CS2815 – Small Enterprise Team Project

Initial Project Description

Information Technology (IT) Assets Metadata Repository

The goal of the project is to develop a holistic Web based system that supports the metadata-based organization of the different source-code related assets of a software company.

Traditionally, software companies rely on source code version control systems (e.g. GitHub and SVN) to store and manage their source code artefacts. However, such systems typically work on a project or product level with a focus on source code versioning. Hence, they provide limited support in providing an "umbrella", comprehensive view of the source code assets landscape of the whole company. Moreover, their search capabilities, documentation support and management of assets associations is fairly limited.

Initial High-Level Requirements

- The system should let users store assets, associate assets with each other and store meta data about those assets so that the collective knowledge of the company can be easily searched and accessed.
    - An asset is a piece of knowledge or a concept captured in a document. An example of an asset might be a class that wraps authenticating users through a service like LDAP. There might be multiple classes or modules of this implemented in several languages each of which could be itself an asset. It should be possible to associate these assets to indicate that they are linked along with other relevant assets such as: documentation on specific instances of LDAP services for example the LDAP services that RHUL uses to authenticate users; projects that make use of these classes and modules; people who have worked on those projects.
    - Assets should have a type and users should be able to create new types of assets. These types should define the set of attributes that can be stored along with an asset e.g. title, link, line number, programming language. Assets of the same time should have the same set of attributes associated with them.
    - Assets should have a type and users should be able to create new types of assets. These types should define the set of attributes that can be stored along with an asset e.g. title, link, line number, programming language. Assets of the same time should have the same set of attributes associated with them.
    - Documents should be stored as links to external files.
- Web-based RBAC (role-based access control) support: for instance, admin-level users will be able to create, update and delete assets types, normal users will be able to create/register, update and delete specific assets, viewers will have read-only access.
- The system should support search, both keyword based and "filter-based" search that would retrieve all relevant information for an asset which is stored in the repository, its associations to other assets (in graphical or tabular form), its documentation and relevant source code links.

- The system should keep a relevant log of all updates that users make in its content. Moreover, in the system's UI, such updates should be visible to users navigating an asset and its associations.

- As an extension, the system could support discussion boards on assets and/or include the notion of "project" and people (e.g. associate assets to projects and people).

Further Notes

This project has a real client, and the goal is that its resulting product is used in a production setting. All intellectual property (IP) related to this project, including any aspects of its development and associated materials, will belong to the client.