Design Document for User Authentication

# DEVELOPERS

Chandrashekhar Patil

# DESIGN OVERVIEW

The User Authentication Toolkit will provide a web-based mechanism for managing user access to the Ecommerce Website. It will contain an object-oriented Java library of methods that apply to user authentication as well as a sample application that utilizes the library. It will be built on the basic authentication model of the Apache HTTP Server, making use of an access control model based on the concepts of groups, realms, and. http access files for managing user authentication. The database authentication modules built into Apache and Java will be used as the underlying database structure.

Proof of Concept will:

1. Have finished all pieces necessary to the Download Wizard. These include:
   1. User Admin Page
      * list users
      * delete users
      * add users
      * edit users

Version 1.0 will:

1. Provide an object-oriented Java library containing basic methods pertinent to user authentication. These methods will be non-specific and reusable.
2. Adhere to the access model (yet to be formalized) based on realms.
3. Provide a web-enabled management tool (i.e. web-interface) that will have the following capabilities:
   * add users to specific realms/groups
   * delete users from realms/groups
   * change access for realms/groups
   * change passwords
   * mail passwords if forgotten
4. Provide a sample application to illustrate the basic uses of the Java library.
5. Use a standard Apache password protection scheme.
6. Maintain a high but adjustable level of security for the [internal] website.
7. Allow authorized users the ability to control and modify access to internal website tools on an individual user basis via a web-interface. This is up for further discussion.
8. Eliminate the need to manually edit .htaccess files.

# PLATFORM

All relevant Java 11 and will run on an Apache server installed on a windows machine.

# DESIGN DETAIL

1. Web Interface: This will be implemented using an html form which calls a java script which in turn uses the authentication library. This will look like html --

>Java script --> Java authentication library.

1. Database: There must be some way to store, retrieve, and edit user access information. This will most likely take the form of the database authentication modules built into Apache and Java
2. Methods/Subroutines in authenticate.pl: These will include modified versions of several subroutines. As well as additions of new subroutines that are needed. These will include:
   * create\_user()
   * remove\_user()
   * change\_passwd()
3. The details of the access model have yet to be hammered out.
4. Implementation details: The library authenticate. will be included in authApp. using a require statement, thus giving authApp.pl access to all its subroutines and global variables.