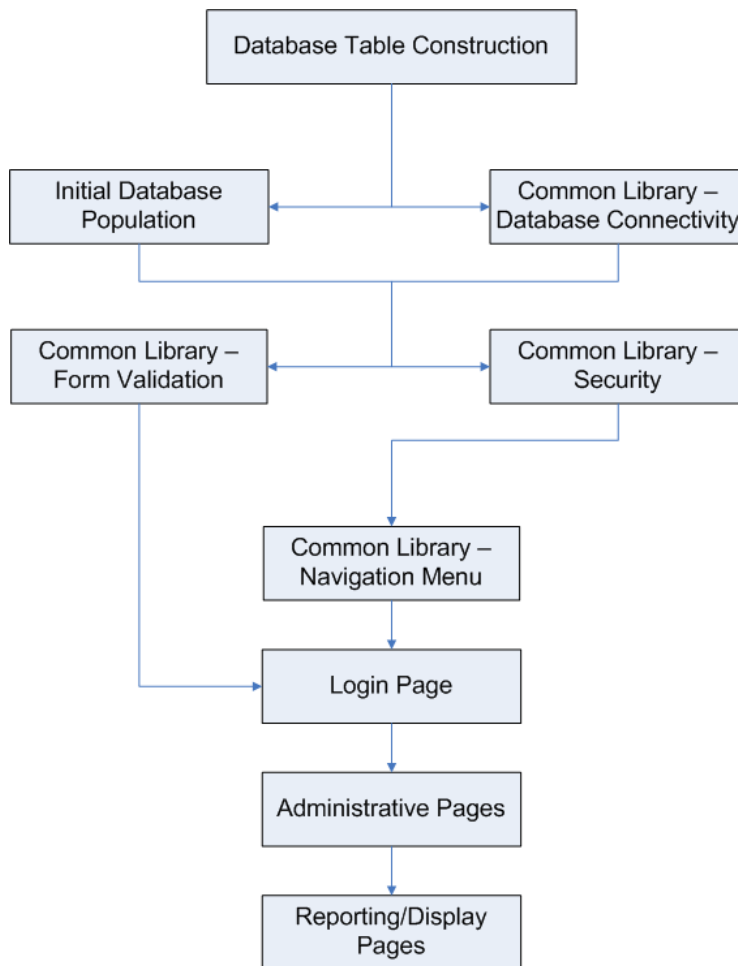


Figure 2.1: Project Design Flow

2.0.7 Estimated Project Time

Table 2.2: Estimated Webpage Development Time

Web Page Name	Estimated Development Time
Login/Home	1 hour
Change Password	1 hour
Reset Password	1 hour
Site Management	1 hour
Help	1 hour
View/Change/Modify User Info	14 hours
View Printable Workout	2 hours
Enter Workout Results	3 hours
Workout History	14 hours
Create/Delete/Modify Workout	4 hours
Assign Workout	1 hour
Create/Modify/Delete Team Roster	2
Body Composition Analysis	1 hour
Flexibility and Postural Analysis	1 hour
Create/Modify/Delete Exercise	1 hour
Create/Modify/Delete Exercise Rule	1 hour
Create/Modify/Delete Exercise Surface	1 hour
Create/Modify/Delete Body Category	1 hour
Create/Modify/Delete Health Attribute	1 hour
Create/Modify/Delete Client Attribute	1 hour
	53 hours

Table 2.3: Estimated Database Development Time

Database Task	Estimated Development Time
Database Table Creation	4 hours
Database Table Initial Population	4 hours
Query Design	8 hours
Constraint Testing	4 hours
Data verification	8 hours
	28 hours

Table 2.4: Estimated Common Function Library Development Time

Function Type	Estimated Development Time
Forum Input Verifications	8 hours
Database Connections	1 hours
HTML Output	2 hours
User Verification and Security	2 hours
Navigation Bar Generation	4 hours
	17 hours

Table 2.5: Estimated Total Project Development Time

Development Task	Estimated Development Time
Webpage Development	53 hours
Database Development	28 hours
Common Function Library Development	17 hours
	98 hours

3.0 Implementation

3.0.1 PHP Pages

Each PHP page was designed to fulfill one or more of the required webpage interfaces listed in table 2.1. In addition, physical pages were constructed to hold common library functionality. The physical pages and their associated design tasks are listed in table 3.6 below:

Table 3.6: Physical Webpages

File Name	Design Task
assignWorkout.php	Assign Workout Page
bodyCategory.php	Create/Modify/Delete Body Category Page
bodyCompAnalysis.php	Body Composition Analysis Page
changePassword.php	Change Password Page
clientAttributes.php	Create/Modify/Delete Client Attribute
clientNavBar.php	Client Navigation Bar Library
cpa.css	Website Style Sheet
datadump.php	Database Tool
dbLibrary.php	Database Function Library
exercise.php	Create/Modify/Delete Exercises
exerciseRules.php	Create/Modify/Delete Exercise Rules
exerciseSurfaces.php	Create/Modify/Delete Exercise Surfaces
flexPostAnalysis.php	Flexibility and Postural Analysis Page
graphHistory.php	Graphing Library
healthAttributes.php	Create/Modify/Delete Health Attributes
help.php	Help Page
index.php	Login/Home Page
Library.php	Common Function Library
logout.php	User Logout Script
manageUsers.php	User Management Page
reactivateUser.php	Reactivate User Page
resetPassword.php	Reset Password Page
roster.php	Create/Modify/Delete Roster Page
site.php	Site Management Page
user.php	Create/Modify/Delete User Page
UserLibrary.php	Library Functions For user.php
viewWorkout.php	View Printable Workout Page
workout.php	Create/Modify/Delete Workout Page
workoutHistory.php	Workout History Page
workoutResults.php	Workout Results Page

3.0.2 Database

The final database schema is shown in the appendix as figure 7.2. Individual tables with a description of their contents are listed in table 3.7 below:

Table 3.7: Database Tables

Table Name	Data Stored
BODY_CATEGORIES	Base Body Categories
BODY_COMPOSITION_ANALYSIS	Body Composition Data
CLIENT_ATTRIBUTES	Base Client Attributes
EXERCISES	Base Exercises
EXERCISE_RULES	Base Exercise Rules
EXERCISE_SURFACES	Base Exercise Surfaces
FLEXIBILITY_ANALYSIS	Flexibility Analysis Data
HEALTH_ATTRIBUTES	Base Health Attributes
ROSTERS	Base Roster Data
ROSTER_USERS	User and Roster Assignments
SITE_CONFIG	Site Configuration Data
TRAINER_USERS	User and Trainer Assignments
USERS	User and Security Data
USER_CLIENT_ATTRIBUTES	User Attribute Data
USER_HEALTH_ATTRIBUTES	User Health Data
USER_WORKOUT	User and Workout Assignments
USER_WORKOUT_EXERCISES	User and Exercise Assignments
USER_WORKOUT_RESULTS	User Workout Result Data
WORKOUT	Base Workouts
WORKOUT_EXERCISES	Base Workout Exercise List

The final database is not completely normalized. Due to design changes during implementation, normalization in tables EXERCISE, WORKOUT, WORKOUT_EXERCISES, USER_WORKOUT_EXERCISES, and USER_WORKOUT was abandoned. Tables EXERCISE and WORKOUT were originally designed as parent tables with different levels of customization, but have instead become template tables. Therefore the table rows titled WORKOUT_ID and EXERCISE_ID, while still under foreign key constraint, are technically redundant data which breaks normalization rules. This does not impact website performance or cause any difficulties unless manual actions are taken on the database through the MySQL console.

4.0 Debug and Testing

The majority of debugging and testing occurred during development as many pages require proper functionality of other webpages before they can be developed. Our client, Jerry Howard, will be responsible for fine-grained debugging of specific functionality and approval of user interfaces.

All changes must be submitted by our client by end of business on Wednesday, June 22.

5.0 Delivery

All listed items will be delivered to Jerry Howard by the end of business on Friday June 24.

- PHP webpages
 - All pages listed in the design section will be supplied as .php script files, organized for installation on the client's server
- Third party source files and documentation
 - Javascript, PHP and CSS components of third party software will be integrated with the site
- Database build scripts
 - Files to automate the creation of the client's database tables and relationships
- Database initial population scripts
 - Files to automate insertion of required base data after database creation
- Site documentation
 - User interface instructions and general usage guidelines

6.0 Project Conclusion

All of the functionality outlined in this design document has been successfully implemented. The final product is a website capable of assisting personal trainers in assigning workouts to their clients and monitoring client performance. However, the workout results page, while implemented, is only capable of displaying a limited number of workout results in a graph. It was intended to have more functionality such as displaying data by month, but its full potential could not be coded within the project time-line.

7.0 Appendix

Administration Pages

- Change password
- Reset password
- Create/Modify/Edit/Delete User
- Enter Workout Results
- Create/Delete/Modify Workout
- Assign Workout
- Create/Delete/Modify Team Roster
- Body Composition Flexibility and Postural Analysis
- Create/Modify/Delete Exercise
- Create/Modify/Delete Body Category
- Create/Modify/Delete Health Attribute
- Create/Modify/Delete Client Attribute

Display/Reporting Pages

- Site Management
- Help
- View Personal Info
- View Health History
- View Printable Workout
- Workout History
- Body Composition Analysis (view)
- Flexibility and Postural Analysis (view)

