FINAL PROJECT -
Web Programming Languages

Group Name- D.A.G

Group Members

Devendra Lad (dvl160030)

Amruta Folane (asf160130)

Gayatri Prabhu (gdp160130)

Overview

The project demonstrates an auction/ bidding website for furniture products. The website resides in various divided parts like the client side using JSP, web server and micro server.

WEB ARCHITECTURE:

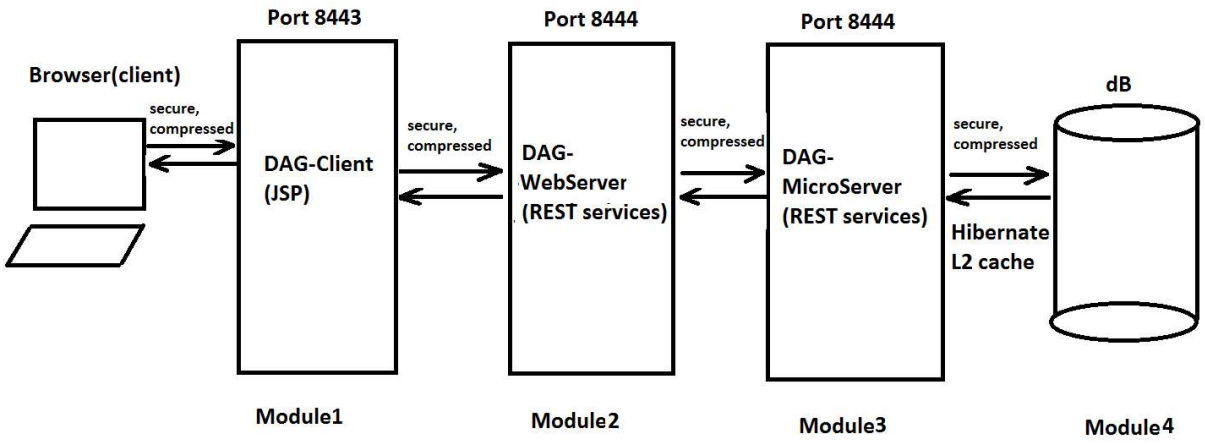

Fig. WEB ARCHITECTURE

Languages: SQL, Java

Technologies:

Module	Considered	Used	Reason
Module 1	JSP, ASP	JSP	familiarity of members with JSP
Module1, Module2	Jersey Client, Apache HTTP Client, Apache CXF	Jersey client	well managed & documented resources, more stable
Module2, Module3	Jersey REST, Spring MVC	Jersey REST	Familiarity with Jersey, Spring is not really JAX-RS

Module4	ORM Hibernate, Active JDBC, Top Link	ORM Hibernate	Popular, Familiarity of member, better support for JSON
Module4	Memcached, Hibernate L1, L2 caching	Hibernate L2 caching	Memcached installation issues for Mac, easy to configure cache with existing hibernate implementation

Implemented functionalities

Front end Services (client side)

The Client displays the web pages on the users' browsers. It uses JSP to display the web pages and Controller Servlets to talk to Web Services.

Sr No.	Functionality	How to Access?	Controller Servlet Used	Web service called
1.	New user registration	New user link on login page	Registration Controller Servlet	register
2.	Existing user login	Login page (index page)	Session Controller Servlet	login
3.	Existing user logout	Navigation tab once logged in	Logout Controller Servlet	logout
4.	User Profile Display	Navigation tab once logged in	Account Controller Servlet	getMyProfile
5.	User Profile Edit	Edit button under User Profile Display	Editor Controller Servlet	updateProfile
6.	Search for items to bid	Search box on welcome page, click on the item of interest in the search list.	Bitems Controller Servlet	searchForProducts

7.	Displays all the bids of the user with their statuses	Navigation tab once logged in	Bid Controller Servlet	getMyBids
8.	Delete your bids	Delete button next to that item under user's bids display	Bdel Controller	deleteBid
8.	Displays all the posts of the user	Navigation tab once logged in	Post Controller Servlet	getMyProducts
9.	Post an item	Post Item button under user's posts display	Pitems Controller Servlet	postItemForSale
10.	Delete the posted item	Delete button next to that item under user's posts display	Pdel Controller Servlet	deleteProduct
11.	Display the bids of the posted item	Click on the posted item of interest, under user's posts display	Pbids Controller Servlet	getProductBids

Web Services

Web Server is the communicator between the client and the micro server.

The following are the web services implemented:

- **CustomerController:**

Sr.No.	Web Service	Input Params	Service returns	Microservices called
1.	**register**	name, password, contactNo, address, emailId	a true string if done successfully*	3 - 'isEmailIdAvailable', 'insertUserData', 'sendEmail'
2.	**logIn**	customerId	a true string if done successfully	2- 'checkLoginCred', 'getUserId'

3.	**logOut**	customerId	a true string if done successfully	1- 'logOut'
4.	**getUserProfile**	customerId, authKey	the user's profile as a JSON object	2- 'isLoggedIn', 'getUserProfile'
5.	**updateProfile**	authKey, customerId, name, password, contactNo, address, emailId	Updates the user's profile	2- 'isLoggedIn, 'updateProfile'
6.	**searchForProduct**	authKey, customerId, keyWord	Searches the products(case-insensitive)	2- 'isLoggedIn, 'searchForProduct'

1. ***register**

This web service is implemented for registering a new user if user does not exist already. The micro service 'isEmailIdAvailable' checks if the email used for registration of the new user is not already in the database. If that returns affirmative, 'insertUserData' is then called to insert all the new user data and make a profile for this user. After this is done, a verification email is immediately sent welcoming the new user using 'sendEmail'.

❑ **ProductController:**

Sr.No.	Web Service	Input Params	Service returns	Microservices called
1.	**postItemForSale**	customerId, name, description, startBid, sellDate	True if item posted successfully	2 - 'isLoggedIn', 'createProduct'

2.	**getMyProducts**	customerId, authKey	jsonObject of all products of the user	2- 'isLoggedIn', 'getMyProducts'
3.	**deleteProduct**	productId, customerId, authKey	a true string if done successfully	1- 'isLoggedIn', 'logOut'

❑ **BidController:**

Sr.No.	Web Service	Input Params	Service returns	Microservices called
1.	**bidForItem**	customerId, Authkey, productId, amount	True if bid posted successfully, msg if any of these: -User not found -Product not found - Sell date passed -Bid is less than startBid	2 - 'isLoggedIn', 'bidForItem'
2.	**getProductBids**	customerId, authKey, productId	jsonObject of all bids of the user's product	2- 'isLoggedIn', 'getProductBids'
3.	**getMyBids**	customerId, authKey	jsonObject of all bids of the user	2- 'isLoggedIn', 'getMyBids'
4.	**deleteProduct**	bidId, customerId, authKey	a true string if done successfully	1- 'isLoggedIn', 'deleteBid'

Micro Services (back end)

Micro Server is the server that transacts with the database.

The micro services are implemented using the **ORM framework** and hibernate does all the auto-population of the database and queries directly. Queries done using **hibernate query language**.

The following are the micro services implemented:

❑ **CustomerController:**

Sr. No	Micro Service	Input params	Service returns
1.	**isLoggedIn**	customerId, authKey (a key generated internally just to keep for a particular session, is made '0' when user logs out)	true (yes, is logged in) or false (is not logged in)
2.	**getUserId**	emailId	customerId
3.	**isEmailIdAvailable**	emailId	true (if absent in dB), false (if present in dB)
4.	**insertUserData**	name, password, contactNo, address, emailId	returns nothing
5.	**sendEmail**	type, emailId	returns nothing**
6.	**checkLoginCred**	emailId, password	authKey***
7.	**logOut**	customerId	true (after logging out successfully)
8.	**getUserProfile**	customerId	the entire customer object with all its details contained.

9.	**updateProfile**	name, password, contactNo, address, emailId	True if successfully updated

1. ****sendEmail**

[type 1 <= registration email, 2<= send item sold notification]

Service does not return anything. Sends the appropriate email to the user depending upon the type of message.

2. *****checkLoginCred**

This service used to validate the user credentials when logging in. It generates a 4- digit authorization key - 'authKey'. The value of authKey is a random 4-digit number if credentials are correct, else it is '0'.

❑ **ProductController:**

Sr. No	Micro Service	Input params	Service returns
1.	**createProduct**	customerId, name, description, startBid, sellDate	String msg if user not found, else true
2.	**getMyProducts**	customerId	List<Product> - arraylist of all products of the user
3.	**searchForProducts**	keyWord	List of all products containing the keyWord
4.	**deleteProduct**	productId	String msg if product does not exist, else true

- **BidController:**

Sr. No	Micro Service	Input params	Service returns
1.	**bidForItem**	customerId, Authkey, productId, amount	True if bid posted successfully, msg if any of these: -User not found -Product not found - Sell date passed -Bid is less than startBid
2.	**getProductBids**	productId	jsonObject of all bids of the user's product
3.	**getMyBids**	customerId	jsonObject of all bids of the user
4.	**deleteProduct**	bidId	a true string if done successfully, else msg if bid does not exist

Problems faced & resolved Solutions:

- Session persistence issue on Chrome

  Solution: Switched to Firefox.
- HTTPs configuration between two Tomcat servers on Mac

  Solution: Specific instructions were not available for Mac, so found and resolved it.
- Memcached installation problems for Mac

  Solution: Used Hibernate cache