**TECHNICAL SPECIFICATIONS**

**IAM APPLICATION**

**Submitted by**

**DEEPAK VIJAYAKUMAR**

**IAM APPLICATION**

**Table of Contents**

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

**2 Subject Analysis.....................................................................................**

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

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

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

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

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

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

3.1 Data structures........................................................................

3.2 Global application flow........................................................

3.3 Global schema and major features schema................

**4 Console operations descriptions.................................................................**

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

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

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

**1. Subject description:**

This Project Identity Access Management system is done to develop an application to manage the information of users in a database. The term **“Admin”** is used as “the person who will manage the information” and the term **“Identity”** is used to denote the “information of user in the system”.

**2. Subject Analysis:**

**2.1 Major features:**

1. It is a Java application where admin can perform below functions related to his/her information:

* Create an Identity
* Update an Identity
* Delete an Identity
* List all identities

All the data of the identities are stored in a database with table name “IDENTITIES”

**2.2 Application Feasibility:**

The development of this application with the given features is quite feasible and hence, the application has been developed.

**2.3 Data description:**

Below are the types of data created to make this application achieve its given features:

* **Class Identity:**

**Variables:**

* Private string uid;
* private String email;
* private String displayName;
* private string phoneNumber;

**Methods:**

* Identity()
* string getUid()
* void setUid(string)
* String getEmail()
* void setEmail(String )
* String getDisplayName()
* void setDisplayName(String )
* string getphoneNumber()
* void setPhoneNumber(string)
* String toString()
* **Class JDBCIentityDAO**

**Methods:**

* + public void writeIdentity(Identity)
  + public List<Identity> readAll();
  + **public** **boolean** updateIdentity(Identity)
  + **public** **boolean** deleteIdentity(Identity)

**public** List<Identity> findIdentity (String)

* **Class AdminDAO**
* **public** **boolean** authenticate(String, String)
* **Class ConsoleLauncher** 
  + Private void createIdentity(Scanner)
  + Private void listIdentities();
  + **Private void** updateIdentity(Scanner)
  + **Private void** deleteIdentity(scanner)

.

**2.4 Expected results:**

The application is supposed to perform following four functionsCreate an Identity

* Create an identity
* Update an Identity
* Delete an Identity
* List all identities

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

* **Limits**
* No GUI is provided
* Update and delete are performed on the assumption that display name is unique
* Less security features
* **Evolutions:**
* Application can be enhanced as a web application or desktop application.
* Multiple Identities can be deleted or updated at a time.
* Further security measures can be implemented

**3. Conception:**

**3.1 Data Structures**

Application mainly using Java List for storing or accessing Identity or configuration datas. Also application handling identity data through Identity Class. Application mainly using following patterns.

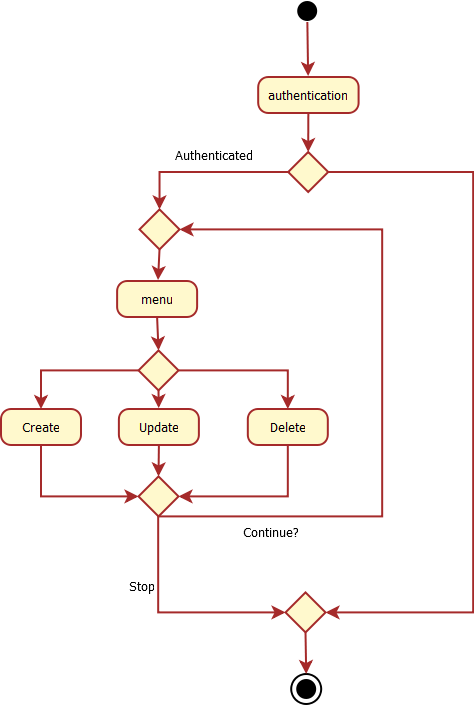
* Single pattern:

Used mainly for handling configuration details and Identity management.

* Data Access Object patterns

Patterns will choose how to handle the identity data. Currently it handles sql data.

**3.2Global application flow**



**4. Console Operations Description:**

User can perform the following functionalities:

1. **Create an Identity**

Admin enters the display name, email and phone number of the user. It is added to the database

1. **Update an Identity**

Display name is entered along with the new details. If identity exists , it is modified else displays message

1. **Delete an identity**

Display name is entered. If identity exists , it is deleted else displays message

1. **List all identities**

List all the identity details

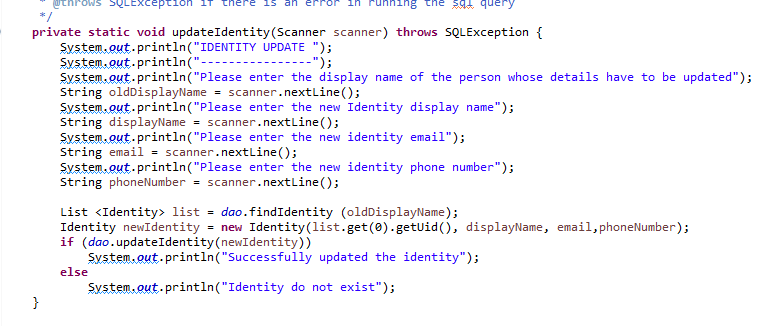
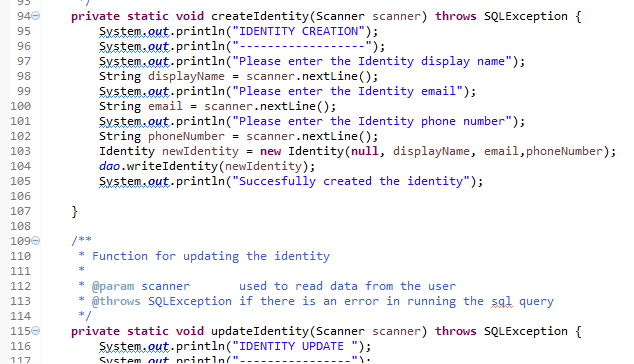
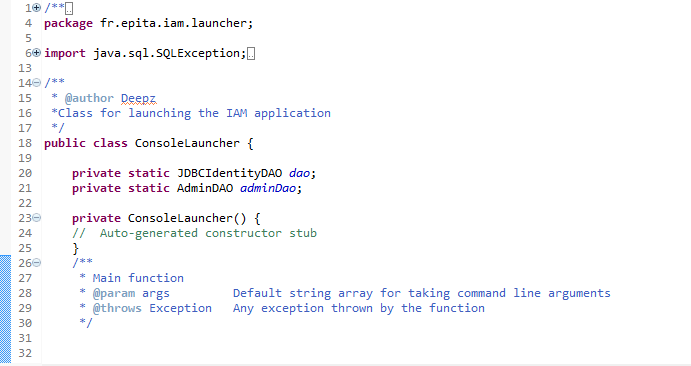
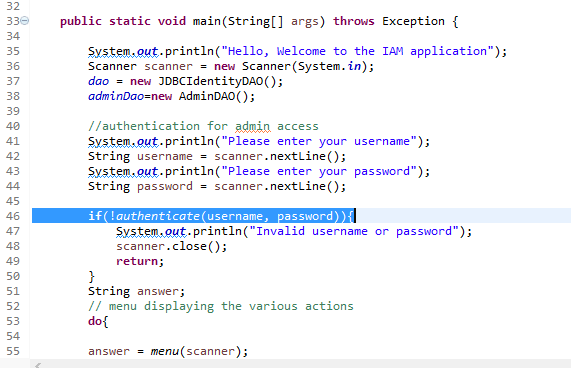
1. **Close**

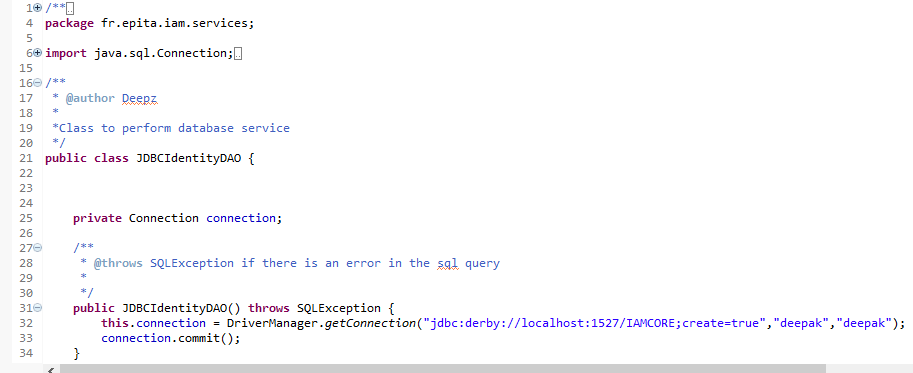
Exits the application

**5 Configuration instructions**

1. Install latest Java run time environment.
2. 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>

**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/>