



THE UNIVERSITY OF TEXAS AT DALLAS

Final Report
Ecommerce Website
Web Coders



Prepared By:

Vineet Dhanawat (vxd130730)

Nevil Ladani (npl130030)

Rinu Budhbhatti (rxb120630)

Guided By:

Dr. Mithun Balakrishna

Course: CS 6301 Advanced Web Software Project

Spring 2014

I. ACKNOWLEDGMENT:

We would like to express the deepest appreciation to Dr. Mithun Balakrishna, who has shown the attitude and the substance of a great mentor: he continually and persuasively conveyed a spirit of adventure in regard to our project, and an excitement in regard to teaching. Without his supervision and constant help this project would not have been possible.

TABLE OF CONTENTS

I.	ACKNOWLEDGMENT:.....	2
1.	INTRODUCTION:	4
2.	TOOLS AND TECHNOLOGIES:.....	5
3.	ARCHITECTURE	6
4.	ARCHITECTURE DISCRIPTION	7
5.	FLOWCHART.....	8
6.	FEATURES	9
7.	WEB SERVICES	11
8.	LEARNING	13
9.	CHALLENGES	13

1. INTRODUCTION:

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

The objective of this project is to develop a general purpose e-commerce store where any product (such as computers, laptops, mobile, electronic devices) can be bought from the comfort of home through the Internet. However, for implementation purposes, this project will deal with an online store for computers and laptops.

An online store is a virtual store on the Internet where customers can browse the catalog by brand or product name and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer information will be retrieved based on the user session using one of the web services.

2. TOOLS AND TECHNOLOGIES:

A service-oriented architecture (SOA) is the underlying structure supporting communications between services. SOA system in ecommerce is essentially a business process which comprises of various services, loosely coupled.

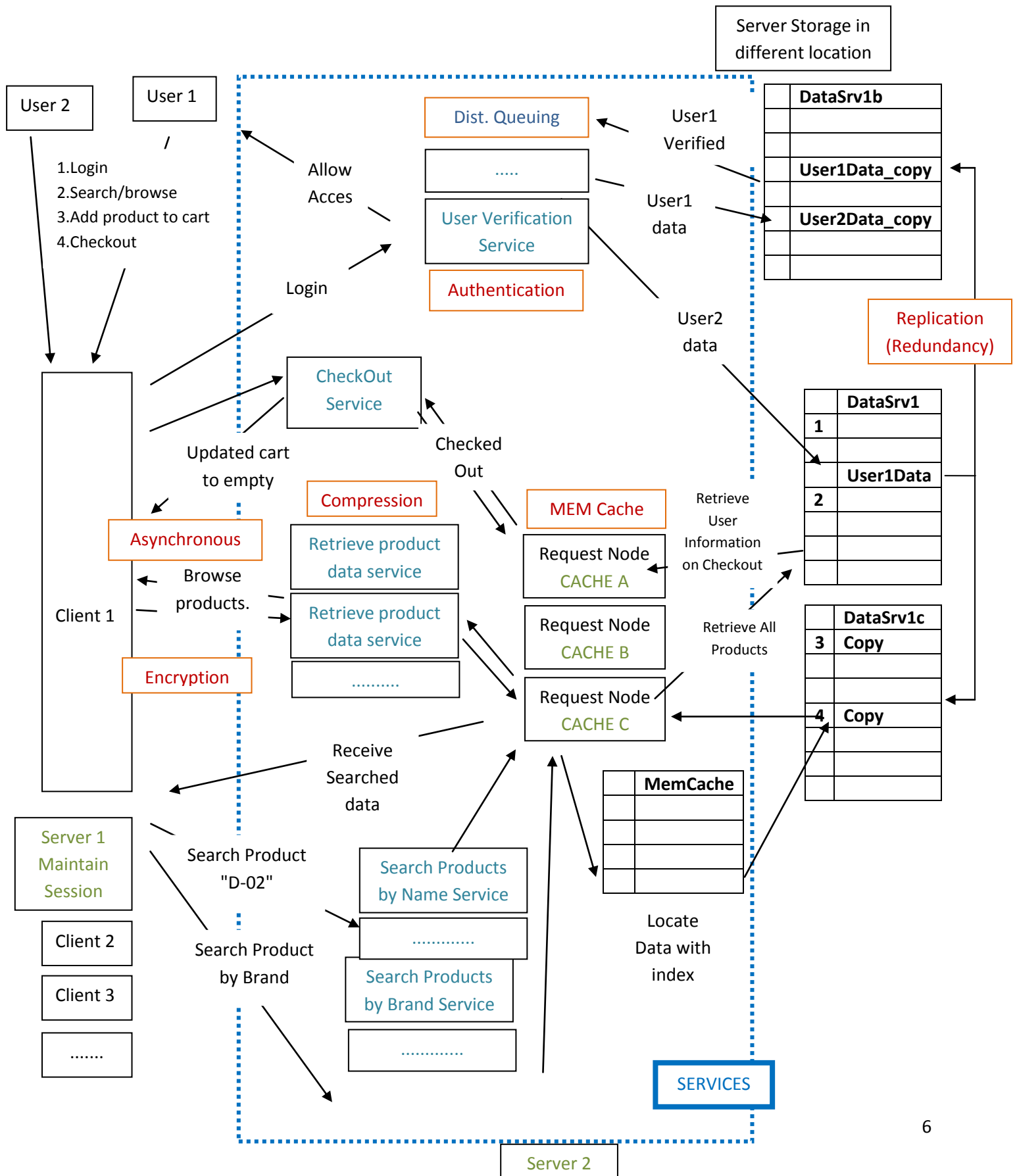
Simple Object Access Protocol (**SOAP**) based **Web services** are becoming the most common implementation of SOA. However, there are non-Web services implementations of SOA that provide similar benefits. The advantage of SOA is due to the protocol independence, it means that different consumers can communicate with the service in different ways. Ideally, there should be a management layer between the providers and consumers to ensure complete flexibility regarding implementation protocols. Web Services that uses SOAP over HTTP protocol for the communication allows user to use existing low cost internet for implementing Web Services. This solution is much less costly compared to proprietary solutions like EDI/B2B. Beside SOAP over HTTP, Web Services can also be implemented on other reliable transport mechanisms like FTP etc

One of the many reasons why we chose JAVA was because it's easily deployed in heterogeneous environment ranging from Linux to Windows. It has many frameworks that ease the development and they are reliable because they are mature enough.

Tools and Languages:

1. **GitHub** as the Configuration tool and **EGit 3.0.3** eclipse plugin.
2. IDE used **Eclipse - Kepler**.
3. Developed in **JAVA/J2EE 7.0**.
4. Deployed and tested on **Apache Tomcat 7.0** on Local Machine.
5. Oracle JAVA coding standards/conventions enforced by **eclipse- JIndent plug-in**.
6. Database Technology used: MySQL 6.0

3. ARCHITECTURE



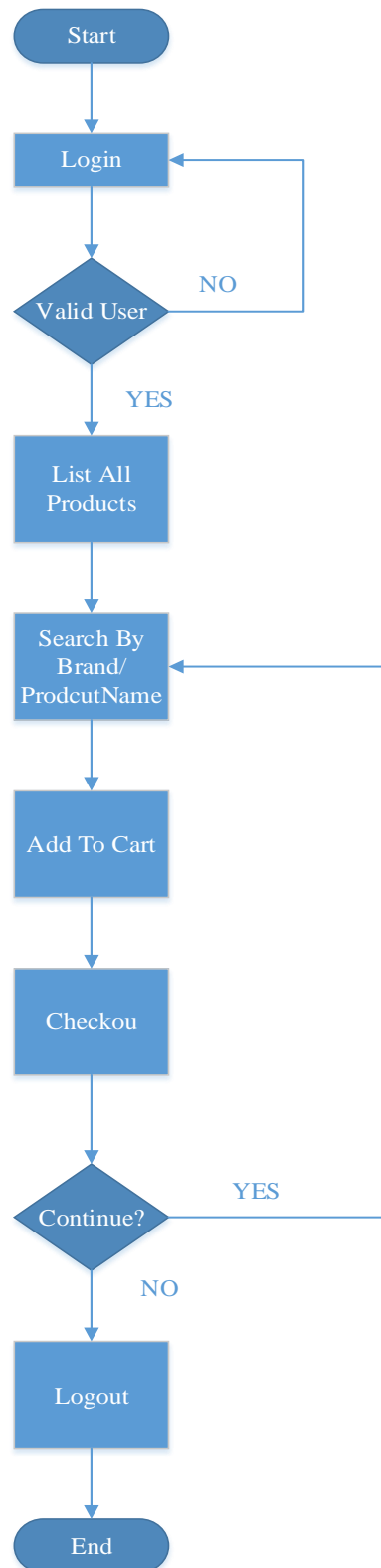
4. ARCHITECTURE DISCRPTION

- The above diagram represents the entire architecture of our web service project.
- The Labels in **Blue** represent **Services** and the Labels in **Red** represent **Core Features**

The system is divided into 2 separate layers which are implemented on two completely different server instances.

- A client side which handles the interaction with the user. We have used jsp and servlet to implement it. The client side maintains the session for the user in the name of the user name. It handles the following functionalities:
 - Login interface: Takes in username password and sends it to validation service to process it for authentication. Once the user is verified, he/she is allowed to access the site.
 - Browse products interface: It displays all the products that are available using the Retrieve product data Service.
 - Search Product by brand: On clicking a particular brand it retries the products from database using Search products by brand Service.
 - Search Products by name: On entering a name or part of name of a product it retrieves the products accordingly using Search products by name.
 - Cart: The user can add, remove, change the quantity of a product in the cart.
 - Check Out: On checking out it retrieves the user information from the database using the Checkout Service.
- Service side which provides the necessary services(5) to the client side
 - User Verification Service: Check the database to confirm if the user and password combination exists.
 - List all products Service: Sends the data of all products to the interface layer.
 - Search products by brand Service.
 - Search products by name Service: Retrieves all products with name or part of name and sends it to client
 - Check out service: On check out the system retrieves the user data using the session name sent from the client side and sends the complete information back to the client.

5. FLOWCHART



6. FEATURES

1. High Availability Application server level:

We have used HA proxy on Ubuntu to set up load balancing. Proxy server makes sure that if one of the server is down the request is sent to another working server. This way the web site is never down and there is high availability of website.

2. High Availability Database Level:

We implemented MySQL replication with ubuntu. This works with two or more MySQL servers. One of which acts as masters (for writing) others would act as slave (for reading). It's extremely important that you never write to a slave database, this is a sure way to screw up your replication and give you some heavy headaches. Always write to the master and read from the slaves.

3. High Performance:

We installed Memcached for Ubuntu. Memcached by default runs on **port 11211** (make sure that is open if you have firewall rules). Config file is at **/etc/memcached.conf**. To implement the code we used a framework named spymemcached. We imported the jar file to implement all the required APIs call related to extracting and storing the object in cache.

Efficient storage of objects. General serializable objects are stored in their serialized form and optionally compressed if they meet criteria. Certain native objects are stored as tightly as possible (for example, a Date object generally consumes six bytes, and a Long can be anywhere from zero to eight bytes).

4. Sharding:

We were able to implement sharding on four nodes.

- Management node
- 2 database shard
- Load balancing node.

But sharding with replication over MySQL could not be achieved, hence for the purpose of demo sharding has not been included.

5. Client Server Encrypted Communication (SSL):

Encryption can be achieved by generating signed certificate and keystore using JDK. The application server must also support the encryption certificate. In the Tomcat we have to add the following code to server.xml

```
<Connector SSLEnabled="true" acceptCount="100" clientAuth="false"
disableUploadTimeout="true" enableLookups="false" keystoreFile="C:\Program Files\apache-
tomcat-7.0.53\conf\ecommerce.keystore" keystorePass="Welcome123" maxThreads="25"
port="8443">
```

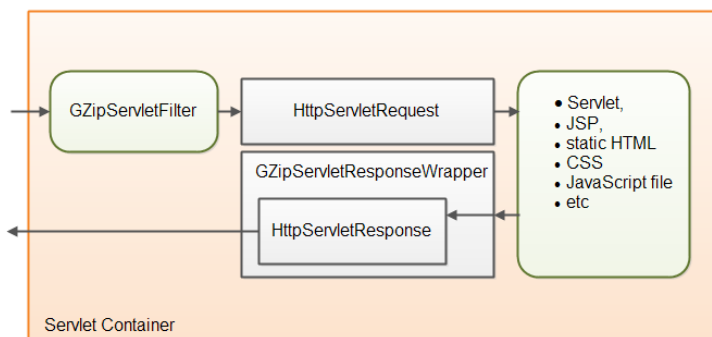
We generated ecommerce.csr which is a certificate file and ecommerce.keystore which is a keystore file.

6. Request response compression:

Server side compression: Tomcat 7 implements the server side compression on its own but for the older version of Tomcat Servers we have to add the following code to the server.xml

```
compression="on"
compressionMinSize="2048"
noCompressionUserAgents="gozilla, traviata"
compressableMimeType="text/html,text/xml"
```

Client side compression: We implemented gzip servlet filter. The request goes through httpServlet, GZipServletResponseWrapper and HttpServletResponse for the compression.



7. Authentication/ Authorization:

We authenticated the user using one of the services. The client would send the username password information to the service. The service would verify whether the combination exists. If successful the user is allowed to access the site. We used javascript to make sure that the user can't access any of the web pages if he is not logged in.

8. Asynchronous Service:

We implemented asynchronous service through AJAX. The call is made to the service, and the page then continues. When the response comes in, required functions are executed, asynchronously. The client doesn't have to wait for the response from Service and can continue with its own work during the response delay.

7. WEB SERVICES

A Web service is a unit of managed code that can be remotely invoked using HTTP, that is, it can be activated using HTTP requests. So, Web Services allows you to expose the functionality of your existing code over the network. Once it is exposed on the network, other application can use the functionality of your program. The software system that requests data is called a *service requester*, whereas the software system that would process the request and provide the data is called a *service provider*.

Five areas of functionality were identified as offering potential value in using Web Services to deliver an efficient ecommerce website. These are:

User Verification

The system uses the verification web service to validate the user by checking the existence of user in the user database. Upon validating the user, the system will maintain a session for each user. Each user will enter unique username to register. And session will be created using username. The service requester will call the service using username and password as the parameter. The service provider will return true if user is validated. And the client maintains a session for each user.

List All Products

Product database contains the list of all the products to be sold online. The system uses List All Products to retain the complete list of products and display it on the website. The client will request for all list product. The service will return an array of all products which is converted to hash map on the client side.

Search by Brand

Users will be able to search and browse for the products across all brands. Having a search box positioned well, will allow your visitors to quickly search for the all the products for specific brand that they are looking for. The client will request the service provider and will pass brand name as the parameter. The service retains data for the specific brand from the product database. The service will return the complete array to the client which will be then converted to the hash map.

Search by Name

Users will be able to search and browse for the products by product name. Having a search box positioned well, will allow your visitors to quickly search for the product that they are looking for. The client passes the product name as a parameter while making a request to the service provider. The service retains data for the specific product from the product database. The service will return the complete array to the client which will be then converted to the hash map.

Checkout

Users will be able to add and remove the desired products multiple times from the shopping cart. Navigation is extremely important on an e-commerce site, users should be able to easily access various sections of the site, along with a quick link back to the shopping cart. The client will pass the session id to the service provider and the service provider will retain all user information like address and the stored payment card information.

Tools and Technology used:

- Webservice: JAVA
- Client : JSP, Servlet
- Database: MySQL and Memcached for cache

8. LEARNING

Web Service

While working on this project, we mainly learned what is web services and how do we implement them for our project. While implementing we came across many challenges like compatible data types while passing the arguments from service provider to service requester. One of which was, the SOAP service was not able to pass data into hash maps which compelled us to convert the data in array, pass it and then again convert it back to hash map.

Configuration

Configuring the system and setting up different tools and technologies such as Tomcat Server, MySQL Database and many more required a lot of exploration and there was a lot to learn about how these technologies interacted with each other.

SOAP

There was a lot to learn in the field of Simple Object Access Protocol. It involved making the services loosely coupled with client as well as with each other. We learned the entire process of how the wsdl file(xml based) generated from Service Endpoint Interface can be used to generate Stubs on the client side. The Stub thereafter can be used to access the service.

9. CHALLENGES

With the numerous advantages of Web services come a few challenges. Most significantly, though Web services themselves are designed to be simple, actually developing and implementing them can be complex. WSDL syntax becomes complicated quickly, especially when building a service with multiple operations in a text-based editor. Even looking at the completed code, it's difficult to follow the chain of connections from a service name, to the binding, to the port type, and so on, never mind writing the code correctly by hand. Best practices dictate that designing the WSDL be the first step in architecting Web services. So we used the contract-first approach. Designing the interface first results in better overall planning prior to implementing the service and helps ensure the service will be effective in multiple client scenarios.

The other challenge we faced was version control. In order to co-ordinate between the team, we needed to implement version control system. A version control is a repository of files, often the files for the source code of computer programs, with monitored access. Every change made to the source is tracked, along with who made the change, why they made it, and references to problems fixed, or enhancements introduced, by the change. So we decided to use Git revision control system.