****

**Submitted by**

**Rahul Kumar Thai Valappil**

Epita Login- thaiva\_r

**IDENTITY ACCESS MANAGEMENT**

**Technical Specification**

Date: May-15-2016

Version :1.0

**Table of Contents**

**1 Subject description...................................................................................**

**2 Subject Analyses.....................................................................................**

2.1 Major features.........................................................................................

2.2 Application Feasibility.........................................................................

2.3 Data description......................................................................................

2.4 Expected results......................................................................................

2.6 Algorithms study …………………………………….................................

2.5 Scope of the application (limits, evolutions) ..............................

**3 Conception………………………………………………………………………………………...**

3.1 Chosen algorithm ……………………………………………….

3.2 Data structures........................................................................

3.3 Global application flow.........................................................

3.4 Global schema and major features schema..................

**4 GUI descriptions.................................................................**

**5 Configuration instructions.........................................................................**

**6 Commented Screenshots..........................................................................**

**7 Bibliography……………………………………………………………………………………….**

**1. Subject description:**

Identity Access Management(IAM) System is an administrative application that deals with manipulation and management of individual identities, their authentication within or across system. The application must perform basic CRUD (Create, Read, Update and Delete) operations on a persistent storage system such as database. The term **“User”** is used as “the person who will manage the information” and the term **“Identity”** is used to signify the “information of user in the system”.

**2. Subject Analysis:**

**2.1 Major features:**

The application must allow the users to maintain and manipulate their identity information through a web interface. Every identity entity being comprised of first name, last name, email address and data birth and these details will persist in server database. Additionally, it should include a layer of security for a login process that will restrict the access from unauthorized users.

Application allows user to perform below functions related to his/her information:

* Configure User account for authentication with purchased license
* Login with valid credential
* Create an Identity
* Search identities from the system
* Update an existing Identity
* Delete an existing Identity

**2.2 Application Feasibility:**

The application has to be accessed through a web browser. Depends on the server configure, the application would allow to deploy on a server with full capabilities of being accessible from within the network, or even online.

**2.3 Data description:**

The application will use Derby database with tables for both Identity management and user authentication. The Hibernate and Spring frameworks are used to communicate between application business layer and data layer. Following are the data representation corresponds to Identity and User table.

* **Identity:**

Identity represents following data.

* Id – Automatically generated field within hibernate
* firstName -String
* lastName -String
* email – String
* birthdate - Date
* **User**

User represents following data:

* Id - Automatically generated field within hibernate
* Username – String
* Password – using java simplified pooled digester to encrypt the data.

Additionally, license needs to register a user is hard coded in Spring application context xml, So currently additional user creation is not possible.

**2.4 Expected results:**

End user should able to perform following functionalities and HTML pages will be loaded with some JavaScript and CSS functionality for the ease of use and aesthetics.

* Configure user authentication
* Perform user authentication
* Create an Identity
* Search Identities
* Update an Identity
* Delete an Identity

The user will have two main HTML pages, one for user authentication and other for performing CRUD operations.

**2.5 Algorithms study**

Application should adopt a simple MVC paradigm with help of Spring framework as business layer and Hibernate session factory as data layer. When user performs actions the view should send data request to the controller. This controller will contact data model in order to produce data requested and send back to the view where it will have visualized in HTML format. To improve security of the system user password data should be encrypted before storing in database.

**2.6 Scope of the application (limits, evolutions):**

* **Limits**
* The application will do no more than the basic CRUD operations of Identity and user authentication
* Deletion or update can perform only one identity at a time
* Currently user management option is not exposed to the user layer
* Identity fields and criteria for search are not configurable
* License issuer is not implemented, so currently using a hard coded license which restricts registration of multiple user account in one system
* **Evolutions:**
* Separate configuration or setting page for managing user accounts
* License issuer module which is responsible for issuing new license and deciding the user access level
* An option to configure Identity fields and criteria for search in administrative level
* Multiple Identities can be deleted or updated at a time.

**3. Conception:**

**3.1 Chosen algorithm**

Chosen Java simplified encryption(jasypt) to encrypt user password before storing to database. The standard and simplified way allows to configure the algorithm through spring application context. The jasypt hashing lets to specify the algorithm (and provider) to be used for creating digests, the size of the salt to be applied, the number of times the hash function will be applied (iterations) and the salt generator to be used. Following are the current digest properties.

pool Size - 2

Algorithm - SHA-256

Iterations - 100

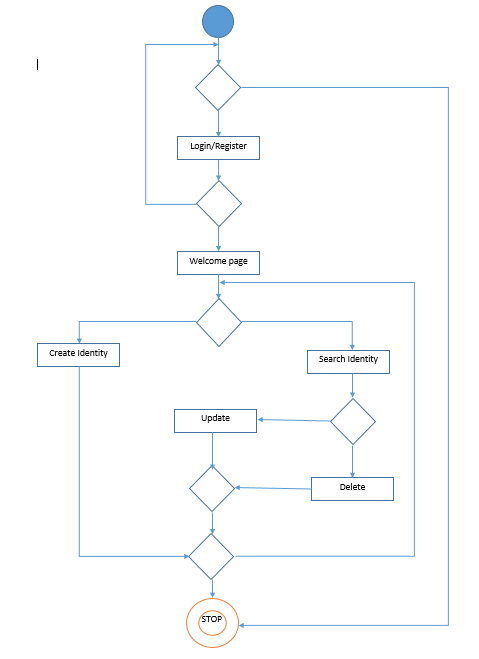
salt Generator - org. jasypt. salt. RandomSaltGenerator

salt Size Bytes - 16

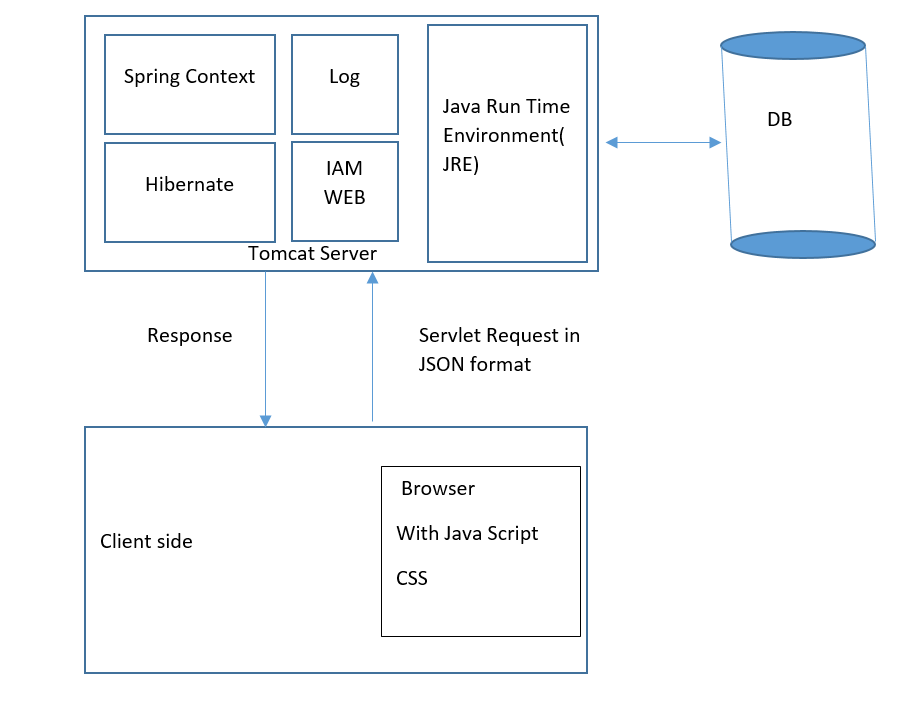
**3.2 Data Structures**

Application mainly Spring Framework, so it provides a comprehensive programming and configuration model for application. Spring allows to execute operation such as database without having to deal with actual database API’s through dependency injection and inversion of control. Also application using Data Access Object patterns which allows flexibility in choosing business service level of identity data. Currently application using Hibernate framework for mapping identity object model to derby database.

**3.3** **Global application flow**



* 1. **Global schema and major features schema**

****

**4. GUI Operations Description:**

**Login page**

Page consist of user login and registration options. Already registered user can enter their username and password for authentication otherwise need to register with valid license by clicking register button.

**Home page**

The page where user can choose their CRUD operations. Home page will get called once the user successfully authenticates his credential.

**Create Page**

The page provides facility to create new identities by entering details. Application will redirect to create page once user clicks on Create option from Home page. The page notify to user if any error happened during the create operation.

**Search Page**

The page facilitate user to search identity from their email address and provide an option to update or delete the identities.

**5 Configuration instructions**

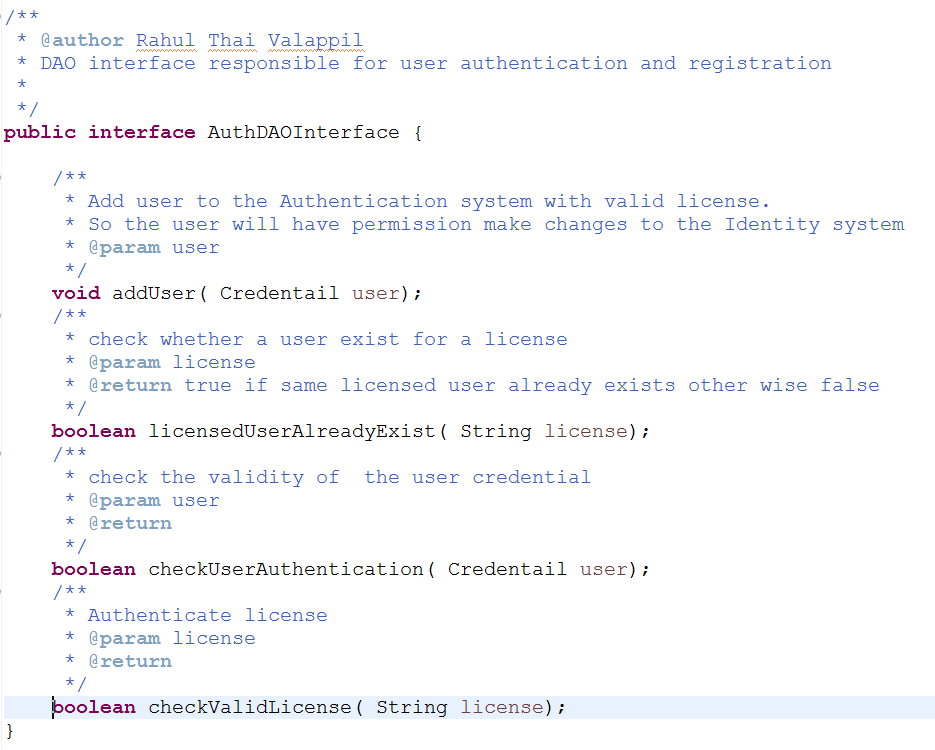
1. Install latest Java run time environment
2. Install Apache tomcat server and start the server
3. User should install the Apache Derby database before doing any operation and start the derby Server by executing startNetworkServer.bat from Derby installation folder. Please visit following links for more details <https://db.apache.org/derby/papers/DerbyTut/install_software.html>
4. Deploy the IAMWEB application using the Tomcat Host Manager.
5. Configure log file location in log4j.properties from resource folder

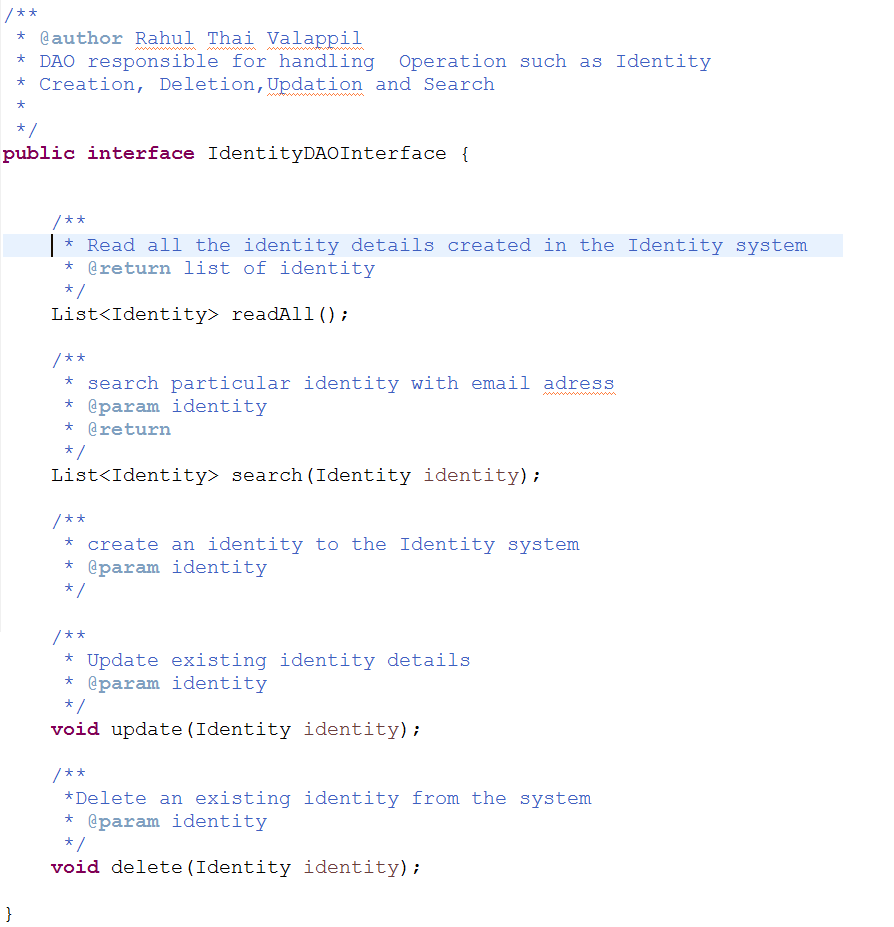
log4j.appender.file.File=./log/trace.log

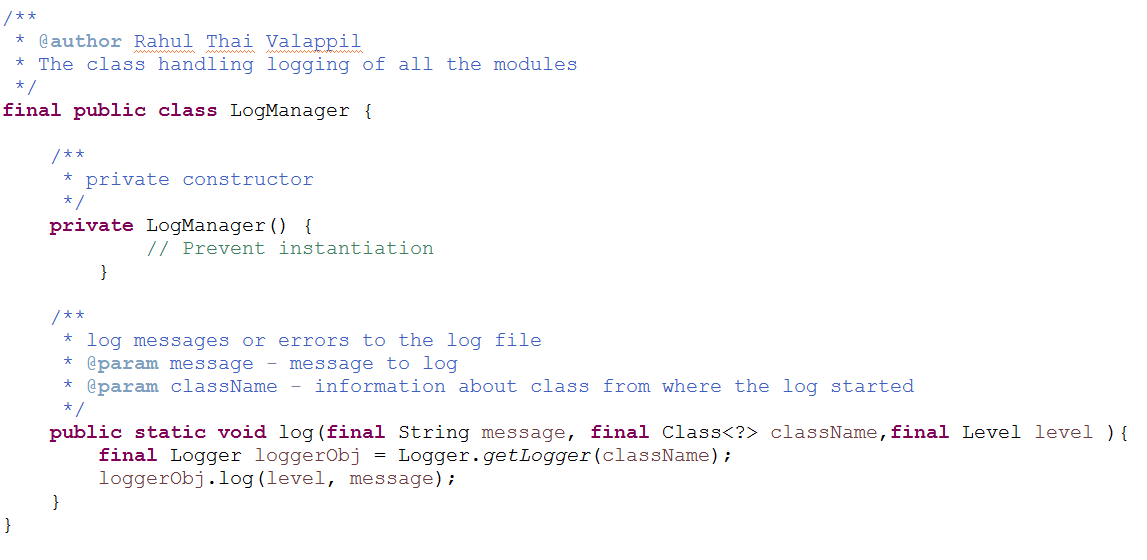
1. For First time user. register with valid license is mandatory to access the Identity management system.

License - **23a4a4bd-83a9-4b95-be9a-638bf5fd35f3**

**6. Commented Screenshots**

****

****

****

**7. Bibliography**

1. <http://thomas-broussard.fr/work/java/courses/index.xhtml>

2. <https://db.apache.org/derby/papers/>

3. <https://docs.oracle.com/javase/tutorial/>

4. <http://www.jasypt.org/encrypting-configuration.html>

5. <https://dzone.com/refcardz/mockito>

6. http://logging.apache.org/log4j/2.x/

7. <http://spring.io/> and http://hibernate.org/