

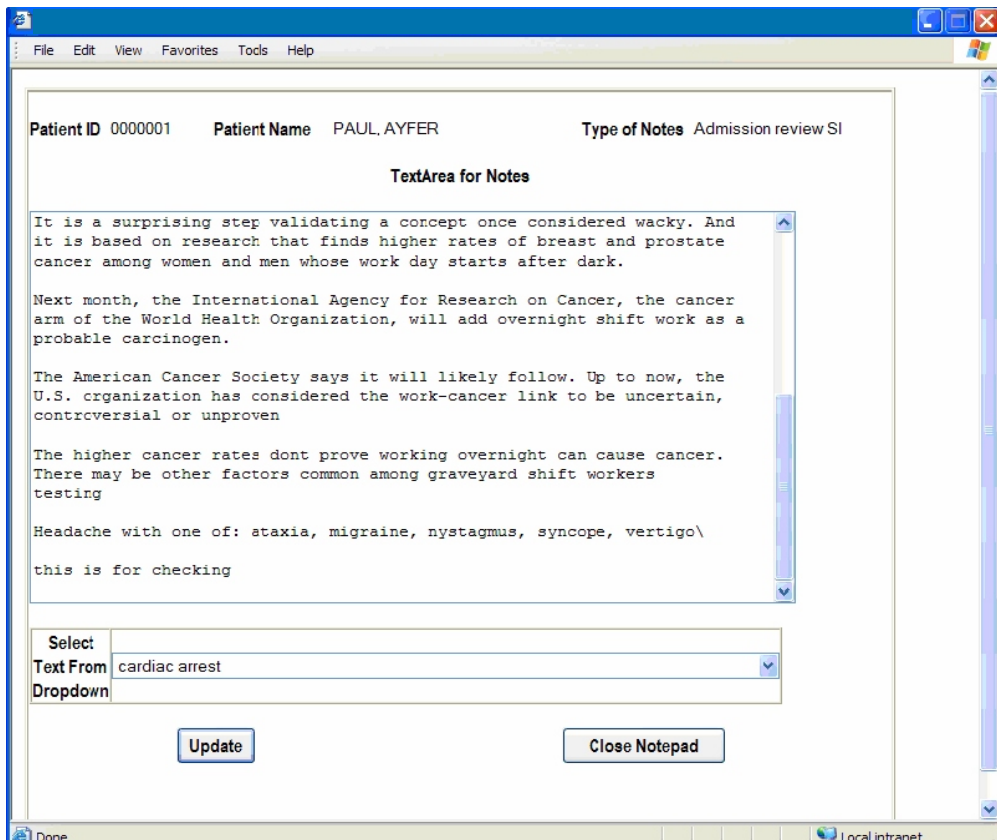
TEXTAREA FUNCTIONALITY IN HATS APPLICATION

HOW TO CREATE TEXTAREA IN HATS APPLICATION WHICH PERFORMS INSERTION, DELETION, REPLACEMENT AND UPDATION OF TEXT

In HATS Application, We may come across a situation where we need to create a TextArea in which User can insert any text, replace selected text and also update / append more text in the Textarea.

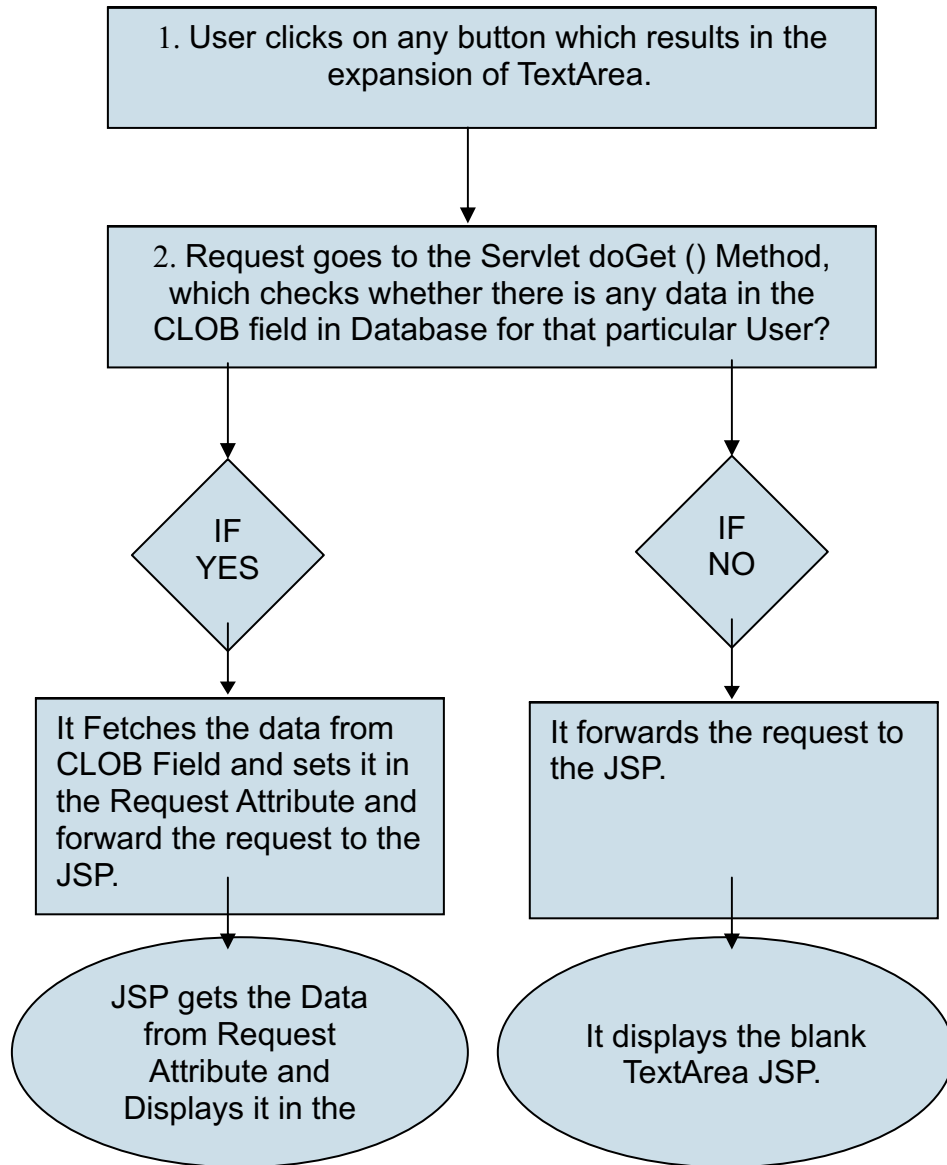
To achieve the above functionality, we need to have one Custom Servlet in between our HATS Application and the JSP. This Custom Servlet follows the pattern of Front Controller and Intercepting Filter and controls the whole logic of forwarding the request coming from HATS Application to the custom JSP. This JSP actually contains the TextArea or we can say that it contains the View.

When the Controller Servlet receives the request from HATS Application, it executes its doGet () method and queries to the database that whether it contains any data in its CLOB field for that particular user Id. If yes, then it fetches that data from the CLOB field and sets that data in the request object and forward the request to the JSP. We can set any number of attributes in the request object before forwarding it to any other resource. When the JSP receives the request, it gets the data from the request object and displays that data to the user in the TextArea. It will look something like the following diagram:

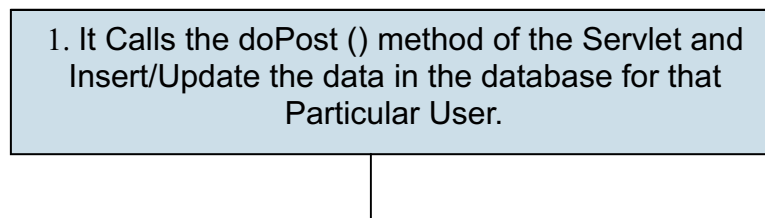


Now, Let suppose User Writes some more text in the above Textarea and click on the UPDATE button, then the doPost () method of the Custom Servlet will be called in the result of this action. In the doPost method we take the updated data from the JSPs Textarea and update that in the Database CLOB field with the help of Update query for that particular user. If it is a new User then it just Insert the Textarea data in the database with the help of Insertion query for that particular user.

Let us review the whole process with the help of the following diagram:



In the JSP, When the User clicks on the Update Button, it performs the following tasks:





© Copyright IBM Corporation 2010
IBM Global Services
Route 100
Somers, NY 10589
U.S.A.
Produced in the United States of America
08-10
All Rights Reserved

IBM, the IBM logo, ibm.com, Lotus®, Rational®, Tivoli®, DB2® and WebSphere® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml Other company, product and service names may be trademarks or service marks of others. The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software. This document illustrates how one organization uses IBM products. Many factors have contributed to the results and benefits described; IBM does not guarantee comparable results elsewhere.