

**SOFTWARE DESIGN SPECIFICATION**  
*Direct6 Software Design*

**Department of Computer Science – College of Engineering and Computer Science**  
**California State University, Sacramento**

**Table of Contents**

<b>1. Introduction.....</b>	<b>1</b>
<b>2. Architectural Design.....</b>	<b>4</b>
<b>3. Interface Design.....</b>	<b>6</b>
<b>4. Database Schema.....</b>	<b>13</b>
<b>5. Component Design Specification .....</b>	<b>16</b>
<b>5.1. Sequence Diagram.....</b>	<b>17</b>
<b>5.2. Component Design Specifications.....</b>	<b>26</b>
<b>6. Performance Analysis.....</b>	<b>32</b>
<b>7. Feasibility and Resource Estimates.....</b>	<b>33</b>
<b>8. Software Requirements Traceability Matrix.....</b>	<b>34</b>
<b>9. Approvals.....</b>	<b>35</b>
<b>Appendix A: Data Dictionary.....</b>	<b>36</b>
<b>Appendix B: Attribute Association.....</b>	<b>39</b>

## 1. INTRODUCTION

This is the software design specification document for the ITCC database project sponsored by Inter Tribal Council of California.

This project is being undertaken by Direct 6 Software Solutions development team. The team is comprised of undergraduate students majoring in Computer Science at California State University, Sacramento. The team members are enrolled in a two-semester senior project course required of all undergraduate majors. Successful delivery of the desired software product will fulfill the senior project requirement for the student team members.

### PROJECT SPONSOR

**Name:** Mr. Steven Archer

**Title:** Planning and Development

Organization name: Inter-Tribal Council of California (ITCC)

Contact information: Cell: (916) 320-0233

email: archers@cwnet.com

### DIRECT 6 SOFTWARE SOLUTIONS DEVELOPMENT TEAM

Jared Arbaugh	Cell: (916) 771 – 3501	Email: <a href="mailto:arbaughj@gmail.com">arbaughj@gmail.com</a>
Dave Diel	Cell: (916) 402 – 8640	Email: <a href="mailto:diel@surewest.net">diel@surewest.net</a>
Ben Mackin	Cell: (408) 646 – 3521	Email: <a href="mailto:bmackin@mac.com">bmackin@mac.com</a>
Khae Saetern	Cell: (916) 208 – 1964	Email: <a href="mailto:khaes1421@hotmail.com">khaes1421@hotmail.com</a>
Benjamin Tovar	Cell: (209) 712 – 1417	Email: <a href="mailto:tovarb@gaia.ecs.csus.edu">tovarb@gaia.ecs.csus.edu</a>
Kou Vang	Cell: (916) 233 – 5051	Email: <a href="mailto:vangkou@gmail.com">vangkou@gmail.com</a>

#### 1.1 Purpose

This document is to provide the specifications for the design for the ITCC database project, which is to be developed by the Direct 6 Software Solutions.

#### 1.2 Scope

This document will include technical details regarding the inner workings of the different pieces of the system, as well as describing usability issues and detailing how all requirements from the Software Requirements Specification document will be met. This document represents the baseline design specification for the ITCC database system. Once approved, any desired changes to this document must be processed through the development team's baseline change procedures.

#### 1.3 Definitions, Acronyms and Abbreviations

- 1.3.1 Baseline** – is a work product that has been formally reviewed and accepted by the involved parties. A baseline is changed only through formal configuration management procedures.

- 1.3.2 **Milestone** – is a scheduled event used to measure progress.
- 1.3.3 **MySQL** - is an open source relational database management system (RDBMS) that uses Structured Query Language (SQL) for adding, accessing, and processing data in a database.
- 1.3.4 **PHP** - is a programming language that allows web developers to create dynamic content that interacts with relational databases.
- 1.3.5 **Project Deliverable** – is a work product that is delivered to the project sponsor, advisor, team or project manager. .
- 1.3.6 **Task** – is the smallest unit of work subject to management accountability.
- 1.3.7 **Csc** – is an abbreviation used to represent Computer Science
- 1.3.8 **SRS** – is an acronym for the software requirement specification. The SRS documents the essential requirements (functions/features/uses, performance, design constraints and attributes) of the software and its external interfaces.
- 1.3.9 **Schema** – is a diagrammatic representation of the relational database including what relationships the database has.
- 1.3.10 **Use cases** – is a graphic representation of a user involved in every possible interaction they will have with the software. Use cases are used extensively in software requirements.
- 1.3.11 **Scope** – is a measurement of what the project is and what it isn't.

#### 1.4 References

Document was prepared using the following references below.

Buckley, R. Guide to Preparing the Software Design Document. Version 10.18.04

Buckley, R. Guide to Preparing the Software Requirements Specification Document. Version 7.3.03

Pressman, Roger S. 2001. *Software Engineering: A Practitioner's Approach*, McGraw Hill.

## 1.5 Overview of Contents of Document

- **Section 2 of this document describes the architectural design.** The architectural design of the system describes what functionality is assigned to what subsystems, where the data will be stored, and how much communication is required through the user interface, between subsystems and with the database. This section is intended to provide a road map into the software.
- **Section 3 of this document describes interface design.** This section provides prototypes of the interface, and it specifies the look, feel and behavior of that portion of the system that is visible to the user.
- **Section 4 contains the database schema.** This section provides the translation of the informational model contained in the Software Requirements Specification (SRS) into a relational database.
- **Section 5 contains the component design specifications.** This section provides the detailed description for the design of the software. The section begins with a traceability matrix that map use cases to the web page and the components needed to server the web pages.
- **Section 6 contains the performance analysis.** It provides details on any performance issues or constraints that may have arisen during the design and implementation phase.
- **Section 7 contains feasibility and resource estimates.** This section is a summary of the computer resources required to build, operate and maintain the software.
- **Section 8 contains the requirements traceability matrix.** The matrix relates the design components and the requirement elements by relating the paragraph numbers in this document to paragraph numbers in the Software requirements Specification Document.
- **Section 9 contains approvals.** This section is a sign off sheet that is used to indicate approval of and agreement to the design specification contained in this document. The Signatories for this document include each member of the project team, and the professor Jiang our faculty adviser.
- **Appendix A includes database tables and attributes.**
- **Appendix B includes listing of all attributes with associated characteristics.**

## 2. ARCHITECTURAL DESIGN

The architectural design of the system describes what functionality is assigned to what subsystems, where the data will be stored, and how much communication is required through the user interface, between subsystems and with the database.

### 2.1. Hardware Architecture

The application uses a two tier architecture which consists of a single server machine that will hosts both the database and serves as the web server. The order of data flow is from the client to the Web Server/Database Server to process the requests as shown in figure 2.1.

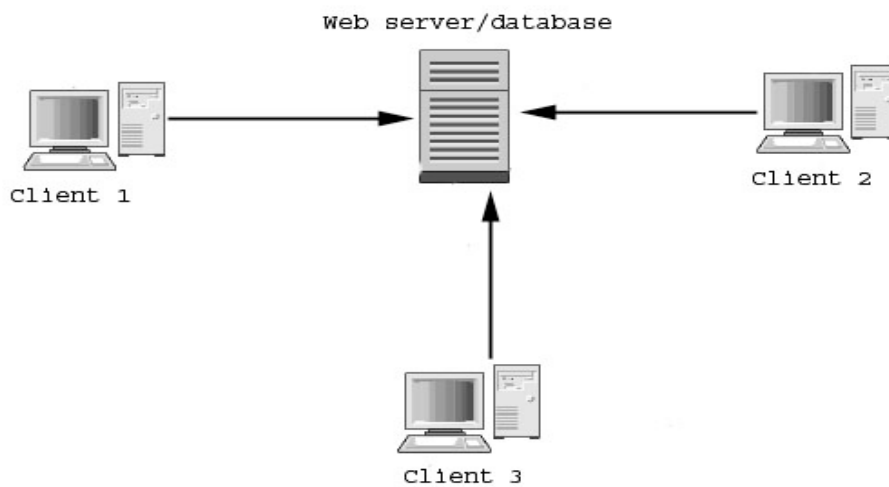


Figure 2.1 – Image showing the hardware, and how it interacts within the system.

### 2.2. Software Design Architecture

This software is a web-based application. The concepts used in the design of the software were that the UI must have an intuitive feel. All the information that an end user will need for the completion of the application process must be easily available.

To accomplish this ease of use the application provides the end user with easily navigation systems and consistency look and feel. With these features the end user can easily navigate from page to page and find information easily. Any request that the user makes from the interface is processed on the Web Server. The Web Server then queries the Database Server which resides on the same physical server for the information to be displayed back to the user. This is a typical Client-Server logic model for the exchange of information and business processing.

### **2.2.1 Presentation Layer**

This is the user interaction layer. Its main goal is to receive event requests and to process incoming data. Its other main goal is the displaying of this data formatted properly into reports. The presentation layer is the graphical user interface of our program.

### **2.2.2 Business Layer**

The business layer takes the incoming data, and applies the necessary SQL commands to manipulate it. This layer also includes the rules and logic of the requirements that the sponsor defined. This software layer is deployed on the client machine.

### **2.2.3 Data Management Layer**

This layer is the database as stored on the ITCC server. Its job is to manage requests, read and write data to the database. This software layer is defined by MySQL.

### 3. Interface Design

This section specifies the look, feel and behavior of that portion of the system that is visible to the user. The behavior of the ITCC Database Software project is shown in figure 3.1.

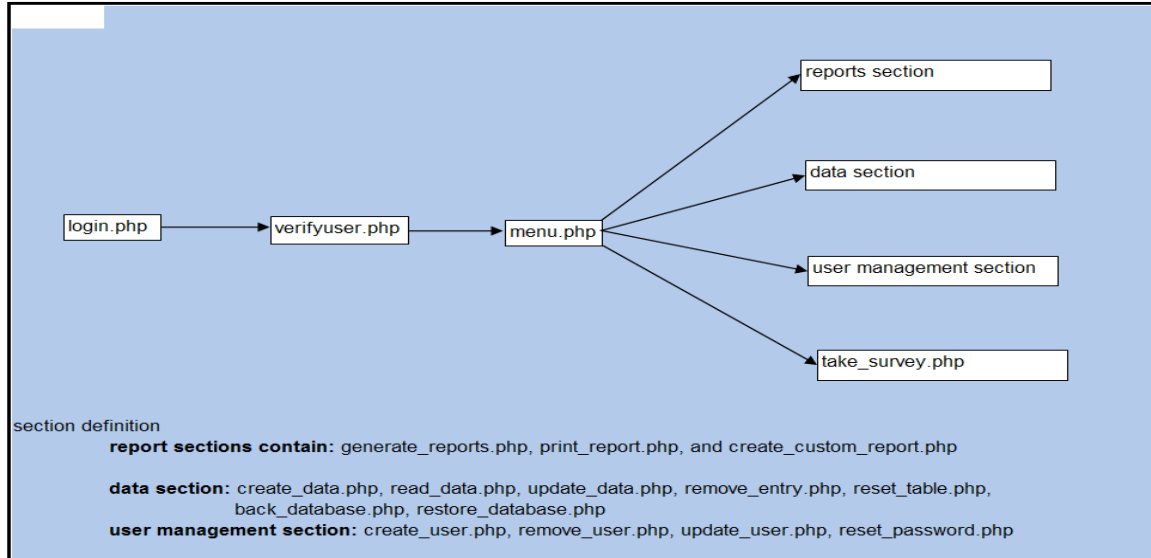


Figure 3.1 – interface design

The following table identifies the PHP files associated with each use case.

Use Case	Webpage
Take Survey	take_survey.php
Login	Login.php
Logout	header.php
Generate Reports	generate_reports.php
Print Reports	Print_reports.php
Create Custom Reports	create_custom_reports.php
Create Data	create_data.php
Read Data	Read_data.php
Update Data	update_data.php
Remove Entry	remove_entry.php
Reset Table	Reset_table.php
Remove User	remove_user.php
Reset Password	Reset_password.php
Update User	update_user.php
Create User	create_user.php
Backup	backup_database.php
Restore Data	restore_data.php

Table 3.1 Mapping between use cases and webpages.

### 3.1 “take\_survey.php”

<input type="text" value="Survey"/>	<b>Survey</b>
	Question1 <input type="text"/>
	Answers <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
	<input type="button" value="Submit"/>

Figure 3.2 – an interface for user to take survey

### 3.2 “login.php”

Login:
Name <input type="text"/>
Pasword <input type="text"/>
<input type="button" value="Submit"/>

Figure 3.3 – an interface for the user to log into the system

### 3.3 Report (create, read, print, generate, and view report)

<table border="1"><tr><td>Survey</td></tr><tr><td>Reports</td></tr></table>	Survey	Reports	<b>Create report</b>	<b>Reports</b>	
	Survey				
Reports					
	<p>Search <input type="text"/></p> <p><input type="text"/></p> <p><input type="button" value="Submit"/></p> <p>Select option</p> <table border="1"><tr><td><input type="checkbox"/></td></tr><tr><td><input type="checkbox"/></td></tr><tr><td><input type="checkbox"/></td></tr></table> <p><input type="button" value="Submit"/></p>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<p>Title</p> <p>.</p> <p>.</p> <p>.</p> <p><input type="button" value="print"/></p>
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Figure 3.4 – a rough interface schema for the user to modify and view data.

### 3.4 Data (create, update, read, and delete)

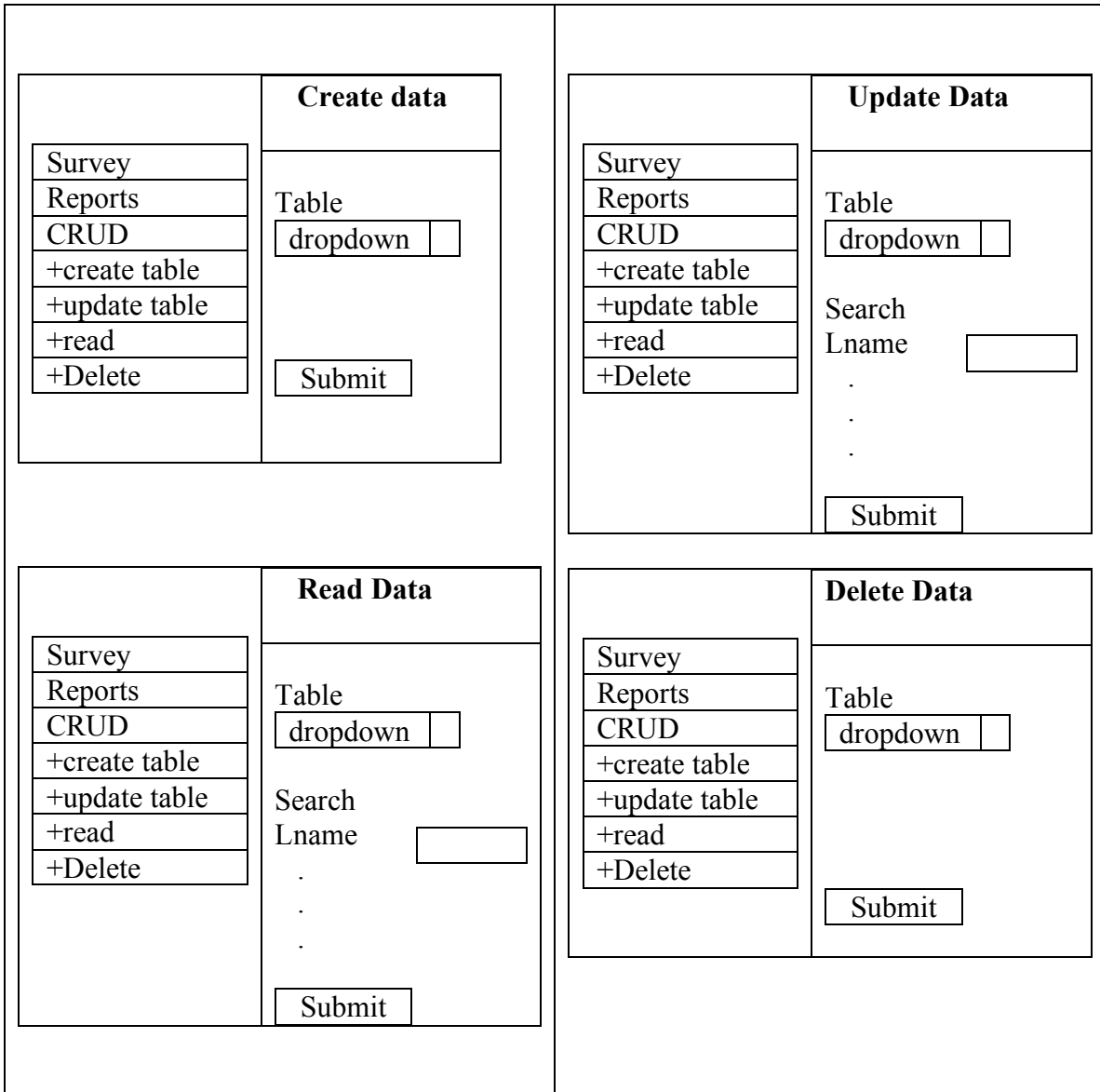


Figure 3.5 – a rough interface schema for the user to modify and view data.

### 3.5 User Management - an interface schema for the administrator to manage users.

#### 3.5.1 Create user

Create User	
Menu Options	User Name: <input type="text"/>
	Password: <input type="text"/>
	User Level: <input type="text"/>
	<input type="button" value="Submit"/>

Figure 3.6 – an interface schema for the create screen.

#### 3.5.2 Delete User

Delete User	
Menu Options	<input type="text" value="UserName"/> <input type="text" value="Password"/> <input type="text" value="Level"/>
	User Name: <input type="text" value="text field"/>
	<input type="button" value="Submit"/>

Figure 3.7 – an interface schema for the delete user screen.

### 3.5.3 Reset Password

<table border="1"><tr><td>Menu Options</td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr></table>	Menu Options								<table border="1"><tr><td colspan="3"><b>Reset Password</b></td></tr><tr><td>UserName</td><td>Password</td><td>Level</td></tr><tr><td>User Name:</td><td colspan="2">"text field"</td></tr><tr><td>New Password:</td><td colspan="2">"text field"</td></tr><tr><td colspan="3">Submit</td></tr></table>	<b>Reset Password</b>			UserName	Password	Level	User Name:	"text field"		New Password:	"text field"		Submit		
	Menu Options																							
<b>Reset Password</b>																								
UserName	Password	Level																						
User Name:	"text field"																							
New Password:	"text field"																							
Submit																								

Figure 3.8 – an interface schema for the reset password screen.

### 3.5.4 Update User

<table border="1"><tr><td>Menu Options</td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr></table>	Menu Options								<table border="1"><tr><td colspan="3"><b>Update User</b></td></tr><tr><td>UserName</td><td>Password</td><td>Level</td></tr><tr><td>User Name:</td><td colspan="2">"text field"</td></tr><tr><td>New Level:</td><td colspan="2">"Drop down box"</td></tr><tr><td colspan="3">Submit</td></tr></table>	<b>Update User</b>			UserName	Password	Level	User Name:	"text field"		New Level:	"Drop down box"		Submit		
	Menu Options																							
<b>Update User</b>																								
UserName	Password	Level																						
User Name:	"text field"																							
New Level:	"Drop down box"																							
Submit																								

Figure 3.9 – an interface schema for the update user screen.

### 3.6 Database Backup and Restore

		<b>Create Back Up</b>			<b>Recovery</b>		
Home	Back Up	Database	Dropdown	Home	File	dropdown	
Recover				Recover			
		Submit			Submit		

Figure 3.10 – a rough interface schema for the backup and restore screens.

## 4. DATABASE SCHEMA

The following section contains the specifications for the database that is developed for ITCC.

### 4.1. ERD diagram

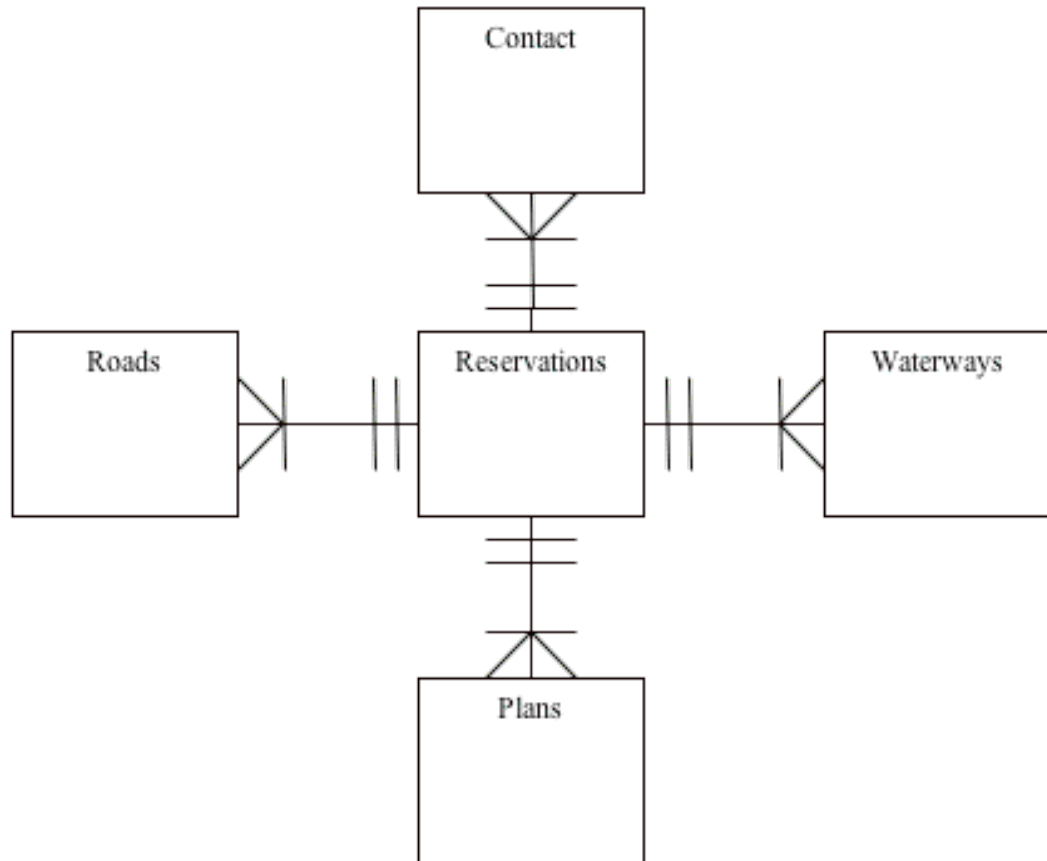
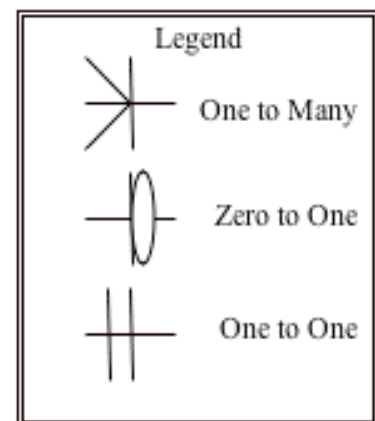


Figure 2.1 Entity Relationship Diagram



## 4.2 Creating the database

This section contains the CREATE statements for the DBMS. The following statements would implement the relationships depicted in figure above:

#uLogin

```
create table uLogin (uname varchar(20), psswd varchar(20), level int);
```

#Contacts contactinfo

```
create table ccontactinfo (rname varchar(50) not null, street varchar(20), county  
varchar (25), zipcode varchar (11), foreign key(rname)references contact(rname));
```

#rancheria/reservation contactinfo

```
create table rcontactinfo (rname varchar(50) not null, street varchar(20), county  
varchar (25),zipcode varchar (11), foreign key(rname)references contact(rname));
```

#Reservation

```
create table reservation (ranchname varchar (50) not null, affiliation varchar(50),  
directions varchar(100), population int, acres int, primary key(ranchname));
```

#roads

```
create table roads(ranchname varchar(50), roadname varchar(20), milesto  
varchar(3), roadtype varchar(11), r_condition varchar (30), primary  
key(ranchname,roadname), foreign key(ranchname)references  
reservation(ranchname));
```

#plans

```
create table plans(ranchname varchar(50), plantype varchar(20), note varchar(100),  
primary key(ranchname,plantype), foreign key(ranchname) references  
reservation(ranchname));
```

#waterways

```
create table waterways(ranchname varchar(50), watertype varchar(20), milesto  
varchar (3), watertype varchar(11), note varchar (100), primary  
key(ranchname,watertype), foreign key(ranchname) references  
reservation(ranchname));
```

#contact person information

```
create table contact(fname varchar (15) not null, lname varchar(15) not null, rname
varchar(50) not null, title varchar(20), id int(11), phone varchar(12), email
varchar(50), primary key(fname,lname), foreign key(rname)references
reservation(ranchname));
```

#survey tables

```
create table surveyCategory(catID int (11) not null auto_increment, catText
varchar(25), primary key(catID));
```

```
create table surveyQuestions(qID int (11) not null auto_increment, catID int (11),
qText varchar(255), type varchar(20), primary key(qID), foreign key(catID)
references surveyCategory(catID));
```

```
create table surveyAnswer(qID int (11) not null, aID int (11) not null
auto_increment, aText varchar (255), primary key aID), foreign key(qID)
references surveyQuestions(qID));
```

```
create table surveyResponse(sID int (11), qID int (11), aID int (11), primary
key(sID), foreign key(qID) references surveyQuestions(qID), foreign key(aID)
references surveyAnswer(aID));
```

### **4.3 Triggers and/or stored procedures.**

Not Applicable

## 5. Component Design Specification – Each person does this for their own module

This section provides the detailed description for the design of the software. The section begins with a traceability matrix that map use cases to the web page and the components needed to serve the web pages.

Use Case	Webpage
Take Survey	take_survey.php
Login	login.php
Logout	header.php
Generate Reports	generate_reports.php
Print Reports	print_reports.php
Create Custom Reports	create_custom_reports.php
Create Data	create_data.php
Read Data	read_data.php
Update Data	update_data.php
Remove Entry	remove_entry.php
Reset Table	reset_table.php
Remove User	remove_user.php
Reset Password	reset_password.php
Update User	update_user.php
Create User	create_user.php
Backup	backup_database.php
Restore Data	restore_data.php

Table 5.1 This table maps the use cases to the appropriate webpages.

## 5.1. Sequence Diagrams

### 5.1.1 Take Survey

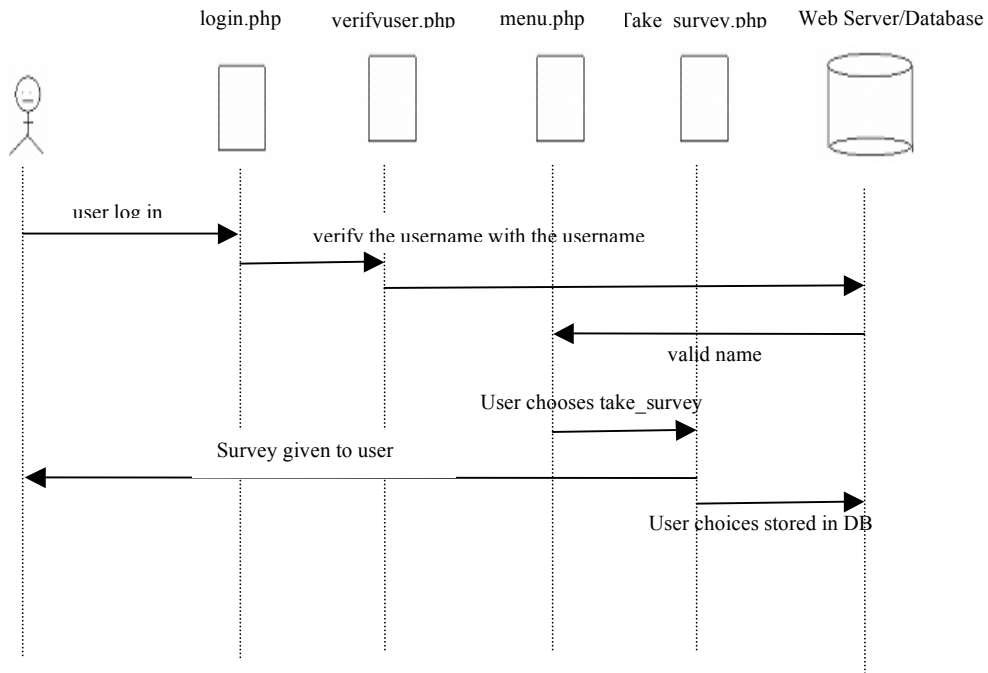


Figure 5.1 - a sequence diagram that models the Use Case Take Survey.

### 5.1.2 Login

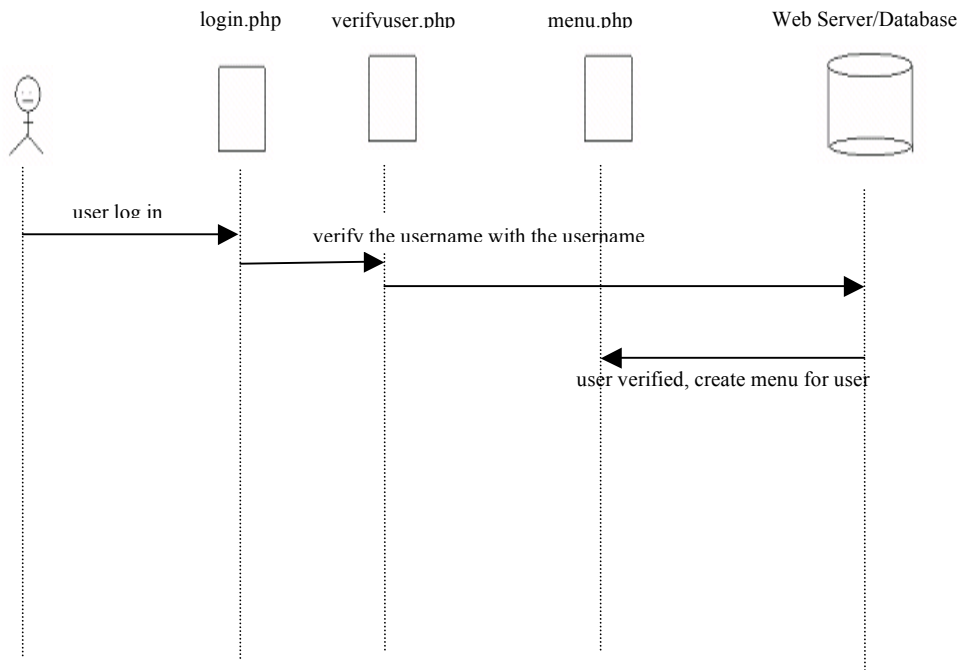


Figure 5.2 - a sequence diagram that models the Use Case Login.

### 5.1.3 Logout

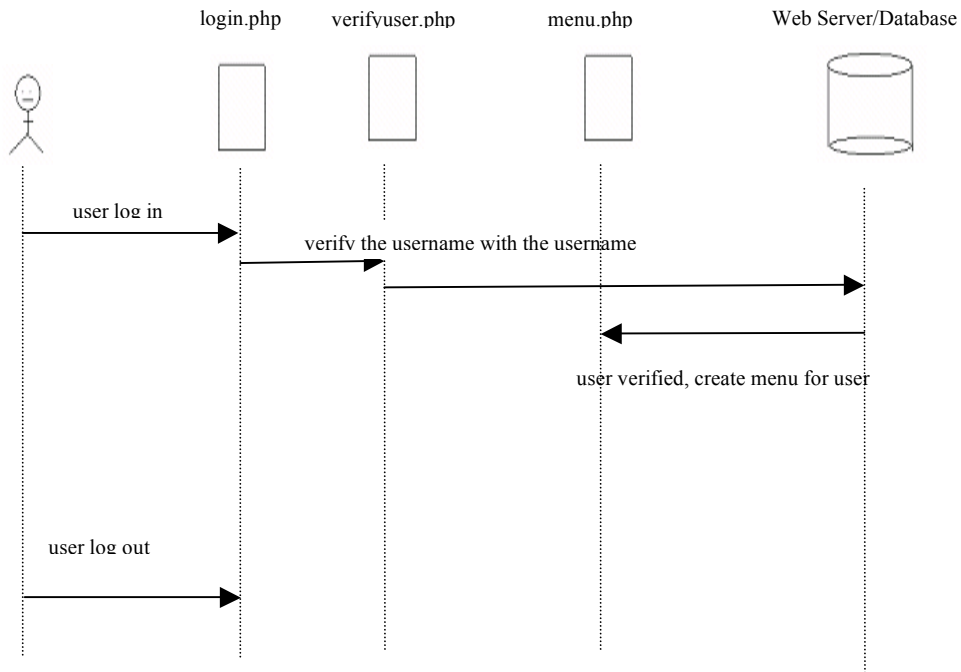


Figure 5.3 - a sequence diagram that models the Use Case Logout.

### 5.1.4 Generate Reports

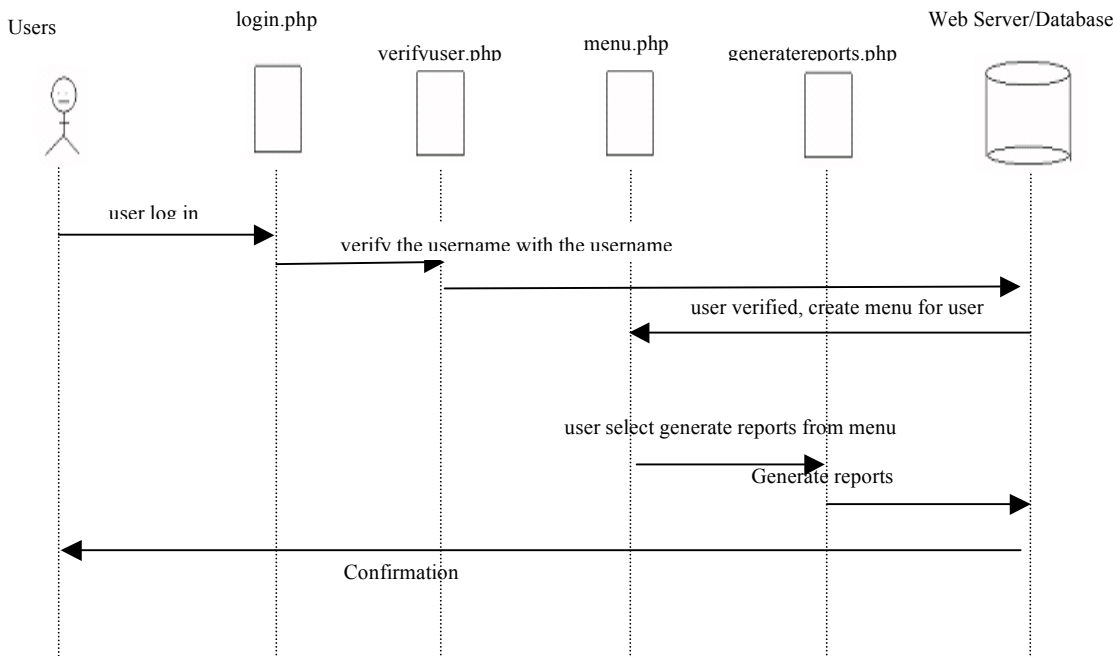


Figure 5.4 - a sequence diagram that models the Use Case Generate Reports.

### 5.1.5 Print Reports

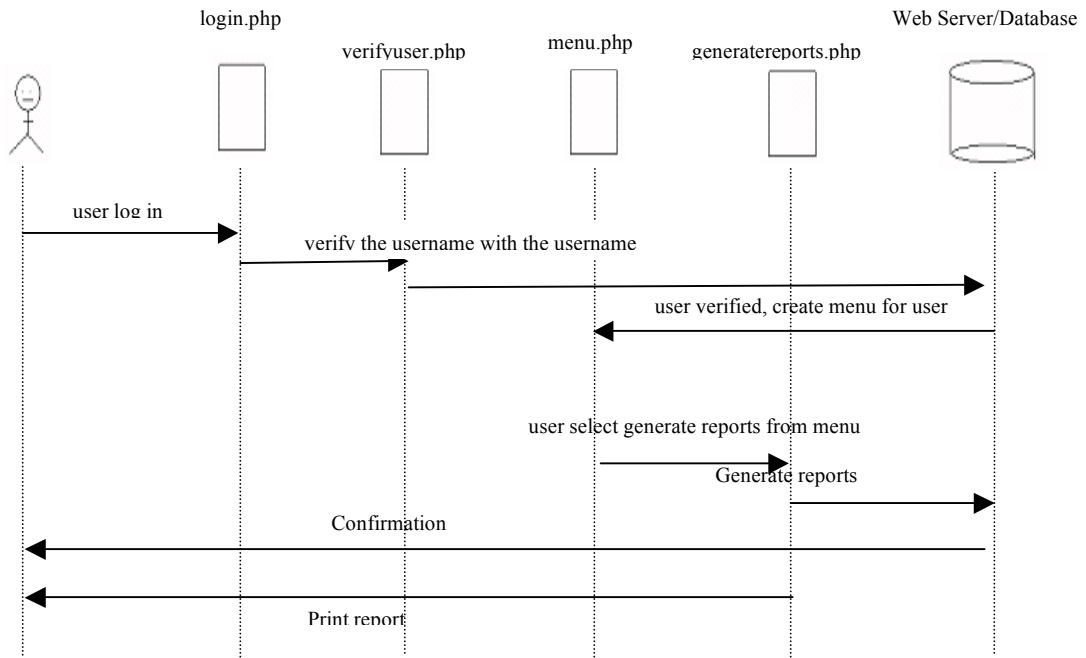


Figure 5.5 - a sequence diagram that models the Use Case Print Reports.

### 5.1.6 Creating Custom Reports

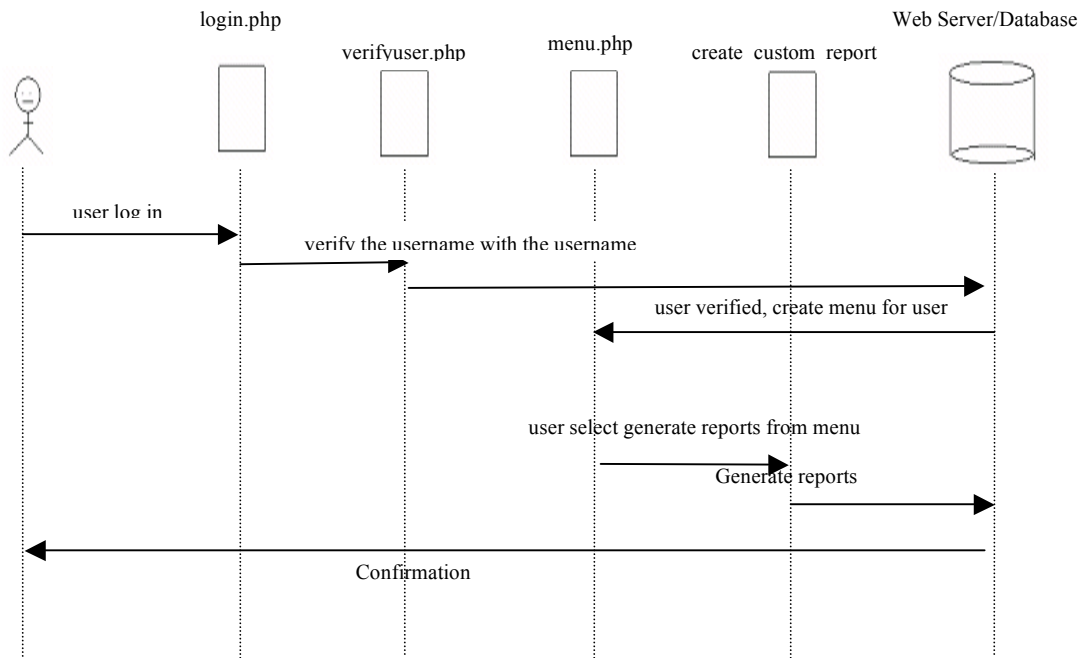


Figure 5.6 - a sequence diagram that models the Use Case Custom Reports.

### 5.1.7 Create Data

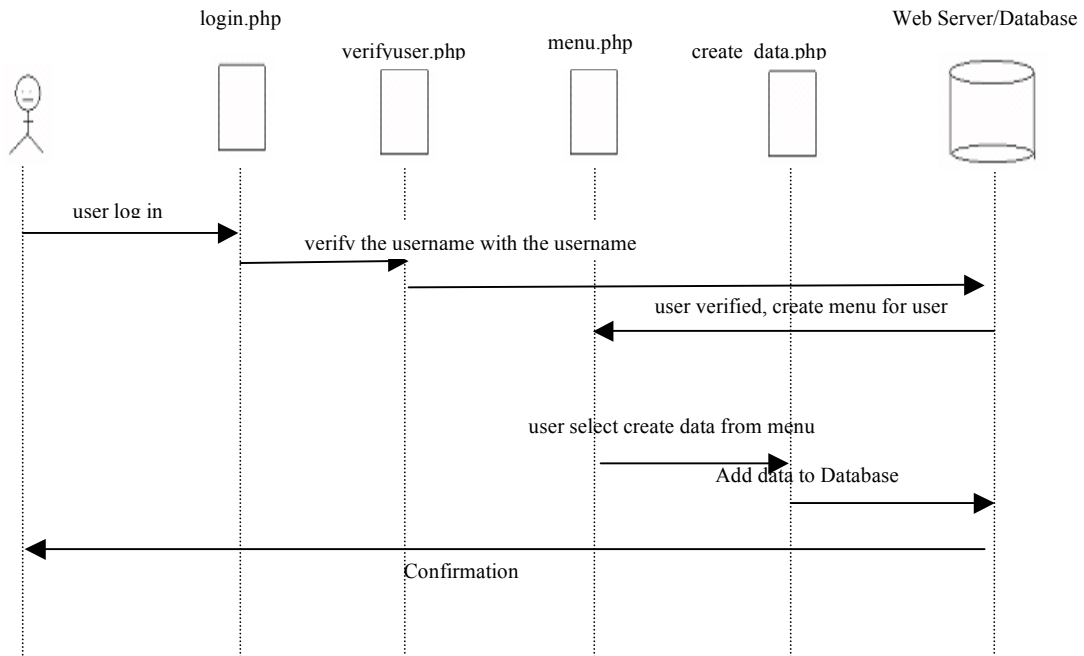


Figure 5.7 - a sequence diagram that models the Use Case Create Data.

### 5.1.8 Read Data

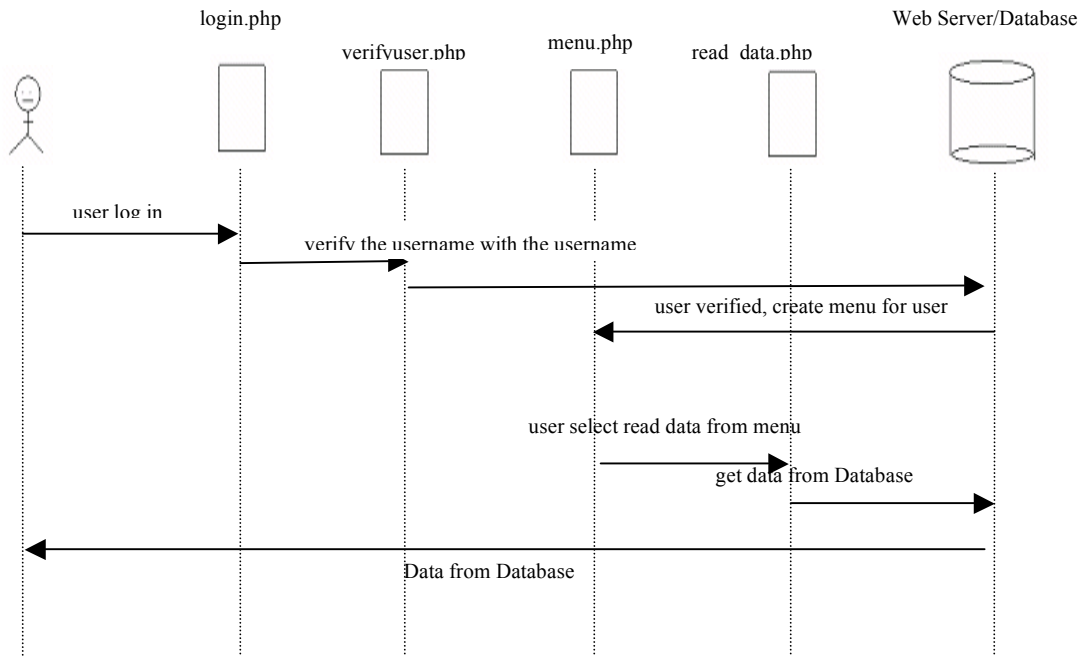


Figure 5.8 - a sequence diagram that models the Use Case Read Data.

### 5.1.9 Update data

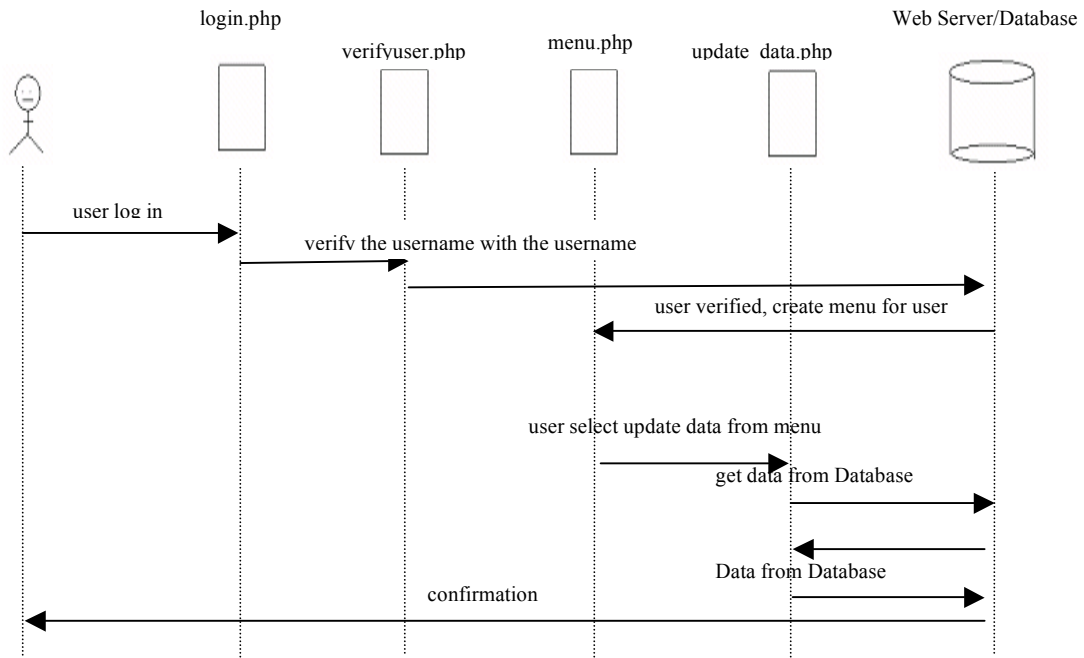


Figure 5.9 - a sequence diagram that models the Use Case Update Data.

### 5.1.10 Remove Entry

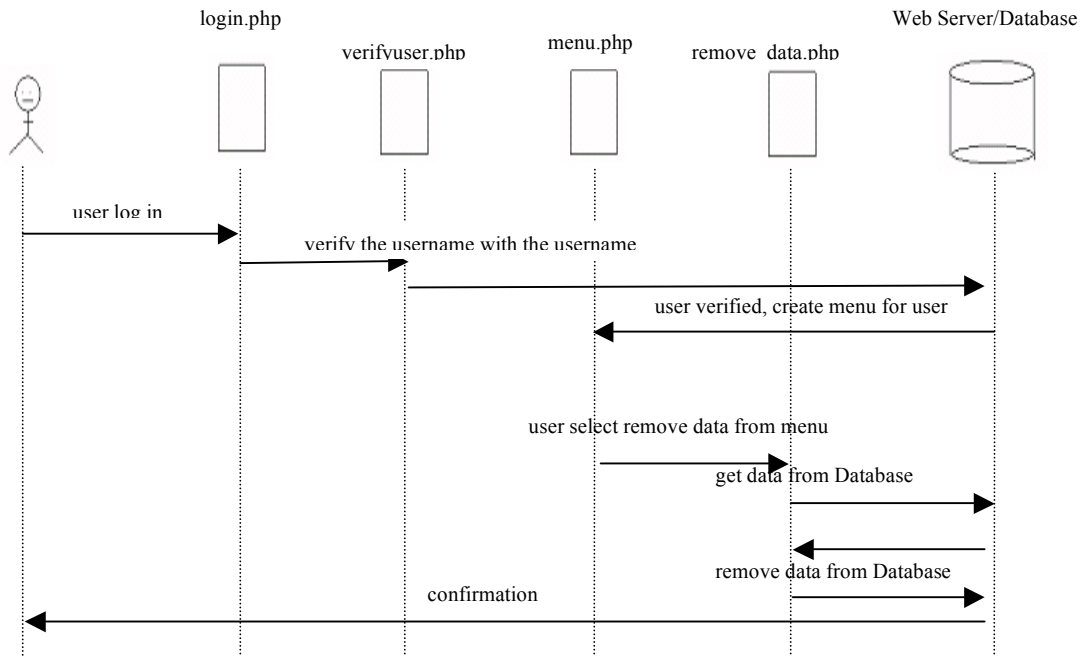


Figure 5.10 - a sequence diagram that models the Use Case Remove Entry.

### 5.1.11 Reset Table

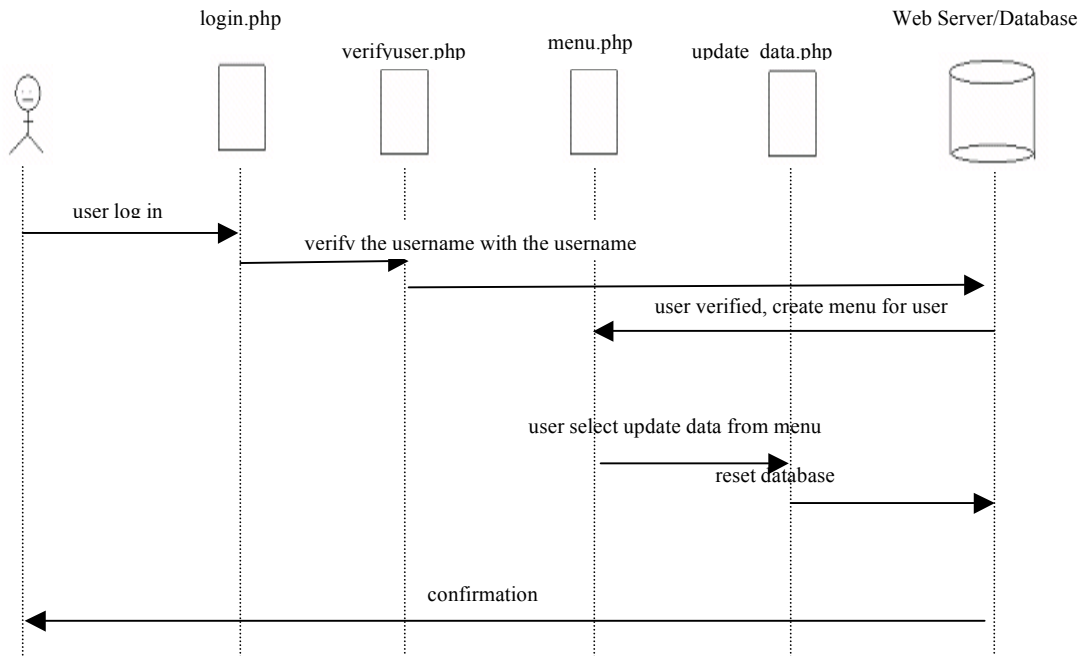


Figure 5.11 - a sequence diagram that models the Use Case Reset Table.

### 5.1.12 Remove user

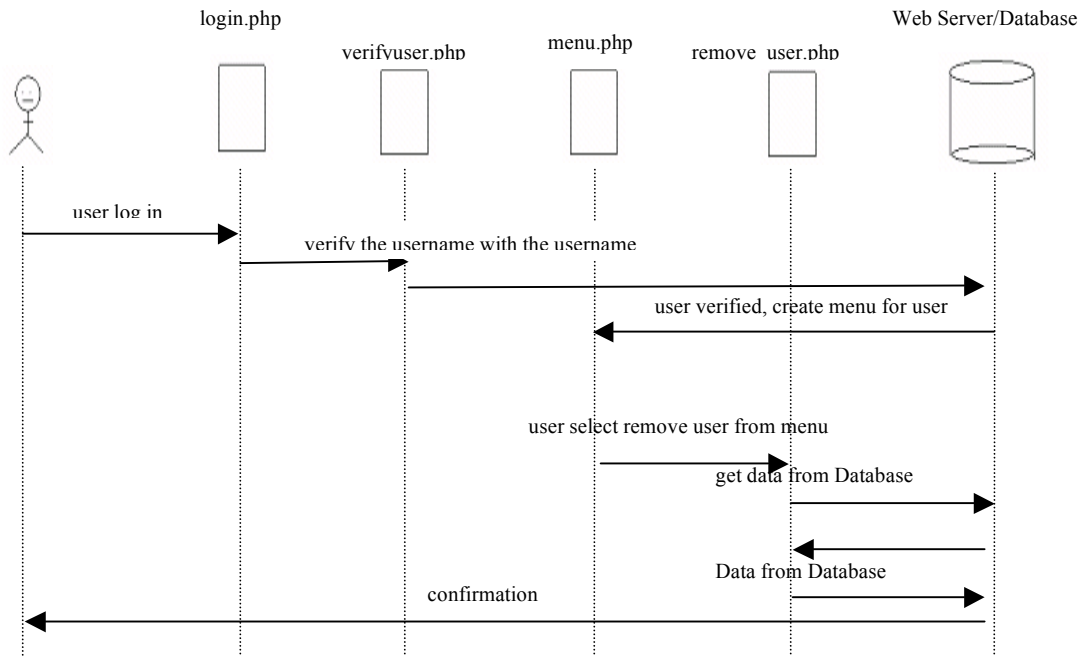


Figure 5.12 - a sequence diagram that models the Use Case Remove User.

### 5.1.13 Reset Password

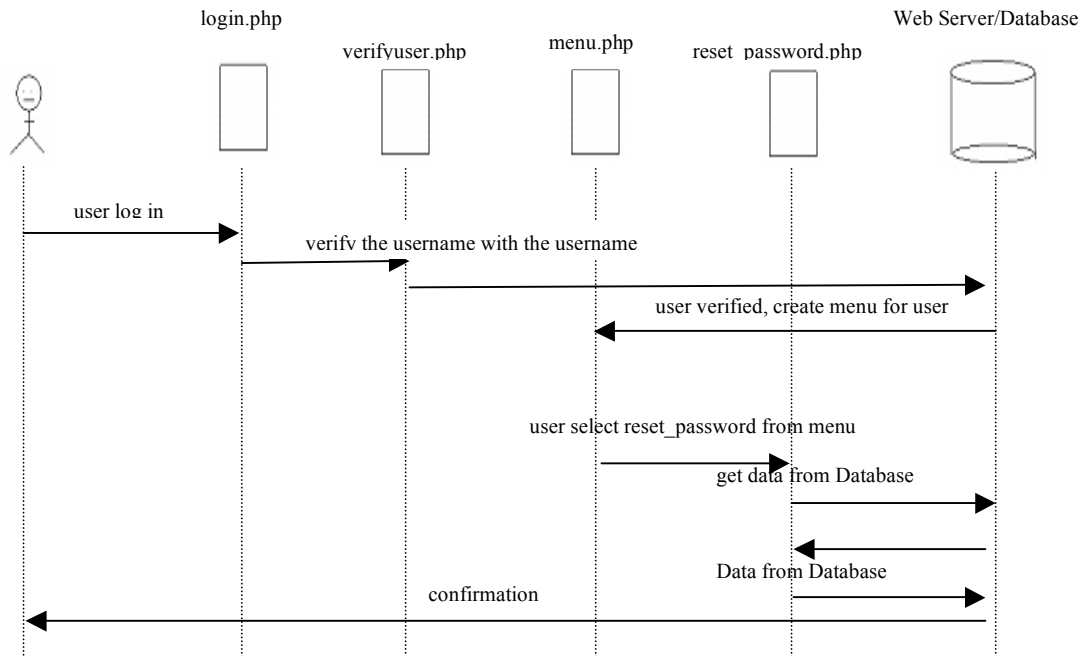


Figure 5.13 - a sequence diagram that models the Use Case Reset Password.

### 5.1.14 Update User

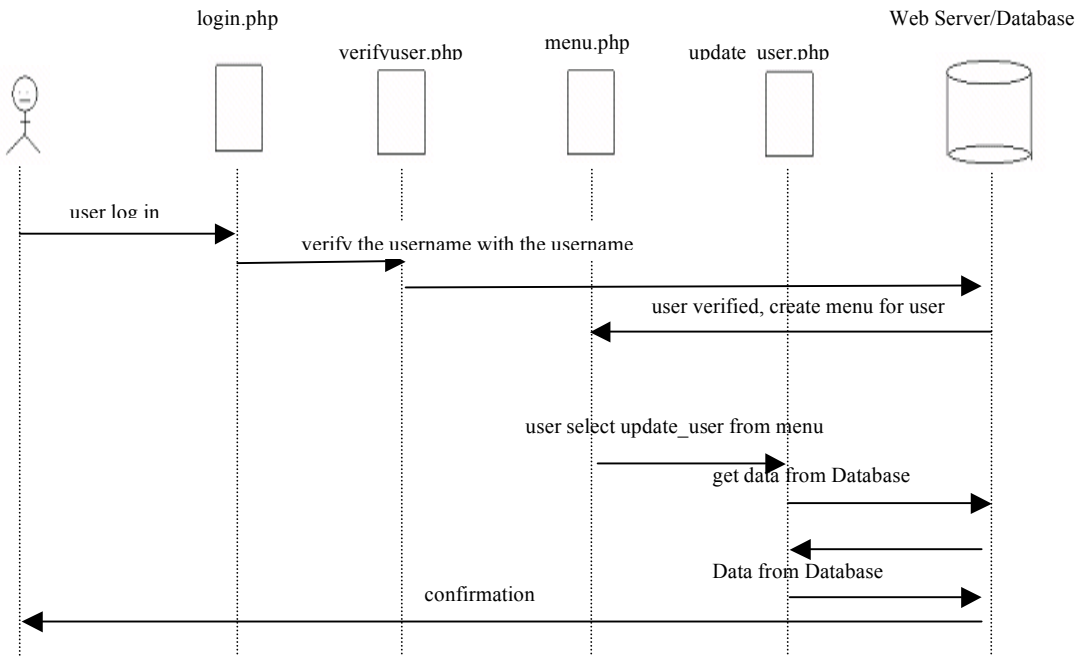


Figure 5.14 - a sequence diagram that models the Use Case Update User.

### 5.1.15 Create User

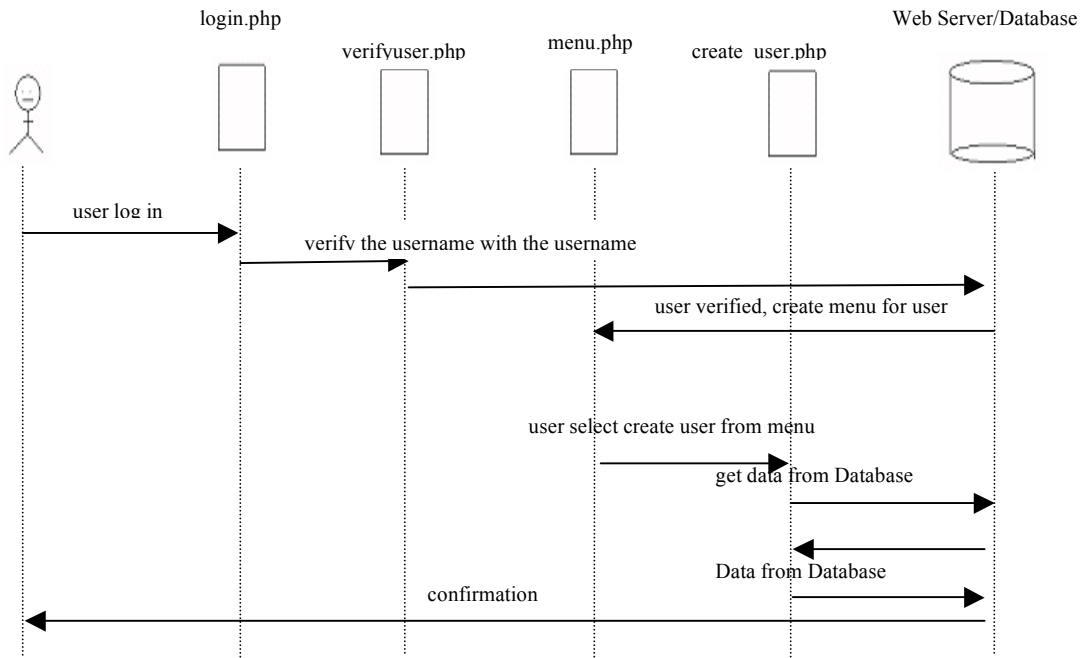


Figure 5.15 - a sequence diagram that models the Use Case Create User.

### 5.1.16 Backup Database

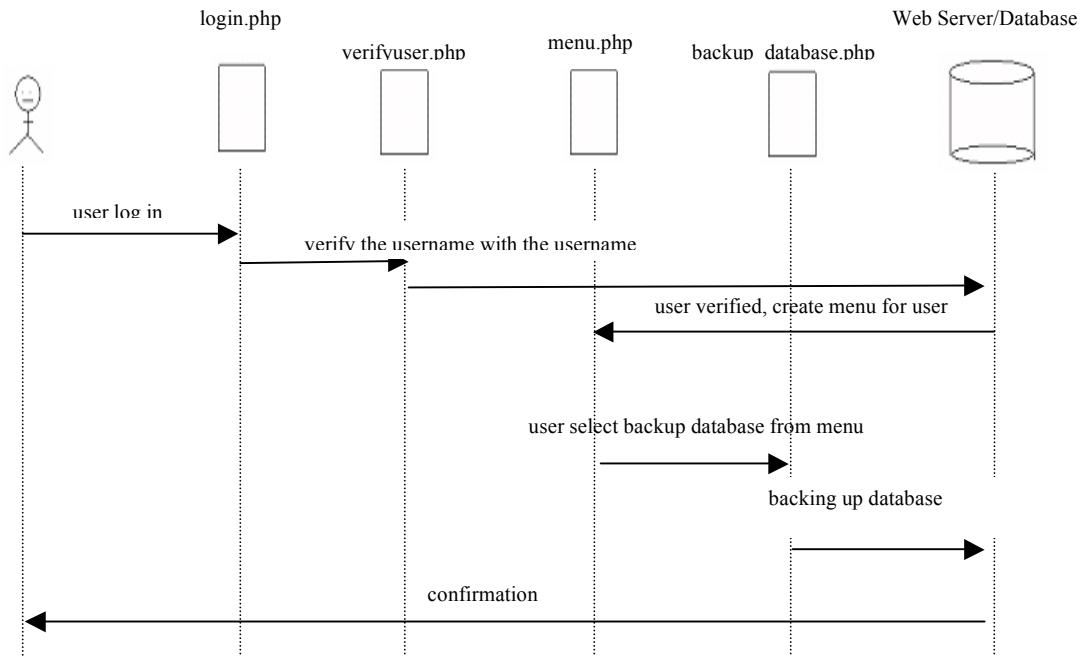


Figure 5.16 - a sequence diagram that models the Use Case Backup Database.

### 5.1.17 Restore Database

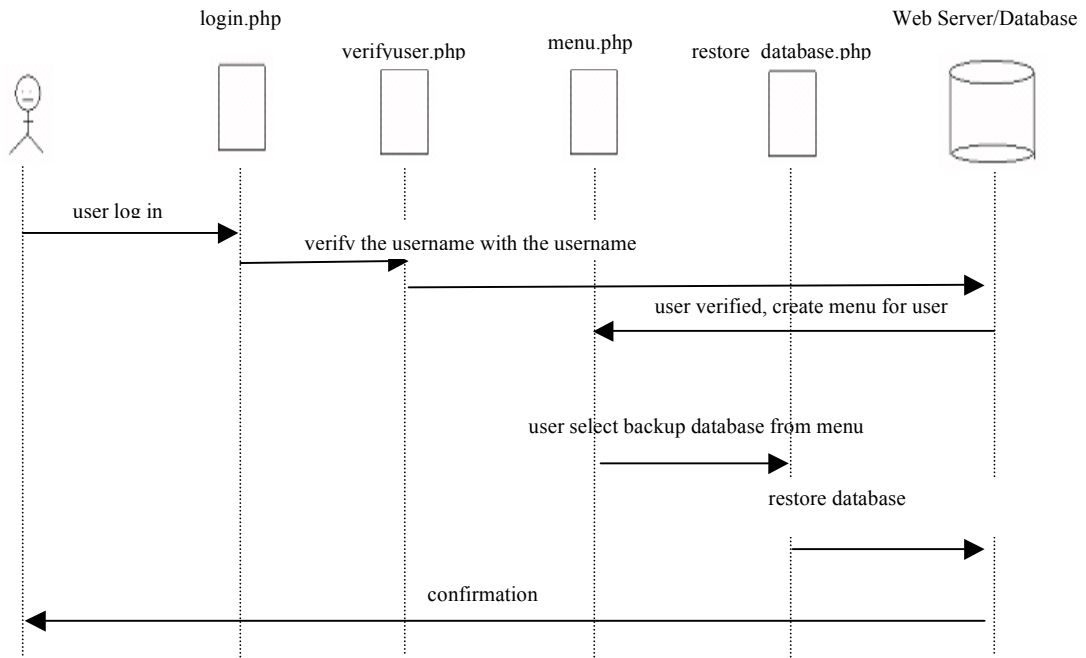


Figure 5.17 - a sequence diagram that models the Use Case Restore Database.

## 5.2 Component Design Specification

### 5.2.1 Take Survey

Users of the system can input surveys online.

**Precondition(s):** Authenticated user is logged into the system.

**Interface Specifications:** Contains Radio Buttons, and a Submit Button.

**Processing Specifications:** Once user selects the survey the DB generates the questions and fields for the user to select.

User then hits submit button and all data is stored to DB.

**Database Requirements:** The surveys must be stored in the DB for this page to display.

**Postcondition(s):** New survey information is stored in the database.

### 5.2.2 Login

User logging into the system.

**Precondition(s):** User must be a valid user in the system

**Interface Specifications:** Contains 2 text boxes Name and password. Two buttons, submit and clear.

**Processing Specifications:** User enters information in required fields then hits the submit button. The Required fields are then tested in the DB for authentication if verified then sent to the users main page according to there level of authority.

If (Username and password are correct) Then  
(Send user level back) Else (Send error)

If (received user level) Then (continue to main.php)  
Else If (received error) Then (Try again)

**Database Requirements:** User must be in the system for this use case to work.

**Postcondition(s):** Access to main.php.

### 5.2.3 Logout

User logging out of the system

**Precondition(s):** User logged into the system.

**Interface Specifications:** consist of a link and found on the header page

**Processing Specifications:** At any time the user can chose to Logout

If (Logout is activated) Then (A confirmation panel is activated)

If (Yes is chosen) Then (Connections are closed and program ends)

Else (No) Then (Program continues)

**Database Requirements:** None

**Postcondition(s):** No connection left opens.

#### 5.2.4 Generate Reports

Users of the system can generate reports based on data stored in the database.

**Precondition(s):** Authenticated user is logged into the system.

**Interface Specifications:** Contains 2 radio buttons, a submit button, text box search, and multiple checkboxes.

**Processing Specifications:** User selects Reports from Main Menu.

User given options based on rights and report types that are generated from the DB. When User clicks on Generate Report button Then program takes user input and generates the specified report.

**Database Requirements:** DB must have information in the tables for this use case.

**Postcondition(s):** Report is generated and displayed on screen.

#### 5.2.5 Print Reports

Users of the system can print reports once they have been generated.

**Precondition(s):** Authenticated user is logged into the system, and a report has been generated.

**Interface Specifications:** Contains submit button, and an identifier Text Box.

**Processing Specifications:** User hits the print button. User is given setup options. User hits enter and the generated report is printed.

**Database Requirements:** None.

**Postcondition(s):** Current report is printed to the selected printer.

#### 5.2.6 Creating Custom Reports

Users of the system can create custom reports based on limited query strings.

**Precondition(s):** Authenticated user is logged into the system. This Use Case creates custom reports based on Bounty or ReservationNames.

**Interface Specifications:** Contains two radio buttons, and checkboxes for all tables in database.

**Processing Specifications:** User given options based on rights

User enters sql commands hits enter. The command is then sent to data base for processing.

**Database Requirements:** database commands processed, if no errors found then the results are returned.

**Postcondition(s):** Report is generated and displayed on screen.

### 5.2.7 Create Data

This use case enables the Administrator or Super User to create data and add it to the tables.

**Preconditions(s):** Authenticated user is an administrator or a Super User

**Interface Specifications:** Drop down box with tables to choose from, text fields that are generated by the choice from the drop down box, and a submit button.

**Processing Specifications:** User is given options to choose from.

User then hits enter and the page creates fields for the user to enter data.

**Database Requirements:** After the user finishes entering data the information is added to the DB.

**Postcondition(s):** A new table is created and is stored in the database

### 5.2.8 Read Data

This use case enables the Administrator or Super to read the table

**Preconditions(s):** Authenticated user is an administrator or a super user

**Interface Specifications:** drop down box filled with table names. A submit button.

**Processing Specifications:** User given options based on rights

User then inputs the option they want then hits enter

Program generates queries and sends them to the DB

**Database Requirements:** DB then runs queries and returns a read only report

**Postcondition(s):** Read from a table in the database

### 5.2.9 Update Data

This use case enables user to update the table.

**Preconditions(s):** Authenticated user is an administrator or a super user

**Interface Specifications:** Radio buttons, input text boxes, and submit button.

**Processing Specifications:** User given options based on rights

User enters the options they desire then hits enter.

Report is generated from returned information.

User is then given choice to update data in the fields

**Database Requirements:** This use case accesses the DB twice once to receive data then again to change data.

**Postcondition(s):** Updated table with new data

### 5.2.10 Remove Entry

This use case enable user to remove or delete an entry from a table

**Preconditions:** Must be logged on with at least Super user rights.

**Interface Specifications:** Radio Buttons for choice, Check Boxes for selection, submit button for conformation.

**Processing Specifications:** User given options based on rights

User is given a choice. Then hits enter. Information is returned

User can chose to remove an entire row. When done the table is updated in the DB.

**Database Requirements:** The DB is accessed twice one to retrieve data then to update the DB when done.

**Postconditions:** An entry in the table is removed from the database

### 5.2.11 Reset Table

This use case enable user to reset the table

**Preconditions(s):** Must be logged on with at least Super user rights.

**Interface Specifications:** Dropdown menu to select table, and submit button for conformation.

**Processing Specifications:** User given options based on rights

User is given a choice. Then hits enter. The User then can reset the table which will empty the DB table when done.

If User dose not chose to Reset the table nothing happens to the DB.

**Database Requirements:** DB is accessed twice once to retrieve data then again to delete the table when done

**Postcondition(s):** Deleted all rows that were setup for reset.

### 5.2.12 Remove User

Administrator can delete users from the system.

**Precondition(s):** Authenticated user is an administrator and is logged into the system.

**Interface Specifications:** Dropdown box with users, and submit button for conformation.

**Processing Specifications:** User given options based on rights

This user is given a list of all users. This user can click on any user in the list and remove them from the DB.

**Database Requirements:** DB accessed twice once for retrieve and again in case of change

**Postcondition(s):** User is removed from the system.

### 5.2.13 Reset Password

Reset password for users.

**Precondition(s):** Administrator must verify user's Identification before reset password.

**Interface Specifications:** Dropdown box for choice, text box for new password, and submit button for conformation.

**Processing Specifications:** User given options based on rights  
This user is given a list of all users. This user can click on any user and change the password.

**Database Requirements:** DB is accessed twice once for retrieve and again for changes.

**Postcondition(s):** Password is reset. New password is stored in the database.

### 5.2.14 Update User

Administrator can update user information.

**Precondition(s):** Authenticated user is an administrator and is logged into the system.

**Interface Specifications:** Dropdown box for choice, dropdown box for levels, and submit button for conformation.

**Processing Specifications:** User given options based on rights  
This user is given a list of all users. This user can change users permissions to the DB.

**Database Requirements:** DB accessed twice once for retrieve and again for update.

**Post condition(s):** New user information is stored in the database.

### 5.2.15 Create User

Administrator can add more users to the system.

**Precondition(s):** Authenticated user is an administrator and is logged into the system.

**Interface Specifications:** Textbox for name, Textbox for password, Dropdown for level, and submit button for conformation.

**Processing Specifications:** User given options based on rights  
This user is given a choice to enter a new user. User hits enter then DB is updated.

**Database Requirements:** DB is accessed once.

**Postcondition(s):** New user has been created and is stored in the database.

### 5.2.16 Back up

This use case enable administrator to back up database.

**Preconditions(s):** Must be logged on with administration rights.

**Interface Specifications:** Dropdown for choice, and submit button for conformation.

**Processing Specifications:** This user selects Back up. User is given options And when done the DB is saved to a location for storage.

**Database Requirements:**

**Postcondition(s):** Create data restore in the database.

### 5.2.17 Restore Data

This use case enable administrator to restore data.

**Preconditions(s):** Must be logged on with administration rights.

**Interface Specifications:** Dropdown for choice and submit button for conformation.

**Processing Specifications:** This user selects Restore data. User is given a location and can chose to load data to the DB.

**Database Requirements:**

**Postcondition(s):** Database successfully restored.

## **6. Performance Analysis**

- Upon analysis of the system hardware specs, we have concluded that PHP and MySQL would be able to run on their system.
- Users must be able to navigate on the internet using an internet browser to access the system.
- Data Stores include a MySQL database for the aspects of the planning and development part of the project. There will be a separate table for users, user passwords, and privilege level as well as the online survey which is composed of 4 additional tables that will handle the survey generation and user responses.

## **7. Feasibility and Resource Estimates**

The resources required to build, operate, and maintain this system are detailed as follows:

- Direct Six Software Solutions will provide all computer resources necessary to build the system. Our system is currently being built on Athena here at CSUS.
- ITCC will provide the Internet Information Services (IIS) web server capable of hosting PHP pages, as well as a MySQL Server for the database
- Users are responsible for providing their own computers with sufficient Internet programs to view and interact with the system.
- ITCC will provide all necessary servers to host web pages, database, and all other files required by the system.

## 8. Software Requirements Traceability Matrix

Use Case			SDS Section		SRS Section
Take Survey			5.2.1		3.1.1
Login			5.2.2		3.1.2
Logout			5.2.3		3.1.3
Generate Report			5.2.4		3.1.4
Print Reports			5.2.5		3.1.5
Creating Custom Reports			5.2.6		3.1.6
Create Data			5.2.7		3.1.7
Read Data			5.2.8		3.1.8
Update Data			5.2.9		3.1.9
Remove Entry			5.2.10		3.1.10
Reset Table			5.2.11		3.1.11
Remove User			5.2.12		3.1.12
Reset Password			5.2.13		3.1.13
Update User			5.2.14		3.1.14
Create User			5.2.15		3.1.15
Back Up			5.2.16		3.1.16
Restore Data			5.2.17		3.1.17

Table 8.1 – a requirements traceability matrix mapping from the SDS to the SRS.

## 9. APPROVALS

This section contains signatures used to indicate approval of and agreement to the conditions and commitments contained in the software proposal.

### Team Members:

---

Dave Diel

---

Jared Arbaugh, Chair

---

Khao Saetarn

---

Ben Tovar

---

Kou Vang

---

Ben Mackin, Project Manager

### Faculty Advisor:

---

Dr. Jiang, Computer Science Department

**APPENDIX A  
DATA DICTIONARY**

<b>Table Name</b>	<b>Data Element</b>	<b>Description</b>	<b>Data Type(Length)</b>
<b>tblContact</b>	fname	Contact First Name. <b>Primary Key</b>	Variable Character(15)
	lname	Contact Last Name. <b>Primary Key</b>	Variable Character(15)
	rname	Reservation Name. <b>Primary Key</b>	Variable Character(15)
	title	Contact's Title	Variable Character(20)
	phone	Contact's Phone Number	Variable Character(12)
	Email	Contact's Email Address	Variable Character(50)
	id	Contact ID	Interger(11)
	Street	Contact's Street Address	Variable Character(15)
	City	City	Variable Character(15)
	County	County	Variable Character(15)
	Zip	Zip Code	Numeric(10)
<b>tblReservation</b>	Acres	Acres of The Land	Variable Character(15)
	ranchname	Ranch Name. <b>Primary Key</b>	Variable Character(50)
	affiliation	Affiliation of The Reservation.	Variable Character(50)
	directions	Direction to The Tribe	Variable Character(100)
	Population	Population of The Reservation	int(11)
	Location	Location of The Reservation	Variable Character(20)

	Street	Reservation's Street Address	Variable Character(15)
	City	City	Variable Character(15)
	County	County	Variable Character(15)
	Zip	Zip Code	Numeric(10)
<b>tblRoads</b>	ranchname	Ranch Name. <b>Primary Key</b>	Variable Character(50)
	roadname	Roads Name. <b>Primary Key</b>	Variable Character(20)
	roadtype	Type of The Roads	Variable Character(11)
	r_condition	Condition of The Roads	Variable Character(30)
	MilesTo	The Length of the Road	Variable Character(3)
<b>tblPlans</b>	ranchname	Ranch Name. <b>Primary Key</b>	Variable Character(50)
	plantype	Type of The Plans <b>Primary Key</b>	Variable Character(20)
	Note	Note Describe The Plans	Variable Character(100)
<b>tblWaterways</b>	ranchname	Ranch Name. <b>Primary Key</b>	Variable Character(50)
	watename	Water Name <b>Primary Key</b>	Variable Character(20)
	watertype	Type of The Plans	Variable Character(11)
	milesto	The Length of the Road	Numeric(10)
	Note	Note Describe The Water	Variable Character(100)
<b>tblCcontactinfo</b>	rname	Ranch Name. <b>Primary Key</b>	Variable Character(50)
	Street	Contacto's Street Address	Variable Character(15)
	City	City	Variable

			Character(15)
	County	County	Variable Character(15)
<b>tblrcontactinfo</b>	rname	Ranch Name. <b>Primary Key</b>	Variable Character(50)
	Street	Contactoꝝ's Street Address	Variable Character(15)
	City	City	Variable Character(15)
	County	County	Variable Character(15)
<b>tbluLogin</b>	uname	User Name.	Variable Character(20)
	psswd	User Password	Variable Character(20)
	level	User Permission Level	int(11)
<b>tblsurveyCatagory</b>	qId	Question ID. <b>Primary Key</b>	auto_increment
	qText	The Content of The Questions	Variable Character(255)
<b>tblsurveyQuestions</b>	qId	Question ID. <b>Primary Key</b>	auto_increment
	catID	Category ID	int(11)
	qText	The Content of The Questions	Variable Character(255)
	type	Survey Description	Variable Character(20)
<b>tblsurveyAnswers</b>	qId	Question ID. <b>Primary Key</b>	int(11)
	aID	answers ID.	auto_increment
	aText	The Content of The Answers	Variable Character(255)
<b>tblsurveyResponse</b>	type	Survey Description	Variable Character(20)
	sId	Response ID.	int(11)

		<b>Primary Key</b>	
	qId	Question ID. <b>Foreign Key</b>	int(11)
	aID	answers ID. <b>Foreign Key</b>	int(11)

## APPENDIX B

### Listing of all attributes with associated characteristics.

#### uLogin

Field	Type	Null	Key	Default	Extra
uname	varchar(20)	YES		NULL	
psswd	varchar(20)	YES		NULL	
level	int(11)	YES		NULL	

#### Ccontactinfo

Field	Type	Null	Key	Default	Extra
rname	varchar(50)	NO	MUL		
street	varchar(20)	YES		NULL	
county	varchar(25)	YES		NULL	
zipcode	varchar(11)	YES		NULL	

#### rcontactinfo

Field	Type	Null	Key	Default	Extra
rname	varchar(50)	NO	MUL		
street	varchar(20)	YES		NULL	
county	varchar(25)	YES		NULL	
zipcode	varchar(11)	YES		NULL	

#### Reservation

Field	Type	Null	Key	Default	Extra
ranchname	varchar(50)	NO	PRI		
affiliation	varchar(50)	YES		NULL	
directions	varchar(100)	YES		NULL	
population	int(11)	YES		NULL	
acres	int(11)	YES		NULL	

#### Roads

Field	Type	Null	Key	Default	Extra
ranchname	varchar(50)	NO	PRI		
roadname	varchar(20)	NO	PRI		
milesto	varchar(3)	YES		NULL	
roadtype	varchar(11)	YES		NULL	
r_condition	varchar(30)	YES		NULL	

## Plans

Field	Type	Null	Key	Default	Extra
ranchname	varchar(50)	NO	PRI		
plantype	varchar(20)	NO	PRI		
note	varchar(100)	YES		NULL	

## Waterways

Field	Type	Null	Key	Default	Extra
ranchname	varchar(50)	NO	PRI		
watername	varchar(20)	NO	PRI		
milesto	varchar(3)	YES		NULL	
watertype	varchar(11)	YES		NULL	
note	varchar(100)	YES		NULL	

## Contact

Field	Type	Null	Key	Default	Extra
fname	varchar(15)	NO	PRI		
lname	varchar(15)	NO	PRI		
rname	varchar(50)	NO	MUL		
title	varchar(20)	YES		NULL	
id	int(11)	YES		NULL	
phone	varchar(12)	YES		NULL	
email	varchar(50)	YES		NULL	

## surveyCatagory

Field	Type	Null	Key	Default	Extra
catID	int(11)	NO	PRI	NULL	auto_increment
catText	varchar(25)	YES		NULL	

## surveyQuestions

Field	Type	Null	Key	Default	Extra
qID	int(11)	NO	PRI	NULL	auto_increment
catID	int(11)	YES	MUL	NULL	
qText	varchar(255)	YES		NULL	
type	varchar(20)	YES		NULL	

surveyAnswers

Field	Type	Null	Key	Default	Extra
qID	int(11)	NO	MUL		
aID	int(11)	NO	PRI	NULL	auto_increment
aText	varchar(255)	YES		NULL	

surveyResponse

Field	Type	Null	Key	Default	Extra
sID	int(11)	NO	PRI	0	
qID	int(11)	YES	MUL	NULL	
aID	int(11)	YES	MUL	NULL	