Web Programming Language Project

Project Final Report

Group Name: TopSpaceHL

Group Member: Li He

Yuchuan Liu

# Project Application

TopSpaceHL Sports News Website.

# Architecture Diagram

Macintosh HD:Users:leoyuchuan:Google Drive:UTDallas:14F-CS6301-Web Programming Languages:Project:Project-Share:Architecture Diagram V1.1.pdf

# Module Description

1. Client Browser: This component is end user who access our website through browser.
2. Load Balancer: Load Balancer is a webserver, which include user interface and requests to Web Server. We host our Load Balancer locally.
   1. Technology Used & Reason
      1. Memcached: Memcached is a Free & open source, high-performance, distributed memory object caching system. It is simple to use and powerful.
      2. HTTP\_Request2: HTTP\_Request2 is a rewrite of HTTP\_Request package for PHP5. It provides easy way to perform HTTP request.
      3. AJAX: AJAX is easy use to perform asynchronous request to our Web Server. We use AJAX to perform normal login, Facebook login, and register. Considering that AJAX exposes code to client, we implement other functionalities using PHP.
      4. PHP: PHP is a popular general-purpose scripting language that is especially suited to web development. It is Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world. It could do Server Side Programming, and could communicate with our Web Server in a safe manner.
      5. Bootstrap: Bootstrap is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It is easy to use and create very nice user interface.
      6. Facebook Single Sign-On: Facebook Single Sign-on is a secure and easy way for people to log in to our website.
      7. Sina Weibo Single Sign-On: This is alternative of Facebook Single Sign-On. We firstly implement Sina Weibo SSO, but we can’t access to it locally. However we host it on a verified domain and it can work accordingly there.
3. Services & Functionalities: These services and functionalities are hosted on our Web Server. They are responsible for manipulating database and response XML file according to requests.
   1. Technology Not Used
      1. Node.js: Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. Since there are no sufficient documents for node.js to achieve our requirements and we are not so familiar with node.js. We didn’t choose to use it.
      2. ASP: ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript. You can also create Web APIs, mobile sites, use real-time technologies like Web Sockets and more! We didn’t use this since we thought PHP is easier to use and we familiar with PHP more.
      3. JSP: Java Server Pages (JSP) technology provides a simplified, fast way to create dynamic web content. JSP technology enables rapid development of web-based applications that are server-independent and platform-independent. We didn’t use this since we thought PHP is easier to use and we familiar with PHP more.
   2. Technology Used & Reason
      1. PHP: As described in previous section.
      2. Propel ORM: Propel is a free, open-source (MIT) object-relational mapping toolkit written in PHP. It is also an integral part of the PHP framework Symfony and was the default ORM up to, and including version 1.2. We use it since we use PHP for our Server Side Programming.
4. Web Server: Web server hosts our services. We host our web server & Load Balancer locally. For hosting technologies, we have following candidates and our choice.
   1. Hosting
      1. Technology Considered & Not Used
         1. Tomcat: Apache Tomcat is an open source software implementation of the Java Servlet and Java Server Pages technologies. We didn’t choose this since we choose to use PHP.
         2. DreamHost Web Hosting Services: DreamHost offers kinds of Web Hosting plans. Our shared-server plan can’t run Memcached in their server. If we want to use Memcached, it’s a little expensive. Then we decided to host our Server locally.
      2. Technology Used & Reason
         1. Apache2: Apache2 provides a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. We choose this because it can easily integrate PHP component for our server. Meanwhile, it is easy to configure, setting up compression, certificate and etc. We host two of our Web Server locally.
   2. Other Technologies Used
      1. Composer: Composer is a tool for dependency management in PHP. It allows you to declare the dependent libraries your project needs and it will install them in your project for you. It is easy to use and can install Propel ORM easily.
5. Database Sharding: Our database sharding based on region.
   1. Technology Considered & Not Used
      1. Oracle: UTDallas CS Department offers Oracle Database. However, UTD only offers one database per student and we need this database for other courses. So choose alternative plan.
   2. Technology Used & Reason
      1. MySQL: MySQL is the world's most popular open source database. Whether you are a fast growing web property, technology ISV or large enterprise, MySQL can cost-effectively help you deliver high performance, scalable database applications. We choose this since it is easy to deploy on Ubuntu, and also our DreamHost Shared-Server offers MySQL database.
   3. Hosting
      1. In order to achieve sharding in our web application. We choose to host one database locally and one on DreamHost.
         1. Locally: We deploy one MySQL database locally.
         2. DreamHost: We host a MySQL database on DreamHost Shared-Server.

# Services & Functionalities

## Brief Description

In this section, we will describe our services and functionalities on Web Server and Load Balancer separately. Our Web Server is in SOA, it support lots of services that can be accessed from all kinds of platforms. Our Load Balancer is more like Web Interface integrates with functionalities to access our services on Web Server.

## Web Server

### Login Service

1. Check Facebook Login: if user login with Facebook Information, this web service will fetch corresponding information in our database to check if this Facebook user ever register with our application. And return XML file with User information in our database or Message that Facebook User not found.
2. Check Normal User Login: Request this service with User Information, this service will verify user information in database and return XML with verification information. (Region is needed when requesting. All services need region parameter if not specified).

### Register Service

1. Register as Facebook User: Request this service with User Information and Facebook information, this service will try to register with information provided. Thus return XML with register success or fail information.
2. Register as Normal User: Request this service with User Information, this service will try to register with information provided. Thus return XML with register success or fail information.

### News Service

1. Get all News: Request this service with “all” in o(peration) parameter. This service will find all news and return them in XML format.
2. Get all News without content: Request this service with “title” in o parameter. This service will find all news without content and return them in XML format.
3. Get all News ID: Request this service with “id” in o parameter. This service will find all news ID and return them in XML format.
4. Get News By News ID: Request this service with “byid” in o parameter and News ID. This service will find News by News ID provided and return News Information in XML.
5. Get News By Team ID: Request this service with “team” in o parameter and Team ID. This service will find News by Team ID provided and return News Information in XML.
6. Get News By Player ID: Request this service with “player” in o parameter and Player ID. This service will find News by Player ID provided and return News Information in XML.

### Post Comment Service

1. Post Comment To Specific News: Request this service with comment information, User information, and News ID. This service will add comment into database and return XML with message that success or not.

### View Comment Service

1. View Comment Under Specific News: Request this service with News ID. This service will retrieve all comments related with the news and return them in XML.

### Subscribe Service

1. Get Subscribed News for Specific User: Request this service with “get” in o parameter and User Information. This service will retrieve related information in database and return them in XML.
2. Get Subscribed News without Content for Specific User: Request this service with “gets” in o parameter and User Information. This service will retrieve related information in database and return them in XML.
3. Subscribe By Team ID for Specific User: Request this service with “subt” in o parameter, Team ID, and User Information. This service will trying to insert subscribe information into database and return process result in XML.
4. Subscribe By News ID for Specific User: Request this service with “subn” in o parameter, News ID, and User Information. This service will trying to insert subscribe information into database and return process result in XML.
5. Unsubscribe By Team ID for Specific User: Request this service with “un” in o parameter, Team ID, and User Information. This service will trying to delete subscribe information in database and return process result in XML.
6. Unsubscribe all for Specific User: Request this service with “unall” in o parameter and User information. This service will trying to delete all subscribe information for that user in database and return process result in XML.

### Scoreboard Service

1. Get All Scoreboards: Request this service with “all” in o parameter. This service will fetch all scoreboard information in database and return them in XML.
2. Get Scoreboards of a Team: Request this service with “team” in o parameter and Team ID. This service will fetch scoreboards of this team in database and return them in XML.
3. Get Scoreboard By Game ID: Request this service with “game” in o parameter and Game ID. This service will fetch scoreboard by ID and return them in XML.

### Team Service

1. Get All Team: Request this service with “all” in o parameter. This service will fetch all Team information in database and return them in XML.
2. Get Team By ID: Request this service with “byid” in o parameter and Team ID. This service will find Team ID and return team information in XML.
3. Get Team Member By Team ID: Request this service with “member” in o parameter and Team ID. This service will find all members in that Team by Team ID and return member information in XML.

## Load Balancer

1. Login Page: User could login with their Facebook Account or Normal Account.
2. Register Page: User could register a new account. Or register an account associate with Facebook redirect from login page.
3. Home Page: There is a list of news on home page.
4. News Page: Detailed News is showed in this page along with comments related to the news. User can also post comment from this page.
5. Scoreboard Page: User could see all scoreboard on this page.
6. Subscribe Page: User could see their subscription on this page.
7. Team Page: User could view all teams in this page. If a team is given, the team info and its members will show up.
8. Other Functionalities/Scripts

fb.php: Check if Facebook Account is registered.

loginprocess: Process login information and verify if login information is correct.

logout.php: Logout from this website and Facebook account.

pcomments.php: Process post comment.

register.php: Process registration information.

subprocess.php: Process subscription.

unsubbyid.php: Process unsubscribing by team id or news id.

# Encountered Problem and Solution

1. Difficulties in setting up Propel ORM for our Web Server. We at first design two same database instances. But in propel documentation, it mostly show how to connect to one database. By searching through Google, we find some solution for this problem. And we make two database schemas for two connections to solve this.
2. When we trying to use Memcached on DreamHost, they can’t install Memcached for our plan. In order to realize Memcached in our web application, we choose to host our Web Server locally.
3. When dealing with PHP HTTP request, we found that there is no functions easy to use for perform this task. We then search for PHP plugins for this task. HTTP\_Request2 become our choice and solve problem with http request.
4. When we trying to use AJAX access services on Web Server from Load Balancer, we found all AJAX are visible from Client. We think this is a vulnerable for our web application. After implementing login with Facebook, login with normal account, and register with AJAX, we found it is almost impossible for us to hide AJAX request to our Web Server. We then turn to use PHP for HTTP request.
5. When implementing Single Sign On, we can let Facebook User login, but we didn’t design database for store Facebook Users’ Information. Though Facebook can log in, but they can’t log into our web application. We then re-design our database and let first time Facebook User to register for our website.

# Reference

1. Memcached: <http://memcached.org/>
2. HTTP\_Request2: <http://pear.php.net/manual/en/package.http.http-request2.php>
3. W3School AJAX: <http://www.w3schools.com/ajax/>
4. PHP: <http://php.net/>
5. Bootstrap: <http://getbootstrap.com/2.3.2/getting-started.html>
6. Facebook Single Sign-On: <https://developers.facebook.com/docs/facebook-login/v2.2>
7. Janrain Developer | Sina Weibo Single Sign-On: <http://developers.janrain.com/how-to/provider-setup/sina-weibo/>
8. Node.js: <http://nodejs.org/>
9. ASP.NET: <http://www.asp.net/>
10. JSP: <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>
11. Propel ORM: <http://propelorm.org/>
12. Tomcat: <http://tomcat.apache.org/>
13. DreamHost: <https://www.dreamhost.com/>
14. Apache HTTP Server: <http://httpd.apache.org/>
15. Composer: <https://getcomposer.org/>
16. UTDallas CS Oracle: <http://cs.utdallas.edu/about/Oracle%20help.htm>
17. MySQL: <http://www.mysql.com/products/>