"CST8218 Project" - Project Proposal for BlogShare

Team Members : Vincent Quirion

Jan Litwinski

Derek Roesner

The purpose of BlogShare is to provide an online interface where users can create blogs and share them with either the public or select users.

Each BlogShare user can create content in the form of blogs and then posts within each blog. Each user may also be able to edit select content created by other users.

By default the author of each blog is exclusively able to edit their content.

However, the author may grant other users permission to edit either individual posts within their blogs or the entirety of one of their blogs.

This makes content sharing quite flexible.

For Example:

The author can share the entire blog as read-only, then specific posts

as editable by all users, and other posts as editable by individual users.

Or

The entire blog as editable by the public other than for specific excluded

users, then make certain posts read-only for all users.

With this is in mind, a goal of BlogShare is to encourage creative collaboration of content, meaning BlogShare is best experienced when users allow their content to be edited by other users.

Collaboration will unavoidably lead to unwanted edits on users content.

To remedy this problem an edit history will be maintained for each editable item. This will allow the author to approve any changes, and if the author hasn’t made a decision after a set amount of time, the change is always approved.

There is also a need to develop a feature that will manage the situation where multiple users desire to edit the same file at the same time.

BlogShare will be displayed within the web browser, using JSP, HTML, CSS and JavaScript.

The layout for BlogShare consists of the following files.

HTML files:

Home.html - Derek

Profile.html - Jan

ProfileEdit.html - Jan

Blog.html - Vince

BlogEdit.html - Vince

PostEdit.html - Vince

SearchResults.html - Derek

Error.html - Derek

8 html pages in total.

JSP files:

Home.jsp - Derek

Profile.jsp - Jan

ProfileEdit.jsp - Jan

Blog.jsp - Vince

BlogEdit.jsp - Vince

PostEdit.jsp - Vince

SearchBar.jsp - Jan

SearchResults.jsp - Derek

Error.jsp - Derek

9 jsp pages in total.

CSS Pages

LookAndFeel.css

Will contain styling classes for:

* Text
* Buttons
* Backgrounds
* HTML Elements (tables, divs)
* etc

PageResize.css,..may also end up being a .jsp file.

SearchBar.css

The webpages will be stored on a Server Running a version of TomCat 8.0

BlogShare’s content, user credentials and associated data will be stored in a

MYSQL database named BlogShareData.

The database tables needed to represent the data to be stored are.

- Users

- Blogs

- User\_Blogs

- Posts

- User\_Posts

- Edits

- User\_Edits

The cardinality of these tables is:

ZERO or MANY *Users* author make ZERO or MANY *Blogs*

ONE *Blog* is authored by ONE and ONLY ONE *User*

ZERO or MANY *Users* may create ZERO or MANY *Posts*

ONE *Post* is created by ONE or MANY *Users*

ZERO or MANY *Users* may make ZERO or MANY *Edits*

ONE *Edit* is made my ONE and ONLY ONE *User*

ONE *Blog* may contain ONE or MANY *Posts*

ONE *Post* is contained in ONE and ONLY ONE *Blog*

ONE *Blog* may have ZERO or MANY *Edits*

ONE *Edit* is applied to ONE and ONLY ONE *Blog*

ONE *Blog* can be edited by ONE and ONLY ONE *User*

ONE *User* can edit ZERO or MANY *Blogs*

ONE *Post* may have ZERO or MANY *Edits*

ONE *Edit* is applied to ONE and ONLY ONE *Post*

ONE *Post* may be edited by ONE or MANY *Users*

ONE *User* may edit ONE or MANY *Posts*

The program will be delivered in four versions, on the following dates.

0.1 - Sept 25th

0.2 – Oct 9th

0.3 – Nov 12th

1.0 – Nov 21th????

Each version will contain different features and will build upon each other to complete the final application. The system architecture will likely evolve as development is completed.

* 1. –

\*Vince & Jan - Design and Develop CSS Style Sheets for Look and Feel of BlogShare

\* Vince & Derek & Jan - Design and Develop static versions of web pages.

\* Derek - Design relational model of Data Base Architecture.

Distribution of tasks to be determined later in development.

0.2 –

- Bug Fixes with basic version of web pages.

- Design and Develop basic dynamic content of web pages.

(Create users, create blogs/posts, configure blog/post settings , etc)

* Implement basic Database Architecture
* Implement Database Connectivity between web page and database.

- Implement basic system architecture.

0.3 -

- Bug Fixes with content of web pages.

- Bug fixes with database architecture.

- Continue to develop dynamic content of web pages.

(Edit contents of post, build edit history, remove edit, credit co-authors)

* Implement database architecture to include an edit history.
* Improve system architecture as needed.

1. –

* Bug Fixes with content of pages.
* Bug Fixes with database architecture.
* Bug Fixes with system architecture.

Each member is responsible for the following aspects of the version presentations.

Each member is responsible for the following aspects of system documentation.