

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДОКУМЕНТООБОРОТА КАФЕДРЫ: ОСОБЕННОСТИ РЕАЛИЗАЦИИ И ИСПОЛЬЗОВАНИЯ

Р.Л.Сорокин¹, А.В.Лепеш²

¹ - студент 4 курса, экономического факультета, группы 871502, Белорусского государственного университета информатики и радиоэлектроники

² – научный руководитель, ассистент кафедры экономической информатики, Белорусского государственного университета информатики и радиоэлектроники, Минск, Платонова 39-807, тел. (17) 239-89-92, e-mail: a.lepesh@belcaf.minsk.by

Аннотация. Данная статья представляет собой одну из цикла работ, посвященных описанию автоматизированной системы документооборота кафедры “DOCumentTurnOver”. Проект “DOCTOR” представляет собой web-приложение, реализующее архитектуру клиент-сервер на базе java-технологий. Основное внимание в этой работе уделено тонкостям программной реализации и особенностям использования приложения.

Ключевые слова: документ, клиент, сервер, Java, web-приложение, пользователь.

1. Введение

Деятельность сотрудников кафедры в высшем учебном заведении ориентирована на обеспечение учебного процесса и включает в себя большое количество различных функций. Это и проведение занятий, и разработка учебно-методических пособий, и оформление всевозможных отчетов и других официальных документов.

В своей работе сотрудникам приходится иметь дело с большим количеством самых различных электронных документов, часто хранящихся на разных компьютерах в личных папках сотрудников. Причем эти документы обычно хранятся без всякой системы, могут дублироваться, что сильно затрудняет поиск нужной версии даже для владельца документа, не говоря уже о ситуации, если документ понадобится другому сотруднику.

Для упорядочения этого процесса на кафедре и была создана представляемая автоматизированная система учета документов кафедры «DOCument TurnOver (DOCTOR)».

2. Предназначение и основные функции системы

Данное приложение основано на следующем представлении предметной области. Существуют пользователи (ими могут быть как сотрудники кафедры, так и другие

люди), которые могут входить в одну или несколько групп пользователей (user, administrator). Структура хранения документов напоминает обычную файловую систему: документы находятся в папках (группы документов), причем одна группа документов может включать в себя несколько других групп.

Это приложение предоставляет возможность различным пользователям в соответствии с их статусом и правами осуществлять доступ к документам и группам документов кафедры для реализации следующего ряда функций.

Пользовательские функции:

1. просмотр документа;
2. редактирование документа;
3. создание нового документа в какой-либо группе;
4. поиск документа.

Функции администрирования:

1. создание и удаление групп документов;
2. создание и удаление пользователей и групп пользователей;
3. выдача прав пользователям.

3. Структура проекта и особенности реализации основных функций.

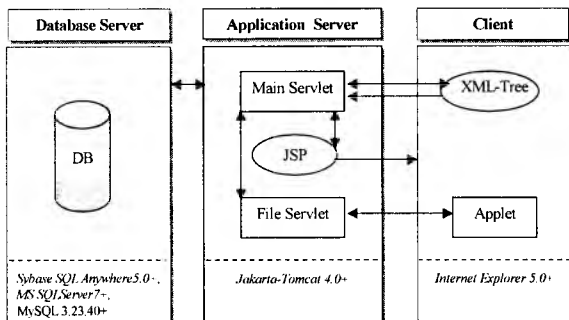
Система «DOCTOR» была создана как Web-приложение, реализующее архитектуру клиент-сервер, на основе использования одних

из наиболее перспективных на сегодняшний день интернет-технологий – JavaServer Pages, сервлетов и апплетов Java, а также XML/XSL для организации отображения дерева каталогов документов.

Схема функционирования приложения отображена на рисунке (рис. 1):

Рисунок 1.

Схема функционирования приложения.



В качестве СУБД для данного приложения может быть использован один из нижеследующих серверов баз данных:

- Sybase SQL Anywhere 5.0 и выше;
- MS SQL Server 7 и выше;
- MySQL 3.23.40 и выше.

Серверная часть приложения представлена двумя сервлетами и несколькими страницами JSP. В качестве Application-сервера используется бесплатный кросс-платформенный сервер приложений Jakarta-Tomcat-4.0.

Main Servlet отвечает за функции:

- аутентификации пользователя;
- установление соединения с базой данных и взаимодействие с ней;
- генерацию XML-дерева структуры каталогов документов;
- перенаправление пользователю нужной JSP-страницы.

FileServlet осуществляет:

- взаимодействие с клиентом (Applet) для обмена файлами;
- внесение соответствующих изменений в базу данных.

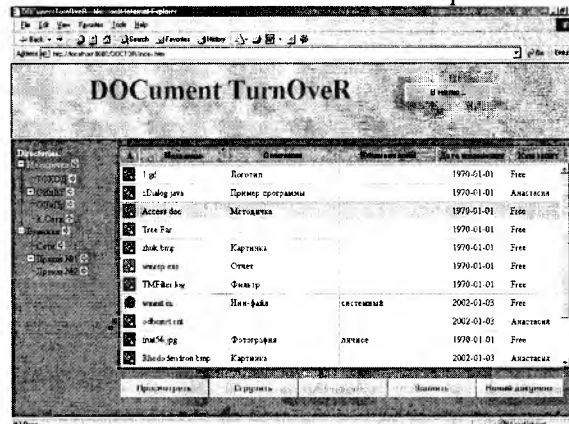
В роли клиента выступает Web-браузер, из которого пользователь загружает данное приложение. В этом качестве должен использоваться Internet Explorer версии 5.0 и выше, благодаря наличию в нем встроенной поддержки XML документов.

Клиентская сторона состоит из 3-х частей: XML-дерева каталогов документов, непосредственно в HTML списка документов, сгенерированного страницей JSP, и апплета для реализации функциональности.

Внешний вид клиентского приложения представлен на рисунке (рис. 2)

Рисунок 2.

Клиентское приложение.



Расположенное в левой части приложения дерево представляет собой средство отображения иерархической структуры каталогов документов.

При выборе в нем соответствующего листа (узла) в правой части приложения отображается список документов, содержащихся в соответствующей папке.

С этими документами пользователь в зависимости от своего статуса и имеющихся у него прав может осуществлять некоторые из ряда следующих функций: (кнопки – просмотреть, забрать на редактирование, загрузить новую версию, добавить новый документ, удалить).

Обратим внимание на некоторую специфику в реализации этих функций.

Просмотр осуществляется в том же фрейме, где находится список документов, при условии, что у клиента установлен соответствующий plug-in для просмотра документов данного типа. В противном случае документ сгружается на ПК пользователя и просматривается соответствующим приложением (вьюером, редактором), понимающим формат документа.

Поскольку все документы хранятся на сервере, для редактирования документа

пользователю необходимо сначала скопировать файл интересующего его документа к себе на ПК, внести все требуемые изменения, а затем загрузить файл обратно. При этом надо учитывать, что в этот период времени другие пользователи не должны иметь возможности забрать документ на редактирование, а могут только просмотреть старую версию. Поэтому когда копия файла передается пользователю для редактирования, сервер помечает этот документ в базе данных как редактируемый, и никто другой не может его взять, пока файл не будет возвращен на место.

В связи с тем, что протокол HTTP не предоставляет полноценных возможностей для копирования файлов, то есть отсутствует возможность оповестить сервер, если во время копирования произошел сбой либо пользователь отменил действие, для копирования файлов документов в данном приложении "DOCTOR" была создана следующая подсистема:

- на стороне клиента – апплет,
- на стороне сервера – *File Servlet*.

Еще один нюанс заключается в том, что Java-апплетам в целях обеспечения безопасности разрешен доступ лишь к ограниченному набору операций на компьютере конечного пользователя, в число которых по вполне понятным причинам не входит копирование файлов на жесткий диск.

Разрешить эту проблему позволяет механизм защиты Java Security API, предоставляющий возможность цифровой подписи апплета (digital signing). Апплет, задействованный в "DOCTOR", подписан самостоятельно сгенерированным сертификатом.

Достоинства системы

В заключение следует отметить, что представленная система учета документов «DOCTOR» удобна и эффективна в использовании, поскольку она обладает следующими достоинствами:

1. Работа с данным приложением осуществляется посредством Web-браузера, благодаря чему не требуется установка

дополнительного программного обеспечения на стороне клиента.

2. Простота в использовании, дружелюбность интерфейса и отсутствие необходимости конфигурирования системы на стороне пользователя.
3. Платформенная независимость серверной части приложения (это преимущество нам дает использование Java-технологий).
4. Возможность применения приложения как в локальных сетях, так и в глобальной сети Интернет.
5. Легкость в администрировании на стороне сервера.

Таким образом, разработанная система представляет собой достаточно удачное решение проблемы учета документов кафедры и позволяет пользователям (т.е. в первую очередь сотрудникам кафедры) избежать путаницы и ненужной траты времени, и в результате повысить эффективность своей работы.

Проект "DOCTOR" постоянно развивается. Одно из новшеств, которые появятся в ближайшем будущем, - переход на межбраузерную независимость и, как следствие, межплатформенную еще и для клиента.