A PROTOTYPE SYSTEM FOR AN ONLINE JOURNAL MANAGEMENT

Sundeep Chandran
B.E., Anna University, Tamil Nadu, India, 2008

PROJECT

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

at

CALIFORNIA STATE UNIVERSITY, SACRAMENTO

SPRING
2012
A PROTOTYPE SYSTEM FOR AN ONLINE JOURNAL MANAGEMENT

A Project

by

Sundeep Chandran

Approved by:

__________________________________, Committee Chair
Meiliu Lu, Ph.D.

__________________________________, Second Reader
Ying Jin, Ph.D.

____________________________
Date
Student: Sundeep Chandran

I certify that this student has met the requirements for format contained in the University format manual, and that this project is suitable for shelving in the Library and credit is to be awarded for the Project.

__________________________, Graduate Coordinator
Nikrouz Faroughi, Ph.D. Date

Department of Computer Science
Abstract

of

A PROTOTYPE SYSTEM FOR AN ONLINE JOURNAL MANAGEMENT

by

Sundeep Chandran

Currently, most of the hard copy journals are going online and becoming E-Journals. The advantages of online journals makes the work of both creator and target audience easier in terms of finance, revenue, time spent, global reach and most importantly the ease of communication.

“Calaveras Station Literary Journal”, a student organization related to English department of the California State University, Sacramento has annual publication consisting of wide variety of topics such as poetry, fiction, non-fiction, critical analysis and poem. Student organization entertains the submissions from students in person and through personal email in most of the cases for publishing in the annual edition.

The objectives of this project are replacing the static web pages of the “Calaveras Station Literary Journal” and develop a prototype in the form of web application facilitating online services such as remote submission, editors notifying selected submissions through inbuilt email, users archiving/downloading/viewing previous year annual editions and administrators adding/deleting the members according to the requirements.

The database methodology used in this project for retrieving the journal information is star schema of data mart. The technologies used for implementation of static web pages...
are CSS, XHTML and the design logo is implemented with Adobe Photoshop. Static web pages are deployed in the Information Resource Technology web server belonging to California State University, Sacramento. Prototype of Journal Management System is simulated using WAMP stack that contains Apache web server, MySQL back end database and PHP server side scripting language. Prototype system can be deployed in WAMP compatible hosting server.

______________, Committee Chair
Meiliu Lu, Ph.D.

_____________________________
Date
ACKNOWLEDGMENTS

I owe my sincere thanks to my project advisor, Dr. Meiliu Lu who guided me on different phases of the project. I am thankful to Dr. Ying Jin to be my secondary reader and review my project. I am thankful to my friends who provided their valuable feedback in implementing my project. Lastly, I thank my parents who gave me an opportunity to do my masters coursework.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Acknowledgments</th>
<th>vi</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>x</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xi</td>
</tr>
</tbody>
</table>

# Chapter

1. INTRODUCTION ........................................................................................................ 1
   1.1 Motivation .......................................................................................................... 1
   1.2 Purpose ................................................................................................................ 2
   1.3 Overview of the report ......................................................................................... 2

2. BACKGROUND AND TECHNOLOGY ........................................................................ 4
   2.1 Introduction ........................................................................................................ 4
   2.2 CSS .................................................................................................................... 4
   2.3 Adobe Photoshop ............................................................................................... 5
   2.4 WAMP ............................................................................................................... 5
      2.4.1 PHP ........................................................................................................... 6
      2.4.2 MYSQL ...................................................................................................... 8
      2.4.3 APACHE ................................................................................................... 9

3. USER REQUIREMENTS AND ANALYSIS ................................................................ 10
   3.1 Introduction ........................................................................................................ 10
   3.2 User Requirements ............................................................................................ 11
   3.3 System Requirements ......................................................................................... 12
3.4 Scope of the Project ................................................................. 13

4. SYSTEM DESIGN ............................................................................ 14

4.1 Work Flow .................................................................................. 14

  4.1.1 Static Work Flow ................................................................. 15

  4.1.2 Dynamic Work Flow .......................................................... 15

4.2 System Architecture ................................................................. 16

  4.2.1 Presentation Logic .............................................................. 17

  4.2.2 Business Logic ................................................................. 17

  4.2.3 Data Logic ........................................................................ 18

4.3 Database Modeling .................................................................... 18

  4.3.1 Dimensional Model ............................................................... 18

    4.3.1.1 Fact Table ................................................................. 18

    4.3.1.2 Dimension Table ........................................................ 19

    4.3.1.3 Star Schema .............................................................. 19

    4.3.1.4 Snow Flake Schema .................................................. 19

    4.3.1.5 Data mart for Journal Management System ............. 20

4.4 User Interface ........................................................................... 22

  4.4.1 Original User Interface .......................................................... 22

  4.4.2 Current User Interface ........................................................ 24

5. IMPLEMENTATION .......................................................................... 27

5.1 Configuring WAMP ................................................................. 27

5.2 Members Module ....................................................................... 28
5.2.1 View Previous Year Annual Editions .............................................30
5.2.2 Download Previous Year Annual Editions .................................31
5.2.3 Upload Current Year Submissions ..............................................32
5.3 Editors Module ...................................................................................34
  5.3.1 View/Download Submissions .....................................................35
  5.3.2 Email to Applicants ..................................................................35
  5.3.3 Selected Submissions to Database .............................................36
5.4 Administrator Module .........................................................................36
  5.4.1 Add/Delete Members and Current Submissions .........................37
  5.4.2 View/Download Selected Submissions ......................................38
  5.4.3 Upload Current Annual Edition to Database ..............................39
6. TESTING ..................................................................................................40
  6.1 Functional Test Cases .......................................................................40
    6.1.1 Registration Function Test .......................................................40
    6.1.2 Login Function Test .................................................................41
    6.1.3 Upload Function Test ...............................................................42
    6.1.4 Editor Function Test .................................................................43
    6.1.5 Administrator Function Test ....................................................44
7. CONCLUSION AND FUTURE ENHANCEMENTS ..............................46
Appendix Project Source Code: Index ...................................................48
Bibliography ..................................................................................................50
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Tables</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Table 1 Register Function Test</td>
<td>41</td>
</tr>
<tr>
<td>2. Table 2 Login Function Test</td>
<td>42</td>
</tr>
<tr>
<td>3. Table 3 Upload Function Test</td>
<td>43</td>
</tr>
<tr>
<td>4. Table 4 Editor Function Test</td>
<td>44</td>
</tr>
<tr>
<td>5. Table 5 Administrator Function Test</td>
<td>45</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Figure 1 Logo of Calaveras Station Literary Journal</td>
<td>5</td>
</tr>
<tr>
<td>2. Figure 2 PHP Interaction with Database</td>
<td>7</td>
</tr>
<tr>
<td>3. Figure 3 System Work Flow</td>
<td>14</td>
</tr>
<tr>
<td>4. Figure 4 Single Tier Architecture Diagram</td>
<td>17</td>
</tr>
<tr>
<td>5. Figure 5 Star Schema</td>
<td>19</td>
</tr>
<tr>
<td>6. Figure 6 Snow Flake Schema</td>
<td>20</td>
</tr>
<tr>
<td>7. Figure 7 Data mart Star Schema</td>
<td>21</td>
</tr>
<tr>
<td>8. Figure 8 Original User Interface</td>
<td>23</td>
</tr>
<tr>
<td>9. Figure 9 Current User Interface</td>
<td>24</td>
</tr>
<tr>
<td>10. Figure 10 Sale Interface</td>
<td>24</td>
</tr>
<tr>
<td>11. Figure 11 Staff Interface</td>
<td>25</td>
</tr>
<tr>
<td>12. Figure 12 Contact Interface</td>
<td>25</td>
</tr>
<tr>
<td>13. Figure 13 News Interface</td>
<td>25</td>
</tr>
<tr>
<td>14. Figure 14 Registration User Interface</td>
<td>29</td>
</tr>
<tr>
<td>15. Figure 15 Login User Interface</td>
<td>30</td>
</tr>
<tr>
<td>16. Figure 16 Members User Interface</td>
<td>31</td>
</tr>
<tr>
<td>17. Figure 17 Download User Interface</td>
<td>32</td>
</tr>
<tr>
<td>18. Figure 18 Upload User Interface</td>
<td>33</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

Calaveras Station Literary Journal, a student organization belonging to the English Department of the California State University, Sacramento has annual publication in the month of May consisting of selected submissions from topics such as non-fiction, fiction, poetry and critical analysis. Journal entertains submissions either manually or through applicants personal email address. Selections are done by the editors responsible for the appropriate topics. Selected submissions appear in the annual publication.

1.1 Motivation

The main motivation behind this project is my voluntary involvement as a web editor for Calaveras Station Literary Journal, a student organization belonging to the English Department of the California State University, Sacramento. My role as a web editor is to update journal information in their static website. In order to improve the functionality of the website by good user interface, project is undertaken. In addition to good user interface following factors influenced me to do this project

- To create a prototype for journal management enabling dynamic interaction between users and server by employing datamart schema in back end database.
- To serve as a case study for Data warehousing coursework.
- To learn PHP, MySQL in WAMP stack.
1.2 Purpose

The main purpose of this project is to replace static web pages in the main server and create a prototype for Calaveras Station Literary Journal and other similar journal application using data mart schema in back end database thereby having the following advantages

- Prospective Applicants can submit their submissions remotely through online.
- Editors related to the appropriate topics able to view the appropriate topic submissions.
- Editors sending the email to selected applicants without logging into their personal email system.
- Editors able to send the selected topics to the executive editor for annual publications.
- Applicants able to view and download the previous year annual editions.
- Applicants able to send their feedback in the contact form without logging into their personal email system.

1.3 Overview of the report

Overview of the report provides details about various chapters and their purpose. In chapter 2, background and technologies used for implementing the project will be discussed. In chapter 3, user requirements, system requirements and scope of the project will be discussed. In chapter 4, system workflow, system architecture, database modeling and user interface will be discussed. In chapter 5, configuration of the WAMP stack,
implementation of the member module, editor module and administrator module will be discussed. In chapter 6, functional test cases related to registration, login, upload, editor and administrator will be discussed. In chapter 7, future enhancements of the developed prototype will be discussed. In chapter 8, conclusion of the project will be discussed. Finally, source code of different functionalities of the system will be discussed.
Chapter 2

BACKGROUND AND TECHNOLOGY

2.1 Introduction

The original web pages of the Calaveras Station Literary Journal had only static pages with tables, non user friendly interface and absence of navigation links. Changes to the journal were very tiresome because of the absence of Cascading Style sheets. Original web pages were replaced by div tags and Cascading Style Sheets for doing changes to the original website in a convenient manner. Adobe Dreamweaver is used for creating the logo for the website. In addition to the former changes, a prototype for the journal is built with features such as remote submission, downloading previous year editions, editors selecting the submissions and administrator adding/deleting the members by using PHP server side scripting language in a WAMP stack and star schema of data mart is used for the back end MySQL database.

2.2 CSS

Cascading Style Sheets is a style sheet language used for adding styles such as font, size, spacing and color to the web documents and separates the document content with document presentation [1]. It has the following advantages

- By making one change in CSS, relevant changes in the related HTML pages are made.
- HTML files size gets reduced and thereby file transfer size also gets reduced.
• It will be helpful for search engines to search for the content because CSS consists of more content than code.
• It makes way for loading the website to the server faster.

2.3 Adobe Photoshop

Adobe Photoshop is a proprietary program developed and maintained by Adobe Systems Incorporated for graphic related purposes. The original Calaveras Station Literary Journal website was without a logo. Below JPG Image is the logo for the current journal. It was created using Photoshop

![Logo of Calaveras Station Literary Journal](image)

Figure 1: Logo of Calaveras Station Literary Journal

The other main advantages of Adobe Photoshop are altering the colors, changing the backgrounds; apply artistic layers onto photos, image manipulation, etc.

2.4 WAMP

WAMP is a windows based web development package consisting of Apache web server, MySQL database and either Perl or Python or PHP scripting language. It also comes with database GUI called PhpMyAdmin that is used for creating databases, tables with indexes
rather than inserting data manually through command line editor. Though it is windows based package, Linux operating system compatible package called LAMP where L stands for Linux, A for Apache, M for MySQL and P for PHP/Perl/Python. Macintosh operating system compatible is called MAMP where M stands for Mac, A for Apache, M for MySQL and P for PHP/Perl/Python.

The advantages of WAMP are of the following

- Solves the configuration problems of the individually installed packages.
- Most of the web hosting providers have WAMP package.
- It is an open source product
- User friendly GUI for developing a web application.
- It is easily portable to the web hosting server.

In this project, I make use of WAMP for developing the prototype web application.

**2.4.1 PHP**

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a server-side scripting language used for developing dynamic web pages of web based applications [2]. It is one of the first language dynamic server sides scripting language that can embed with the existing HTML file without calling it externally.
The above figure 2 from [3] briefs about the PHP interaction with database.

1. Client enters the request in the web browser in his/her desktop PC
2. Request is passed from browser to Web server (Apache) through internet
3. Web Server recognizes the .php extension and passes the file to the PHP interpreter.
4. PHP interpreter reads the file from disk drive.
5. PHP interpreter runs the file and exchanges the data from the database.
6. PHP interpreter returns the output of the result to the Apache web server.
7. Apache web server in turn forwards the output to the internet.

The main advantages of PHP are

- It is simple to understand. Users knowing C, HTML can easily learn PHP
- It has in built security features in “.ini” file where the user can configure his own security settings according to system requirements.
• It can easily interface with databases such as MySQL, oracle, etc. and with servers such as Apache, IIS, etc.
• It runs fast and does not makes use of lot of system resources.
• It is an open source product thereby it has lot of scope for updated features and one need not spend money for using the scripting language.

2.4.2 MySQL

MySQL is an open source and one of the highly used Relational Database Management System licensed under General Public License. It is owned by Oracle Corporation. It is mostly administered by command line sql commands for storing and retrieving the data from its database. Many third part GUI tools are also used for creating databases, tables and queries such as phpMyAdmin, Toad for MySql, etc.. Stored Procedures, triggers, cursors, views, etc can be implemented using MySQL database. The main advantages of MySQL are

• Its default table format Miasma consumes very little space related to CPU, memory and disk and results increase in performance
• It integrates with PHP easily and used in popular LAMP stack where M denotes MySQL.
• It is used for managing many database and web based applications.
• It is flexible and works in Linux, Unix and Windows operating system.
• It has the option to change the port to overcome security intrusion.
2.4.3 APACHE

Apache HTTP server is open source software developed and maintained by Apache Software Foundation [4]. It is one of the most used web server in the world. Apache modules support server side languages such as PHP, Perl, TCL, Python and authentication schemes such as mod_access, mod_auth, mod_digest, etc... The main advantages of Apache are

- It is easy to administer the server because of the clear instructions in its configuration files
- It is platform independent and can work in Windows, Macintosh, Unix, Linux, FreeBSD operating systems and is portable.
- Users can add their own module to Apache and can meet their requirements.
- Easy to interface with databases
Chapter 3

USER REQUIREMENTS AND ANALYSIS

3.1 Introduction

Requirements analysis is the process of knowing the user requirements and expectations for the desired product by taking into the consideration of all the stakeholders involved and thereby avoiding ambiguity and conflicts [5]. Three steps of requirement analysis are eliciting, analyzing and recording the requirements [6].

- Eliciting Requirements- It is undertaken by taking into the consideration of the stakeholders involved for the system to be developed
- Analyzing Requirements- After gathering the requirements, they are subjected to be analyzed for its feasibility.
- Recording Requirements - After figuring out the requirements from the analysis, they needed to be recorded in form of use cases and user stories.

In order to gather the requirements, there was an interview with Calaveras Station Literary Journal stakeholders. I was able to elicit the following requirements

- Coming up with a very good user interface
- Replacing the tables in the web pages with classes and Cascading Style Sheets.
- Coming up with back end database for the front end web application.

It was followed up with analyzing the requirements where we decided to do the first two requirements and a prototype for third requirement.
Finally, use cases were created for recording the agreed requirements. Use cases for various roles will be discussed in User Interface Specification section.

3.2 User Requirements

User requirements are requirements specified by the customer in starting stages of the Software Development Life Cycle.

The following are the list of user requirements for Prototype System for Online Journal Management system

1. Prospective applicants able to view/download/upload the submissions
   a. Applicants needed to be registered by choosing their username, password, first name, last name and email-id in the registration page.
   b. After applicant registers, he/she needs to be redirected to the members page for choosing
      i. Download Previous Annual Editions
      ii. View Previous Annual Editions
      iii. Upload new submissions by choosing the relevant topic - Users will be redirected to the upload page for uploading word/pdf/text files.

2. Editors of the Journal need to select the submissions inserted it into the database.
   a. Editors will be able to view only the relevant topics allocated to them.
   b. Editors should send email to the selected applicant regarding their selection.
c. Editors should be able to both download and view the submission.

3. Administrator will have the privileges to add/delete the existing users, files from the database.
   a. Administrator should upload the annual edition into the database.
   b. Administrator should be able to delete the current submissions in the database.
   c. Administrator should be able to view the selected submissions from all the topics.

4. Contact form for sending the users feedback to the journal email address.

3.3 System Requirements

System requirements play an important role for a software system that is built and ready to be deployed on a server with required software resources. The following are the system requirements for “Prototype System for Online Journal Management System”

- WAMP package is needed for the deploying server.
- Currently, Calaveras Station Literary Journal static web pages are deployed on the IRT server maintained by Academic Information Resources Center, California State University, Sacramento.
- Prototype System for Online Journal Management can be deployed using WAMP stack in IRT server and other hosting providers available in the open market.
- Online users can access the Calaveras Station Literary Journal through standard browser in their desktop/laptop/work station.
3.4 Scope of the Project

Project scope defines the set of deliverables and objectives of the project by taking into time, budget and other important factors into the consideration [7].

Application consists of the following deliverables

- Remote submissions through web
- Is not limited to only California State University students.
- It can be used as a case study for the Data warehousing course.
- Data mart Applicability to the back end database to know the no of submissions made per topic.
- It provides good scalability for increase in the topics.
- It is built by open source software packages such as
  - PHP- Open source and most widely used scripting language for the web development. It is one of the most used scripting languages in well known Content Management Systems such as Joomla, Drupal, WAMP, LAMP, and XAMP.
  - Apache – Widely used web server for applications deployed on the web and mostly used by the web hosting providers.
  - MySql- Widely used open source back end database tool which is easy to learn, store/retrieve data back and forth and to maintain.
Chapter 4

SYSTEM DESIGN

Design of the system plays an important role after analyzing the requirements. In this chapter, work flow of the system, system architecture and database modeling will be discussed.

4.1 Work Flow

![System Work Flow Diagram]

Figure 3 System Work Flow
The above work flow diagram can be separated into following two parts

- Static Work Flow
- Database Driven Dynamic Work Flow

### 4.1.1 Static Work Flow

In the static work flow, anyone who has access to the internet can view the static web pages. Here general information about the journal is displayed. It does not need any data interaction with the database. The following information about the journal can be viewed.

- About Calaveras Journal
- About the current working staff
- About the sale of the upcoming journals
- Gallery photos of the release party
- General Contact Information

The above information is hosted in the Information Resource Technology server of California State University, Sacramento.

### 4.1.2 Dynamic Work Flow

In the dynamic work flow, client server and database interaction dynamic web pages are present. Work flow is according to the following

- Firstly, new users need to register
• After registering, users can login and have the following options
  • View previous year annual editions
  • Download the previous year editions
  • Upload the upcoming annual edition related to appropriate topics.
• After users upload the submissions, editors can do the following
  • View or Download the users uploaded submissions
  • Judge the submissions and send email to applicants
  • Insert the selected submissions to the database.
• Finally, administrators can do the following
  • Able to add or delete the existing users in the database.
  • Delete the current submissions in the database
  • View or download the selected submissions by the editor
  • After assembling the data from various files manually, upload the new annual edition to the database.

Dynamic work flow functionalities are simulated in the WAMP server.

4.2 System Architecture

A prototype System for Online Journal Management follows single tier architecture. In single tier architecture, presentation logic, business logic and data tier logic are all combined into a single component.
The above single tier architecture diagram [8] is separated into following.

4.2.1 Presentation Logic

In software development, presentation logic is concerned with how business objects are displayed to users of the software [9]. In presentation logic, client can post the request in the existing user interface and send the request to the server. HTML and CSS are the languages used to display the static content of prototype web application.

4.2.2 Business Logic

Business logic is the set of rules performing the behind the scene tasks by handling information exchange between database and user interface [10]. PHP scripting handles the business logic by interacting with the database server and user interface in the prototype web application.
4.2.3 Data Logic

DML is a part of Data Access Logic that is used for inserting/deleting/updating the database in the database server. MySQL is the backend database used in the prototype web application.

4.3 Database Modeling

A database model provides a theoretical idea about the organization, retrieval and manipulation of the data across the database [11]. In prototype web application, dimensional model with star schema will be discussed.

4.3.1 Dimensional Model

Dimensional modeling is a type of logical design where its data is organized for the faster access [12]. Here the data is stored in two kinds of tables namely

- Fact Table
- Dimension Table

4.3.1.1 Fact Table

Fact table consists of multiple foreign key associated with appropriate dimensions key. In addition to the multiple foreign keys, it can also contain its own attribute.
4.3.1.2 Dimension Table

Dimension table consists of a primary key that is referenced to part of the multiple keys in the fact table. In most of the cases, dimension tables are more than fact table.

Dimensional model can either of the following schemas

- Star Schema
- Snow Flake Schema

4.3.1.3 Star Schema

It is simple type of schema that can contain either one or more fact tables referencing many dimensions table. It is used for the databases that are designed to handle limited requests. The following diagram illustrates the star schema.

![Star Schema Diagram]

Figure 5 Star Schema

4.3.1.4 Snow Flake Schema

It is complex than star schema. It consists of one or more fact tables referencing multiple dimension tables wherein each dimension tables in turn references to other dimension tables. The following diagram illustrates the snow flake schema.
The following diagrams details about the star schema for the prototype web application.

4.3.1.5 Data mart for Journal Management System

Data mart is a data repository derived from the data warehouse serving the specific sets of user by providing them the appropriate data [13]. In prototype web application, star schema of data mart is used for knowing the number of submissions made by the applicants. Star schema consists of centralized fact table and set of dimensions table.
Figure 7 Data mart Star Schema

The Figure 7 illustrates star schema of the data mart star schema. Here, QuantityFilesFact acts as a fact table for finding the number of files submitted by the user and the number of files viewed by the moderator. Uses of the following dimensions tables

- **QuantityFilesFact**

  It is the fact table of the star schema that consists of set of foreign keys related to dimension tables such as users, uploaded files, moderators and topics. Here the metric that need to be measured is the number of files.
• **UserDimension**

It has the information related to the user username, password, first name and last name. Here userid acts as a primary key.

• **UploadedfilesDimension**

It has the information about the file uploaded for the submissions. It consists of filename, filetype, filesize, topic_id tells the topic for which the file is uploaded and file path stored in the WAMP server.

• **TopicsDimension**

It consists of topicid as a primary key and topic name.

• **ModeratorsDimension**

It consists of moderator username, password and their email-id. Here moderators_id acts as the primary key and has it references towards the fact table. Topic_id in the table acts as a foreign key to the topics dimension table.

### 4.4 User Interface:

Original and current user interface of the “Calaveras Station Literary Journal” will be discussed.

#### 4.4.1 Original User Interface

Figure 8 is the screenshot version of the original home page of the “Calaveras Station Literary Journal”. Original web page had tables, contrasting background and was lacking with the following features

• Cascading Style Sheets
• Div Tags
• Good User Interface
• Navigation Links

Figure 8: Original User Interface

Original user interface was difficult for the visitors to gather important information about the journal announcements. It was also a difficult task to make a minor change because of the absence of the Cascading Style Sheets, navigation links and div tags. Previous web editors had to edit the content according to the design of the tables. In addition to user interface, online submissions were not provided in the original website.
4.4.2 Current User Interface

Current interface overcomes the drawback of most part of the original “Calaveras Station Literary Journal” by having Cascading Style Sheets, div tags, navigation links and right combination of the background color for the web page.

Figure 9: Current User Interface

The above screenshot is the current home page for the “Calaveras Station Literary Journal”. Home page gives the general information about the journal.

Figure 10: Sale Interface
The above screenshot displays list of sale locations for the journal.

- Executive Editors- Adam Burrell
- Creative Non-Fiction Editors- Verna Dreisbach, Jess Storrs
- Critical Analysis Editors- Kimberly Hudson, Lindsay Snodgrass
- Fiction Editors- Dave Powers, Kyle Hardwick
- Poetry Editors- Aschala Edwards, Ann Wehrman
- Web Master- Sundeep Chandran
- Advisors- Professor Doug Rice
- Design Advisor- Professor Richard Pratt

Figure 11: Staff Interface

The above screenshot lists the staff associated with the journal.

Inquiries regarding submissions and general information should be directed to

- Calaveras Station Literary Journal
  Calaveras Hall 102, English Department Office, California State University
  6000 J Street
  Sacramento, CA 95819

- All views expressed are those of the writers only. First North American serial rights held by Calaveras Station Literary Journal. All individual rights reserved by the individual authors. Guest authors retain all individual rights.

Figure 12: Contact Interface

The above screenshot briefs about the contact information of the journal.

- Journal submissions and deadlines to be announced soon
- Please send your inquiries to poetry.and.crit.cslj@csus.edu

Figure 13: News Interface

The above screenshot briefs about the deadlines about the submissions and email address of the journal to which the enquiries need to be sent.

The above static user interfaces are implemented in the Information Resource Technology server, California State University, Sacramento. In addition to the static user
interface, prototype is developed for client server interactive dynamic web pages and will be discussed in the implementation section.
Chapter 5

IMPLEMENTATION

In this chapter, configuration of the WAMP server and the implementations of all the modules of our prototype system will be discussed in detail.

5.1 Configuring WAMP

WAMP stack is used for simulating the prototype journal web application. WAMP can be installed only in windows operating system. By installing WAMP stack, apache web server, MySQL database and PHP are installed. After installing WAMP, following precautions need to be taken.

- **Local host Port**: Default port of the WAMP server’s local host is 80. If system has either IIS or Skype installed, either one can occupy same port 80 which will be a problem for WAMP. It is recommended to change the port number from 80 to 8080.
- **Httpd.conf**: After changing the port to 8080, apache httpd.conf file need to have the following changes
  - String “Listen 80” need to be changed to “Listen 8080”
  - String “Server Name local host” needed to be changed to “Server Name localhost:8080”
• **Wampmanager.ini**: By going to WAMP installation folder C:\WAMP, replace the “local host” with “localhost: 8080” in “wampmanager.ini” file for three occurrences and wampmanager.tpl for two occurrences.

• **PHPMyadmin**: In PHPMyadmin database management tool, password and privileges for root user need to be set by getting into system tray icon-> MySQL followed by MySQL console to open the console and using the command
  - “update user set password = password<“password”> where user= “root”
  - Flush privileges

• After changing the password navigate to wamp/apps/phpMyAdmin, open config.inc.php in an editor and add the password “password”.

• Finally, restart the WAMP server for the changes to be applied.

5.2 Members Module

In member’s module, new user need to register first followed by login to enter the member’s page for viewing/uploading/downloading the journals.

**Registration:**

New members need to register by getting into Register page. After getting into register page, new members are required to give the following information

• Username

• Password

• First name
- Last name

The screenshot of the register page is given below

![Registration User Interface](image)

Figure 14: Registration User Interface

After the registration, “You have successfully Registered” message pop ups in the browser and username, password, first name and last name are inserted into the mysql database.

**Login:**

Registered users enter their username and password for getting into members page. Username and password entered by the user is checked with the mysql database and if there is a match, users are redirected to members page. If wrong username and password is entered, “Wrong Username or Password” is displayed.
Figure 15: Login User Interface

The above screenshot is the login user interface with username and password details needed to be entered by the user.

In member’s page, registered users can select the following from the dropdown box.

- View Previous Year Annual Editions
- Download Previous Year Annual Editions
- Upload Current Year Submissions

5.2.1 View Previous Year Annual Editions

Here the annual editions of the previous year editions can be viewed by the user. User needed to select the year for viewing the annual edition related to that year. Path to the
files are stored in the mysql database. Annual Edition files are stored in a pdf format for the user to view.

![Members User Interface](image)

Figure 16: Members User Interface

### 5.2.2 Download Previous Year Annual Editions

Here the previous year annual editions can be downloaded locally to the user workstation. Previous year annual editions file path are stored in the mysql database. The following download user interface screenshot shows the list of annual editions sorted by year for download.
5.2.3 Upload Current Year Submissions

Here the users are allowed to submit their submissions for the upcoming annual edition by topics. Currently, prototype web application has 7 topics. Users should select the appropriate topic for the submission need to be made. After selecting the topic, users can upload the files in the following format

- MS-Word
- PDF
- Text Files
After uploading the files, user email and files are stored in the mysql database for editors to judge the submission.

Figure 18: Upload User Interface

The above screenshot shows the topics in a dropdown box for the user to select a particular topic for submitting his submission.
Figure 19: File Upload User Interface

The above screenshot shows the file upload option for appropriate topic and applicant email address. After user uploads the information, path to the files are stored in the mysql database for future retrieval by the editors and administrators.

5.3 Editors Module

In prototype web application, role of editors are limited to only their assigned topic. Here each editor is assigned one topic for selecting the submission to the annual edition journal. Editors are assigned username and password for viewing the submissions made by the user related to the appropriate topic.
Figure 20: Editor User Interface

The above screenshot briefly details about the list of files submitted under topic1 by the applicants. Editors will be able to do the following

5.3.1 View/Download Submissions

By selecting the Download/Open link, editors will be able to both open and download the file submitted by the users. Editors will be able to take a decision after reading the contents in the file.

5.3.2 Email to Applicants

In prototype web application, phpmailer class is used for using gmail as a SMTP server and sending email to the users about their selection. A valid gmail address and password is needed for setting gmail as SMTP server using phpmailer class. In this prototype web application, testingwamp@gmail.com is used as username. By selecting the selected Email/NotSelectedEmail link, editors will be able to send the selection status to the applicant email address displayed. In turn, editors can reply to the applicants email. In this case, editor1 email address is mentioned as editor1@gmail.com.
5.3.3 Selected Submissions to Database

Finally, editors need to send the selected submissions to the mysql database by selecting the SubmissionSelected link thereby administrator can gather all the content of the selected submissions and make it a single file.

5.4 Administrator Module

Administrator has lot of functionalities before the commencement of the edition submission and at the end of the edition in cleaning up the existing data in the database. Administrator can do the following
• Add/Delete Members and Current Submissions
• View/Download Selected Submissions
• Upload Current Annual Edition to Database

5.4.1 Add/Delete Members and Current Submissions

In addition to the new members registering, administrator can add new members and delete the existing members from the database. Current registered users pop up in the delete users section

![Delete Existing Users](image)

Figure 22: Delete User Administrator Interface

The above screenshot represents the members present in the database. Administrator can select the member to delete from the database.

![ADD A New User](image)

Figure 23: Add Member Administrator Interface

The above screenshot represents adding a new member through administrator.
Figure 24: Delete Current Submissions Administrator Interface

The above screenshot displays the current submissions that can be deleted by the administrator after the annual publication.

5.4.2 View/Download Selected Submissions

After Editors select the individual submissions from the applicants and insert it into the database, administrators will be able to both view and download the selected submission by the editors.

Figure 25 View/Download Selected Editions Administrator Interface

The above screenshot shows the list of submissions selected by the editors.
5.4.3 Upload Current Annual Edition to Database

Finally, administrator gathers all the content and makes it into a single pdf file for uploading the current annual edition into the database for its future reference by the journal members.

Please choose an Annual Edition file to upload

Choose File  No file chosen
Upload

Figure 26 Annual Edition File Upload Administrator Interface

The above screenshot shows the feature to upload the file to the database for future reference.
In this chapter, functional test cases of various features of the prototype web application will be discussed.

6.1 Functional Test Cases

Functional testing is a type of black box testing. Here functions are tested by giving the sample input and expecting the desired output.

6.1.1 Registration Function Test:

The following table (Table 1) describes the user registration functional test that focuses on the registration details given by the user for the first time. Here, empty fields are not allowed and partial fields never get inserted into the database.
Table 1: Register Function Test

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Path</th>
<th>Test case</th>
<th>Result</th>
<th>Operational Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>./register.php</td>
<td>Username, password, first name and last name details given</td>
<td>You have successfully registered</td>
<td>Backend database and login page is checked for a new entry.</td>
</tr>
<tr>
<td>2</td>
<td>./register.php</td>
<td>Either of the username, password, first name and last name field details are empty</td>
<td>Field should not be empty. Sorry your file is not registered.</td>
<td>Backend database is checked for a non entry.</td>
</tr>
</tbody>
</table>

6.1.2 Login Function Test:

The following table (Table 2) tests the login functionality of the users. Registered members are redirected to member’s page after matching the identical fields in the database. Username and password fields are not allowed to be empty. Error message pops up informing the user to enter information. If the information is wrong then “Invalid username and password” message is displayed.
Table 2: Login Function Test

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Path</th>
<th>Test case</th>
<th>Result</th>
<th>Operational Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.login.php</td>
<td>Username, password, details given</td>
<td>Redirected to members page</td>
<td>Backend database details are cross checked with login details</td>
</tr>
<tr>
<td>2</td>
<td>.login.php</td>
<td>Either of the username or password field details are empty</td>
<td>Field should not be empty. Sorry you are not logged in</td>
<td></td>
</tr>
</tbody>
</table>

6.1.3 Upload Function Test

The following table (Table 3) tests the upload functionality of the member’s files. Registered members need to upload only Word/PDF/Text files. Otherwise “Please upload only Word/PDF/Text files only” message is displayed. Members need to give their email address in addition to the file upload for storing the record in the database.
### Table 3: Upload Function Test

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Path</th>
<th>Test case</th>
<th>Result</th>
<th>Operational Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>./upload.php</td>
<td>Word file or text file or PDF file uploaded and email address of the applicant entered</td>
<td>The file has been uploaded successfully.</td>
<td>Backend database is checked for the file path and file entry.</td>
</tr>
<tr>
<td>2</td>
<td>./upload.php</td>
<td>Either of the email address or file not uploaded</td>
<td>Enter the email address</td>
<td>Backend database is checked for empty file name and email address</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sorry your file was not uploaded.</td>
<td></td>
</tr>
</tbody>
</table>

### 6.1.4 Editor Function Test

The following table (Table 4) tests the functions of the editors related to their login, sending email to the applicants email address with the selection status and storing the selected submissions in the database.
### Table 4: Editor Function Test

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Path</th>
<th>Test case</th>
<th>Result</th>
<th>Operational Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>./editorlogin.php</td>
<td>Wrong username and password</td>
<td>Incorrect username and password message displayed.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>./editortopic.php</td>
<td>Selected Email</td>
<td>Message sent</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>./editortopic.php</td>
<td>SubmissionSelected</td>
<td>Successfully submitted to the database</td>
<td>Database is checked for a new entry.</td>
</tr>
</tbody>
</table>

### 6.1.5 Administrator Function Test

The following table (Table 5) tests the functions of the administrator login validation, adding and deleting existing members, adding the new topic, deleting the existing current journals and final upload related to the current annual edition.
### Table 5: Administrator Function Test

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Path</th>
<th>Test case</th>
<th>Result</th>
<th>Operational Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>./adminlogin.php</td>
<td>Wrong username and password</td>
<td>Incorrect username and password message displayed.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>./admin.php</td>
<td>Deleting Existing Users/current</td>
<td>User/submission not appearing in the list box</td>
<td>User/submission getting deleted from the database.</td>
</tr>
<tr>
<td>3</td>
<td>./admin.php</td>
<td>Add a new user/topic</td>
<td>New user/topic added to the list box</td>
<td>Database is checked for a new entry.</td>
</tr>
<tr>
<td>4</td>
<td>./admin.php</td>
<td>Upload new annual Edition</td>
<td>The file has been uploaded successfully.</td>
<td>Database is checked for a new entry.</td>
</tr>
</tbody>
</table>
Chapter 8
CONCLUSION AND FUTURE ENHANCEMENTS

This project aims to replace the static user interfaces of the “Calaveras Station Literary Journal” with a good and user friendly interface. It makes the role of the future web editors much easier because of the Cascading Style sheets and <div> tags. Currently, static pages replacement with good user interface is implemented in the “Information Resource Center” server belonging to California State University, Sacramento. In addition to the static interfaces, a prototype is developed for client server interaction with database driven dynamic web pages using PHP. Current prototype can be easily integrated with content management systems like joomla, drupal and word press. Back end database uses star schema of data mart employing centralized fact table and more dimensions table. This can be used as a case study for students pursuing a course in data warehousing course such as CSC 177 in California State University, Sacramento.

This project has a lot of room for future enhancements. There can be an automation methodology to merge the contents related to the selected submissions instead of administrator manually adding the contents of individual selected submissions. There can be a mobile version of the web application in the form of android/iphone app because of advent of mobile revolution and good portability. Integration of the web application with social media services like facebook will increase the target audience of quality submissions.
The developed project gave me a good exposure to the practical implementation of various phases of Software Development Lifecycle and opportunity to learn the related technologies.
APPENDIX

Project Source Code: Index

```html
<! 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" />
<title>Calaveras Station</title>
<link href="css/layout.css" rel="stylesheet" type="text/css" />
<style type="text/css">
</style>
</head>
<body>
<div id="pamwrapper">
<div id="logo"><a href="index.html"><img src="images/sample2.jpg" alt="mystic" border="0" /></a></div>
<div id="navigation"><a href="index.html" class="style1">Home</a> | <a href="news.html" class="style1">News</a> | <a href="staff.html" class="style1">Staff</a> | <a href="sale.html" class="style1">Sale</a> | <a href="photogallery.html" class="style1">Gallery</a> | <a href="contacts.php" class="style1">Contact Us</a>
<div id="headerImg"></div>
<div id="bodyArea">
<p>Calaveras Station Literary Journal provides publication opportunities for all currently enrolled students writing short fiction, poetry, creative nonfiction, and critical analysis. Students work as editors and are involved in soliciting and evaluating submissions for possible publication. The journal is published annually, in conjunction with design students, each May and is presented to the public at a release party at which time published students read their works. We work closely with Collective Reading Series.</p>
<p>Except for guest authors, this publication contains only the original work of students who were enrolled at CSUS during the semester their work was selected for publication. Students from all departments at CSUS are encouraged to submit original poems, stories, critical and expository essays, and creative nonfiction.</p>
</div>
</div>
</body>
</html>
```
BIBLIOGRAPHY

1. General Information about Cascading Style Sheets, [Online].


2. General Information about PHP, [Online].


3. PHP Interaction with Database, [Online]


4. General Information about Apache HTTP Server, [Online]


5. General Information about Requirements analysis, [Online].

   Available: http://searchsoftwarequality.techtarget.com/definition/requirements-analysis, accessed February 2012

6. Steps Of Requirement Analysis, [online]

7. **Project Scope, [Online]**


8. **Single Tier Architecture Diagram, [Online]**


9. **General Information about Presentation Logic, [Online].**


10. **General Information about Business Logic, [Online].**

    Available: [http://www.investopedia.com/terms/b/businesslogic.asp#axzz1rfQd5XmN](http://www.investopedia.com/terms/b/businesslogic.asp#axzz1rfQd5XmN), accessed March 2012

11. **Wikipedia, General Information about Database Model, [Online].**


12. **General Information about Dimension Model, [Online].**
