FREE AND FOR SALE WEBSITE

Group Name: Synergy

APRIL 29, 2016

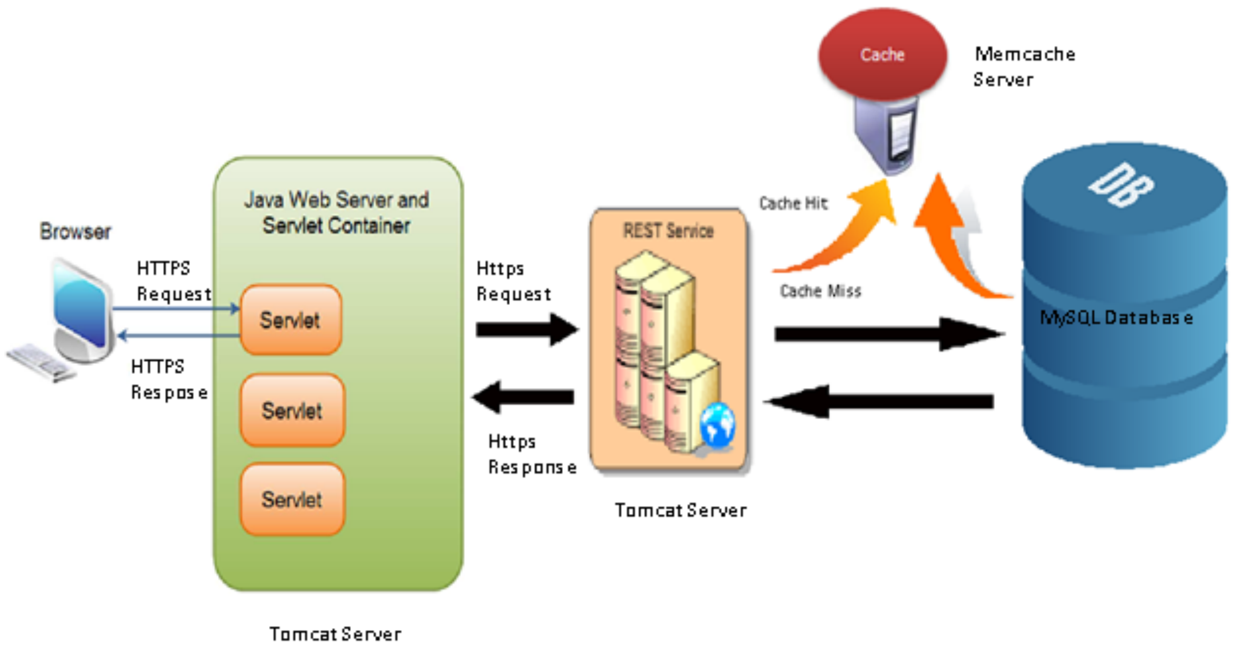
GROUP MEMBERS
Utsav Dholakia (uxd150130), Sagar Mehta (sam150930),
Siddharth Shah (sas151830)

Free And For Sale Website

Architecture:

The Free and For Sale Website is scalable web application based on the service-oriented architecture (SOA).

The architecture consists of four main components:



Architecture Diagram

1) Java Web Server and/or Servlet Container:
    - The web server hosts website and make calls to the Rest Service to obtain data.
    - The interaction between the web server and Rest Service; and browser and web server is secured using https protocol.
    - The web server is developed using JSP/Servlets and deployed on Tomcat Server.
    - The interaction between webserver and Rest Service Server is secured using secret key.

2) Rest Service:
    - The Rest Service is the one which interacts with the database and provides data to the web server.
    - The Rest Service first checks in the cache and then if there is cache miss interacts with database for getting the data.
    - The Rest Service are developed using Jersey Framework and deployed on Tomcat server.

3) Cache:
    - This stores the frequently used data and makes data available faster to the rest service preventing expensive database calls.
    - Whenever there is cache miss, data will be populated from database.
    - Whenever there is cache hit, data will be send from cache.

4) Database:
    - The database will be called in case of cache miss.
    - Whenever there is cache miss, it will first save data in cache and then return data to the rest Service.
    - We have used Hibernate ORM for connecting with MySQL database.

Technology Used:

Template: Bootstrap (Source URl: https://w3layouts.com/classic-style-ecommerce-flat-bootstrap-responsive-web-template/)

Webserver: Apache Tomcat v 8.0.33
Other Options: Apache Http Server.
Tomcat was preferred because of the following reasons:

- Provides the Java Servlet and JSP support for dynamically served pages
- Works as a light-weight testing server
- Can be run in different modes to promote better performance

Rest Service Framework: JAX-RS Jersey Rest framework.
Other Options: Spring MVC
JAX-RS was considered because

- JAX-RS has the advantage of creating APIs that are simpler to create and digest messages for in different browsers and mobile devices.
- The Rest Services are deployed on Apache Tomcat server.
- JAX-RS targets the development of web services (as opposed to HTML web applications) while Spring MVC has its roots in web application development.

Cache: MemCache Distributed Cache.
Other Options: Redis.

- Memcached is a good option for implementing a distributed caching mechanism.
- It stores data in the format of key value pair. Currently, only single node is implemented.

.

Database: MySQL v5.7.11
Other Options: Orcale, PostgreSQL.

- MySQL is relatively light-weight, can be extremely fast when applications leverage architecture.
- Lots of features stay free as the database servers grow such as replication and partitioning.
- MySQL excels when high speed reads can be used for web, gaming and small/medium data warehouses and OLTP systems.

Development Environment: JDK 1.8(JSP/Servlet Framework)
Other Options: Javascript, Python

- Java is high performance compiled languages, with great support from a well known vendors, and entire ecosystem of companies that sell everything - from IDEs to libraries to automation tools.

ORM: Hibernate 4.3.6
Other Options: JPA, IBatis.

- Hibernate generates SQL for you which means you don't have to spend time on generating SQL. Hibernate is highly scalable.
- It provides a much more advanced cache.

Compression: GZip compression.

Functionality:

Website Overview:

- The website is for the users to sell their used products.
- The user can search and purchase the products by adding it to the cart and write comments for the sellers.

Inventory:

- Whenever the user logs on to website, the user will be shown different products to choose from.
- The user can search for particular item using itemName.

Register/Login:

- The user has to first register on the website for making any purchase.
- After user is registered, the user is redirected to home page where user can purchase products.

Product Detail Page:

- The user can click on any item to check for more product details.

Cart:

- The user can add product to cart.
- The user can update, remove product from the cart. Once the user purchases the product, confirmation email is sent to user.

History:

- The user can view past transactions.

Review:

- The user can view / give review.

Edit Profile:

- The user can view and edit his own profile.

Other

- The user will receive email when user opts to leave message on contact Us page.

Web Services:

Module	*Service*	*Descripion*	*Input*	*Output*
Invent o-ry	/InventoryServices/InventoryForHomePage	This service fetches the list of different products.	No input	ProductID, Product Name, Cost, Image for all products
	/InventoryServices/SearchInventory	The service returns the search results	ItemName	ProductID, Product

				Name, Cost, Image for all products
	/InventoryServices/FetchMoreDetails	The service returns details of individual product.	ProductID	ProductID, Product Name,Price, Description, All Images
User	/loginservices/checkuservalidity	The service validates the user.	username,passwrd	username,userId
	/registrationservices/newregistration	The service registers the user.	USerName, password, Name, Address	username,userId
	/profileservices/getprofile	The service gets Profile information	userID	USerName, Name, Address
	/profileservices/updateprofile	The service updates the profile information.	USerName, Name, Address	Success Message
Cart	/CartService/addToCart	The service adds the product to user cart.	ProductID	Success Message
	/CartService/getCart	The service gets the cart details.	UserID	Products in the cart
	/CartService/saveUpdateCart	The service saves the carts.	Products in the cart	Success Message
	/CartService/purchaseCart	The service purchase the products from the cart and sends the confirmation email	Products in the cart	Success Message
History	/TransactionService/getTransactionHistory	The service gets the transaction history of the user.	userId	Past transaction data like items purchase, cost, date of purchase.

Review	/ReviewServices/getReviews	The service gets the past review given by the user	userId	Reviews
	/ReviewServices/saveReviews	The service saves the review.	Reviews	Success
ContactUS	/ContatcUsService/SendEmail	The service sends the email for the message received confirmation.	emailId	Success Message

Problems Faced:

1) Send Email: The email server configuration was not working correctly. The solution was to configure Gmail smtp server for sending email.
2) Servlet Redirection: The redirection from servlet to JSP was giving null pointer exception because of data on JSP being null. The solution was to set beans correctly in request object.
3) Hibernate Configuration: The hibernate was working very slow. As the hibernate session factory and hibernate session was being created again and again. The solution was to create static hibernate session factory.
4) Add to Cart: We had form with two submits one for adding to cart and other for viewing the detail, so submit was not working as expected. The solution was to use attribute onclick to call form.action method.
5) Hibernate Eager Fetch Not working: The problem was there was no data in joined entity that was populated by Eager Fetch. The solution was to make another Database call.
6) Jquery Post call: The problem was Jquery Post call was not working with JSP. The solution was to use form to submit data.

Group Information

Group Name: Synergy:

Group Members: Utsav Dholakia (uxd150130),

Sagar Mehta (sam150930),

Siddharth Shah (sas151830)