

**By Kondalarao.Palugudi**

**REPORT**

**Identity and Access Management System** [**IAM Project**]

**Overall Contents**

1. Subject Description

Introduction

i). Overview

ii). Objective

iii). Development Process

2. Subject Analysis

i). Major Features

ii). Application Feasibility

iii). Data Description

iv). Expected Results

v). Scope of the application

3.Conception

i). Chosen Algorithm

ii). Data Structures

iii). Global Application flow

iv). Global schema and major features schema

4. Console operation Description

5. Configuration Instructions

6. Commented Screenshots

7. Bibliography

Overview

1. Setup a Dynamic Web project and convert it to maven project

2. Add the dependencies - be sure to clean install in maven to put your needed jars in the project so you won't have library issue while coding

3. Setup the database

4. Create model class - add JPA annotations

5. Create DAO components - @Repository

6. Create Service components - @Service

7. Create Controller component - @Controller

- be sure to put BindingResult immediately after @ModelAttribute or else you will got issues

8. Create the View components - JSPs

9. Configurations

- jdbc.properties

- spring-servlet.xml

- log4j.xml

- hibernate.cfg.xml

- finally web.xml

**1. Subject Description**

**Introduction**

**i). Overview:**

This report will help to provide the total information of IAM project development.

Identity and Access Management (IAM) can be defined as following:

A comprehensive set of processes that enable end users to securely access a broad range of internal and external IT systems that controls the digital representation of users and manage information about identities.

**ii). Objective**

* The main objective of this project is to develop an application by using Java Core and Web

Java concepts

* Design Patterns
* Java Server entities (JSP/Servlet)
* Tools like And or Maven
* Framework as Spring, Hibernate

to retrieve, modify, delete and search the information in the database.

Objectives used for Application:

1. Authenticates a user

2. Create an Identity

3. Update an Identity

4. Delete an Identity

5.Search an Identity

iii). Development Process

PLATFORM : Windows 8.1

IDE : Eclipse

LANGUAGE : Core Java

**2. Subject Analysis**

**i). Major Features**

* Highly user-friendly
* Platform Independent
* Easy to use
* Robust
* Data entry restricted to avoid errors
* Clean separation of various components
* Easy Modification

**ii). Application Feasibility**

* This current application is a prototype of a system that can be created for employing a highly secured

environment of Identity and access Management.

* The costs are much reduced as we do not depend on graphical interface, instead look for a high system performance
* All the components used in this application are open sources like development platform, servers, and databases .

**iii). Data Description**

The data description and data access objects are clearly specified below.

The Schema for data is <IDENTITY\_UID,IDENTITY\_DISPLAYNAME,IDENTITY\_EMAIL,IDENTITY\_BIRTHDATE>

IDENTITY\_UID : INT, Auto generated, Unique

IDENTITY\_DISPLAYNAME : STRING

IDENTITY\_EMAIL : STRING

IDENTITY\_BIRTHDATE : INT

DAOs

* **Authenticate** : This module takes user name and password. This module validates user before login to IAM system

Input parameter : user name, password

Output parameters : Authentication Accepted / Denied

* **createIdentity** : This process is used to create a new identity in the database.

Inputparameter : uid,displayName, email, birthdate(Id is generated automatically)

Output parameters : Entry added to database

* **deleteIdentity** : This process is used to delete an identity from the database.

Inputparameter : uid

Output parameters : Identity deleted from the database.

* **updateIdentity** : This process is used to change any record which is already exist in the database and updates the database according to enter Identity.

Input parameter : uid,displayName, email, birthdate.

Output parameters : Identity modified in the database.

**iv). Expected Results**

* The end result of the application can be looked as a highly sophisticated, user friendly and secure tool created for Identity and access management.
* This tool is capable of authenticating the user, creating a new identity, updating an existing identity,and deleting an identity from the database.
* The database used is a derby database.
* The tool needs to communicate with the database and return with the results in quick time.

**v). Scope of the Application**

**Scope:**

* Privacy: Online transaction, whether financial or exchange of information, could be greatly improved by the adoption of IDM solutions which focus on privacy.
* Improved user experience, Cost savings, security policy enforcement etc.

**Limitations:**

* Does not open the system to manage permissions and attributes of users.
* Possibility of decrypting the system password.
* Lack of user GUI/web interface which could disturb first time users.
* Lack of added feature like in modern address book.

**Evolution:**

* We are working to enhance the IAM as a complete web-based software, users and login and keep details, export details, import to new system, send SMS from there and save other information.

**3. CONCEPTION**

**i). Chosen Algoritm**

The algorithm that we have used is the exact match algorithm. It can be seen below.

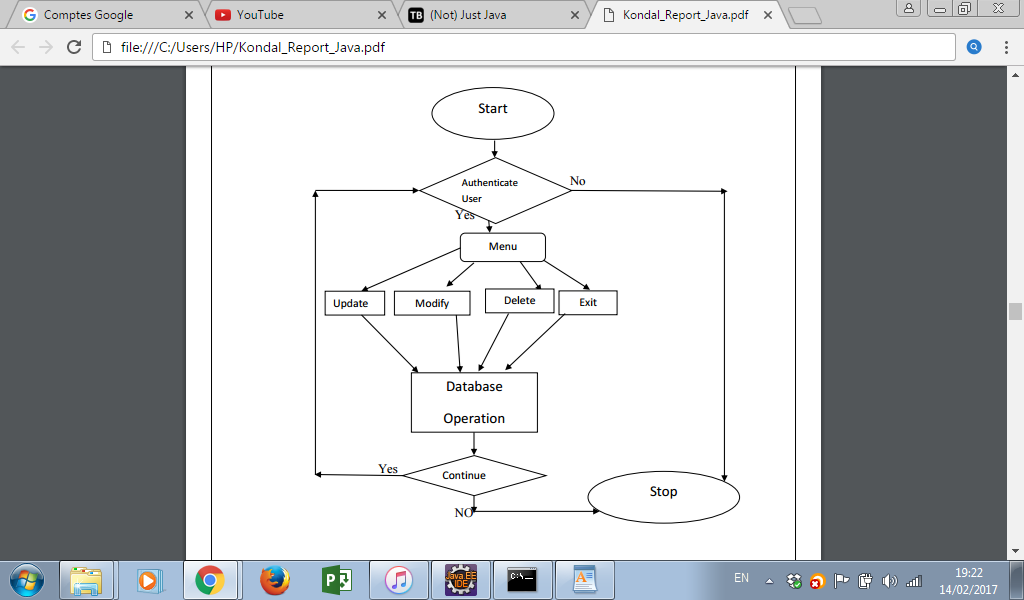
Searching is done with the identity email

**ii). Data Structures**

The data structures provided by the Java utility packages are very powerful and perform a wide range of functions. These data structures consist of the following interface and classes:

* Enumeration
* BitSet
* Vector
* Stack
* Dictionary
* Hashtable
* Properties
* XML

**iii). Global Application flow**



**4. CONSOLE OPERATION DESCRIPTIONS**

Implemented Console Operations in this System are

1. Authenticate user

2. Create an Identity

3. Update an existing Identity

4. Delete an Identity

Each Operations are explained below

* **Authenicate User**

User authentication is done by an authenticate method , that takes input as username and password, and calls a connection method which connects to the database and authenticates the user if provided credentials are correct, else will not authenticate the user and system stops.

* **Create an Identity**

This console operations allows an authenticated user to create a new identity. Identity uid, displayName, email and birthdate are provided by the user.The method used for creating a new identity is createIdentity.

* **Update an Identity**

This console operation allows a user to update an existing identity. To update an identity the user must provide the uid of an identity, which is to be updated.The method used is updateIdentity.

* **Delete an Identity**

This console operation allows a user to delete an existing identity and the user is supposed to provide the uid of the identity which is to be deleted. Method used is deleteIdentity.

* Search an Identity

This console operation allows a user to search an identity and then access to its information.

**7. BIBLIOGRAPHY**

<http://thomas-broussard.fr/>

<http://www.tutorialspoint.com/java/java_data_structures.htm>

<http://www.javabeginner.com/>

<http://www.tutorialspoint.com/>