Infoware Services

Abstract

This project will be designed and developed for FAQ's posted to various departments. The University examination branch wants to maintain all the frequently asked questions in the database subject wise, to available them to all the students. And they will update the questions as and when required. The primary goal of Infoware Services is to provide an environment that is both convenient and efficient to use in retrieving and storing database information.

The database system must provide for the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results.

In "Info ware Services" maintaining the details of various frequently asked questions posted by the students to the administrators of consult department are to be kept in a data base and also can be retrieved easily when the admin answers that question.

Existing System

The existing system is a manual one. When the student wants to clear his query he personally contacts the concerned person in the university. The concerned person will clear his doubt.

Disadvantages:

- *Difficulty in contacting the concerned person.*
- *Difficult to handle more students for administrator.*
- *Difficulty of maintaining frequently asked questions in a data base*

Proposed System

The Info ware Services is to replace the existing manual system with a software solution.

The University examination branch wants to maintain all the frequently asked questions in the database subject wise, to avail them to all the students. And they will update the questions as and when required in online.

The proposed system has one super user to control the data, appointing the administrators in each department. The administrator is responsible to answer all the questions posted by students.

When the student post a question in certain department the respective administrator will answer the question and a copy of the answer will be sent to the student.

Merits of This System

- *Faster processing when compared to existing one.*
- *Maintaining frequently asked questions*
- *Modifications of answers can be carried out immediately*
- *Administrator can handle more number of students.*
- *Easy maintenance of administrator details.*

Scope of the System

The proposed system scope is limited to Intranet only. In this system there is communication between administrator and student, but not between student and student. One department is assigned to one administrator. So that he is not able to answer the questions more than one subject in spite having knowledge on another subject. We can also implement internationalization (i18n) to support user interface in various/local languages

.

Module Description

The project contains three modules

1. *Super User*
2. *Administrator*
3. *Student*

Super User

The responsibility of the super user is to centrally control the data, appoint administrators to all the departments and shifting the administrators' one department to other and replacing with others when there is a vacancy in any departments.

Administrator

The major responsibility of administrator is to answer all the questions posted by students on a regular basis. The administrator can add, modify or remove questions from all the categories. Administrator also reserves a right to post a question into a particular category even if it is not a question frequently asked. The administrators have the capability of changing his password as when he requires.

Students

These are the users at the lower level of our abstraction. Before posting a question, the student has to select the department to which it belongs. A copy of the answer will be sent to the student after the administrator answered his question.

Features to be implemented

- *Session management*
- *Connection pooling*
- *Normalized database*
- *Prevention of duplication login*
- *Design patterns*
- *Three-tier architecture*
- *Maintainability*
- *Easy deployment with Ant script.*
- *Exception handling*
- *Client-side validations*

Technologies to be used

- ***Web Presentation: HTML, CSS***
- ***Client – side Scripting: JavaScript***
- ***Programming Language: Java***
- ***Web based Technologies: JNDI, Servlets, JSP***
- ***Database Connectivity API: JDBC***
- ***Build Tool: ANT***
- ***Debug Tool: Log 4J***
- ***CASE tool: Rational Rose, Visual Paradigm, Enterprise Architect***
- ***Backend Database: Oracle/SQL Server/MY SQL/MS Access***
- ***Operating System: Windows XP/2000/2003, LINUX, Solaris***
- ***J2EE Web/Application Server: Tomcat/Weblogic/Websphere/JBoss/Glass Fish***
- ***IDEs: Eclipse with My Eclipse plug-ins/Net Beans/RAD***
- ***Browser: IE/Mozilla***

Hardware requirements

- *Pentium processor -------- 233 MHZ or above*
- *RAM Capacity -------- 128MB*
- *Hard Disk -------- 20GB*
- *Floppy disk -------- 1.44 MB*
- *CD-ROM Drive -------- 32 HZ*
- *KEYBOARD -------- 108 Standard*