

# CS 6314.501 Web Programming Languages (Fall 16)

## Project Report

Group name:

### Ansage

(un-zaa-gh; *German; bid, announcement*)

Members:

Umang Shah  
uks160030@utdallas.edu

Sreenivas Venkitachalam  
sxv163530@utdallas.edu

Computer Science Dept,  
The University of Texas at Dallas

## 1. Architecture

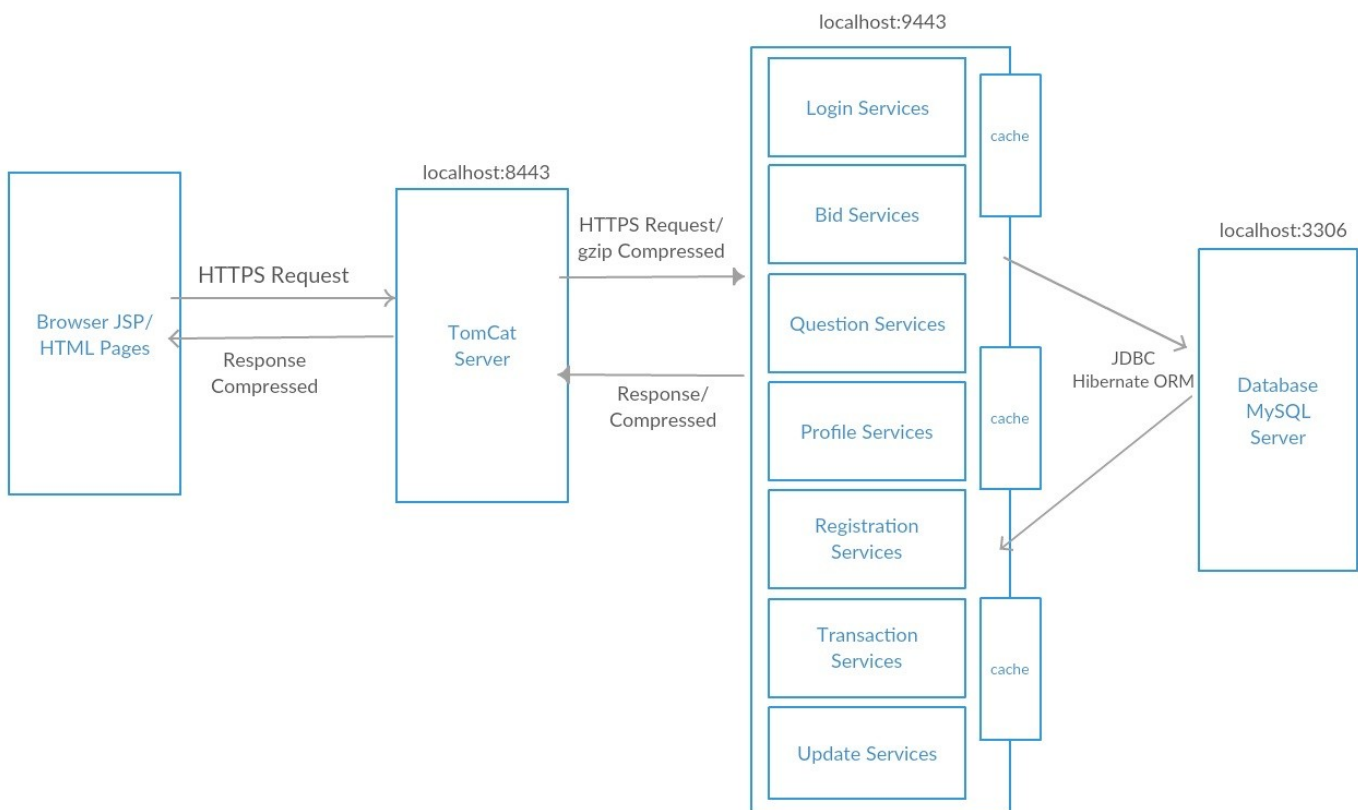


Figure 1. Architecture of Ansage

1. Browser JSP/HTML Pages: Serves the clients with an interface. Dynamic responsive web pages take user request and send the request to the server.
2. Application Server at localhost 8443 collect the request from Client side and make a web Service call to server running on port 9443. It will be HTTPS Request and the payload will be gzip compressed.
3. Server running on localhost 9443 hosts RESTful Web Services which receive the request and process the request by fetching data from Mysql database Server.
4. Mysql server is running on localhost:3306 which is accessed by web services.
5. Server on localhost 9443 will send the response back to application Server which will be in compressed format.
6. Application Server sends the response back to JSP Pages which has presentation logic to display the information.

## 2. Modules and Technology Used

### 2.1 Front End

- Considered responsive templates like Bootstrap and Foundation. **Bootstrap** was used as it provides more features and it provides better environment for customization.
- For Web pages several technologies were considered like ASP.NET with C#, PHP etc. **JSP** was used along with client side scripting languages like **JAVASCRIPT**, **JQUERY** and **HTML5**, **CSS** for generating dynamic Web pages.
- **JAVA SERVLETS** are used to interact with Clients via request-response model based on **HTTPS** and Client requests are sent to Web Services using JAVA Servlets by Web Services calls.

### 2.2 Web Services

- Web Services technologies like SOAP, RESTFUL were considered. **RESTFUL** was used because of its less reliance on Tools and agnostic to language and platform.
- RESTFUL-Apache CXF, **JAX-RS** were considered. **JAX-RS JERSEY** was used as it provides support for data binding and advanced API for content negotiation,

### 2.3 Database

- **Hibernate ORM** Framework was used for Object relational mapping to database. Hibernate was selected as it have extensible features to implement various complex functionalities and its framework supports popular relational databases.
- Relational databases like MYSQL, Microsoft SQL SERVER 2012 were considered. **MYSQL** was used as it has flexible features to work with Hibernate ORM framework.

## 2.4 Cache

- Memcached, EHCache were considered. Out of which **EHCache** was select to implement the distributed cache due to its compatibility with Hibernate ORM.

## 3. Functionalities

New User Registration	<ul style="list-style-type: none"> <li>• Registration form is present where user details like name, email, password, bio data, skills and tagline are collected. Implemented various Client side checking like valid email and password confirmation.</li> </ul>
Existing User login and logout	<ul style="list-style-type: none"> <li>• Existing user is able to login and log out from application.</li> </ul>
User profile information display and editing	<ul style="list-style-type: none"> <li>• A view profile page is there where user can view basic profile details like name, coins, email, tagline, skills, biodata. When User clicks Edit button he can change the profile details and updated values are saved and displayed to user.</li> </ul>
User Login information	<ul style="list-style-type: none"> <li>• When user login basic information of last login is shown which includes date and time of last login.</li> </ul>
Ability to post questions	<ul style="list-style-type: none"> <li>• A user can post question where user can post a question. Along with posting question user can specify question characteristics like category, question description</li> </ul>
Ability to bid questions	<ul style="list-style-type: none"> <li>• A valid user can bid (Solution for question) questions posted by other users where he can specify the bid offer for the answer.</li> </ul>
Page listing all the bids for given question	<ul style="list-style-type: none"> <li>• A user who posted the question can view all the bids posted for the question. Bids are displayed in a</li> </ul>

	sortable table which can be filtered based on bid offer, number and list of skills of bidder, bidder coins.
Search for questions by user which he would like to bid	<ul style="list-style-type: none"> <li>• A User can search for questions that he would like to bid. Search will return all questions which is matching the search criteria given by user.</li> </ul>
Table display	<ul style="list-style-type: none"> <li>• Results are displayed in a sortable table.</li> <li>• Search Results filtering capabilities are enabled on result items.</li> </ul>
Shopping Cart and Order purchase confirmation	<ul style="list-style-type: none"> <li>• When a user wants to add a bid to his/her shopping cart, ADD button is given which will enter the corresponding bid to shopping cart.</li> <li>• When a user wants to removes a bid from Shopping cart, REMOVE button is given which will remove the corresponding bid from shopping cart.</li> <li>• When a user wants to get more than one item (Bid offered by particular user for answering question) he/she can update the count of item.</li> <li>• When order is submitted order is confirmed and placed and a purchase order received confirmation email will be sent to purchaser and bidder.</li> </ul>
Unavailable page retrieve generic 404 page	<ul style="list-style-type: none"> <li>• When a page is not available a pretty and generic 404 page is displayed.</li> </ul>

## 4. Web Services

Registration Services	<ul style="list-style-type: none"> <li>• Receive the user details - name, email, tagline, skills, password, bio from Servlet.</li> <li>• Details are saved in database. User is asked to Login.</li> </ul>
Login Services	<ul style="list-style-type: none"> <li>• Take the user credentials (email, password) and validate against credentials in the database. If User is valid his email, name, coins, id along with details of last date, time and location of valid login is sent back to servlet.</li> </ul>
Question Services	<ul style="list-style-type: none"> <li>• Details of question posted by user are collected. These details are saved into database. Question id is sent back to</li> </ul>

	<p>servlet.</p> <ul style="list-style-type: none"> <li>• Accepts search query, filters it and returns questions that are best matched. Results include question, description, category, owner id, owner name , current number of bids for questions, owner coins details.</li> <li>• When a user tries to retrieve a question by giving question id, question is returned along with question id, description, name and profile id of owner who asked the question is returned back to servlet.</li> <li>• Also provides service to display all questions.</li> </ul>
Bid Services	<ul style="list-style-type: none"> <li>• When a user places a bid to answer a question - question id, bidder id, bid offer is given to this service and details of bid are saved to database.</li> <li>• When a user wants to retrieve all bids for a question, question id is given to service and it returns all bids for particular question which include id, bid offer, name, coins, skills , bid id of bidder</li> </ul>
Profile Services	<ul style="list-style-type: none"> <li>• A User profile is retrieved which includes user name, tagline, bio data, skills and profile id of user is returned. Profile id is given as parameter to retrieve the record.</li> <li>• When a user updates profile parameters updated are fetched and updated details are sent back to servlet.</li> </ul>
Transaction Services	<ul style="list-style-type: none"> <li>• When a user selects a bid this Service is called with bid id as input parameter. Details of bid will be added to cart successfully.</li> <li>• When a user removes a bid from Cart this service is called with bid id as input parameter. Corresponding bid will be removed from Cart.</li> <li>• When User wants to view his cart this Service is called with User id as input parameter. Cart with Bids added by user will be returned back to servlet.</li> </ul>
Update Service	<ul style="list-style-type: none"> <li>• It is a service called periodically to retrieve coins of user when User id is passed to Web Service. It will fetch the coins and sent back to servlet.</li> </ul>
Answer Services	<ul style="list-style-type: none"> <li>• When a user is given permission to answer, he is allowed to answer that particular question. Question id, Request id</li> </ul>

	<p>are passed as parameters to Web Service.</p> <ul style="list-style-type: none"> <li>• When a User answers to a question answer is saved for the particular question. Parameters given to Web Services are question id and Request id.</li> </ul>
--	---

## 5. Problems Encountered

- Implementing Web Services was a major challenge.  
Solution: Learned JAX RS and used various internet materials and lecture slides to implement Web Services using RESTFUL.
- Transferring of Data was a major issue. Transferring a list from Web Service to Servlet and transferring a list of data to Web Services was causing issue.  
Solution: Used various materials to learn the different ways of transferring data. Problem was resolved by transferring JSON Objects and JSON Arrays.
- Automatic updates were another issue.  
Solution: So learned the concepts of AJAX implementation to resolve the issues.
- Implementation of Hibernate ORM Framework was another challenge. Initially connection between Hibernate and database was issue.  
Solution: Used several online materials to resolve this issue.
- Implementing Cache was another issue. There was various version related problems for ehcache. Initially cache was not hitting and data was not updated in Cache.  
Solution: Resolved by referring various Hibernate Tutorials and online materials.

## 6. Conclusion

In the project we implemented a Service Oriented Architecture and created a scalable distributed system. We worked on various technologies for the same. And this project provided us with a great learning experience.