IDENTITY MANAGEMENT SOFTWARE

Project Guided by: **Thomas Broussard**

Project Submitted by: **Ravina PATEL**

Fundamental Semester

M.Sc. Information System Management

Contents

1. Abstract

The Identity Management Software project is implemented in orders to manage the user identities. This is achieved using Java. It works with 1 kind of login - an admin account who has special privileges.

The project provides us a simple interface for maintenance of customer or student information. It can be used by schools or banks to maintain the records. The project is very useful for those who want to know about Identity Management System. The project provides facilities like registration and profile creation thus reducing paperwork and automating the record generation.

The environment used for development is Eclipse IDE. The database used is Apache Derby.

2. PROCESSING ENVIRONMENT-HARDWARE AND SOFTWARE

Operating system is a platform on which the specified application will be used. Once it has been complete, the software we are about to make, will execute on both client machine and server machine.

2.1 Configuration Requirements:

- ➢ Users Views: User

- ➢ Platform Operating Systems: mac OS Sierra X

- ➢ Technologies Used: Front End - Java

    Back end: Apache Derby

- ➢ Software Requirements: Java SE 8

- ➢ Tool Used: Eclipse Neon

3. Subject Analysis

The major features of this application are:

> ➢ Identity creation
>
> ➢ Identity access
>
> ➢ Identity modification
>
> ➢ User authentication
>
> ➢ Data access restriction

3.1 Application feasibility

The Identity management software application is built up on a well-defined and feasible structure. The application maintains a separate project for Core. Each project is further classified into modules. Some of the important classifications/features are as follows-

> ➢ **Src -** It contains separate folders for maintaining configuration properties, a datamodel for identity class, JDBCIdentityDAO connection and a services folder which contains the implementation package for equating and comparing the identity objects.

Src

- This folder includes classes for Search, view, update, delete identity.
  - ConsoleLauncher.java

- Authentication class
  - Autenticate.java

- It also contains an 'exceptions' package which maintains for exceptions different DAO operation.
  - MyException.java

- Database Connection Folder
  - JDBCIdentityDAO.java

Method used in console launcher class:
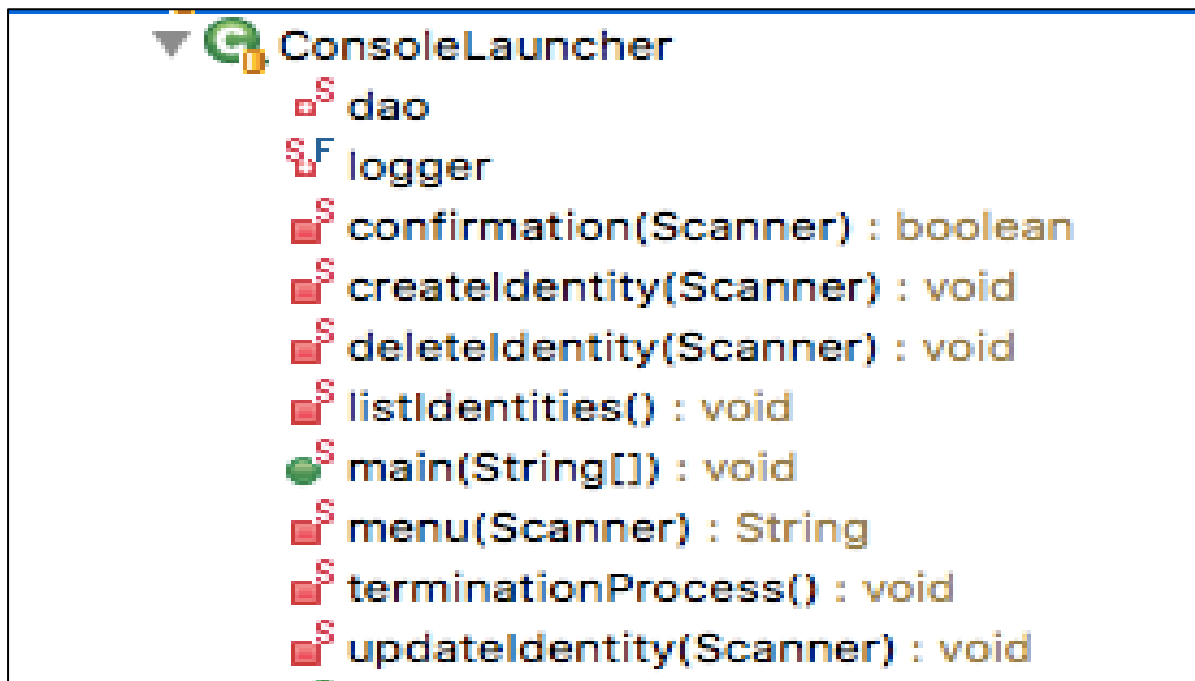
3.2 Using "logger.info"

When implementing a software, it is very useful to log information. We have implemented *"info.logging"* from Apache to implement loggings for this project

```
ConsoleLauncher.java ⊠
iam-core-ref_Ravina_PATEL/src/fr/epita/iam/
launcher/ConsoleLauncher.java
  4  package fr.epita.iam.launcher;
  5  import java.io.IOException;
 14
 15  /**
 16   * @author RAVINA_PATEL
 17   *
 18   */
 19
 20  public class ConsoleLauncher {
 21
 22
 23      private static final Logger logger =  LogManager.getLogger(ConsoleLauncher.class);
 24      private static JDBCIdentityDAO dao;
 25      private ConsoleLauncher()
 26      {
 27      }
 28      /**
 29       * @param args
```

The xml file contains logger to print on console as shown in figure below
    ✦ log4j2.xml

```
log4j2.xml ⊠
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <Configuration status="WARN">
 3     <Appenders>
 4         <Console name="Console" target="SYSTEM_OUT">
 5             <PatternLayout pattern="%msg%n" />
 6         </Console>
 7         <File name="MyFile" fileName="all.log" immediateFlush="false" append="false">
 8             <PatternLayout pattern="%msg%n"/>
 9         </File>
10     </Appenders>
11     <Loggers>
12         <Root level="info">
13             <AppenderRef ref="Console" />
14             <AppenderRef ref="MyFile"/>
15         </Root>
16     </Loggers>
17 </Configuration>
```

Jar files have been added for logger.

A jar file commons-logging-1.2.jar is included in the lib folder as shown in figure below:

4. Data description

The project deals with managing the identities of Information system which allows the users to create their user accounts by entering their display name (user name) and email-id. The admin of this software has special rights of viewing all identities, updating and deleting identities.

There is only one type of user for the system:

Registered User

❖ The registered user is given the rights to view his/her profile as well as update the profile.

❖ The user can update all attributes except Unique Id and name.

❖ The unique ids are auto-generated and available once the user is registered.

❖ This maintains the uniqueness of the identities.

❖ Only Registered User can enter Username and password to view and update profile.

4.1 Data Modification Options for the users:

The Identity Management Software provides following functionalities to the user:

i. View Profile

The registered user can view his/her profile at any point of time. The only requirement for this, is that the user should be a registered user. If the user is not registered, he has to registers using the form that is made available and then he is registered.

ii. Update profile

If at any point of time, the user wants to change the contact details like email address or contact no then he can update this information but not the unique id. Once the changes are made, they are saved into the database against that unique id for further access.

4.2 Data Access Option for USER

The Identity Management Software provides the following functionalities to user:

4.2.1 View All Profiles

The user will be able to see all the identities that are registered and present in the database.

4.2.2 Search identities

The user can search for a particular user by entering his *uid*

4.2.3 Delete Profile

The user will have the right to delete the identity. For deletion, the user will use the 'Delete identity' option.

5 Database

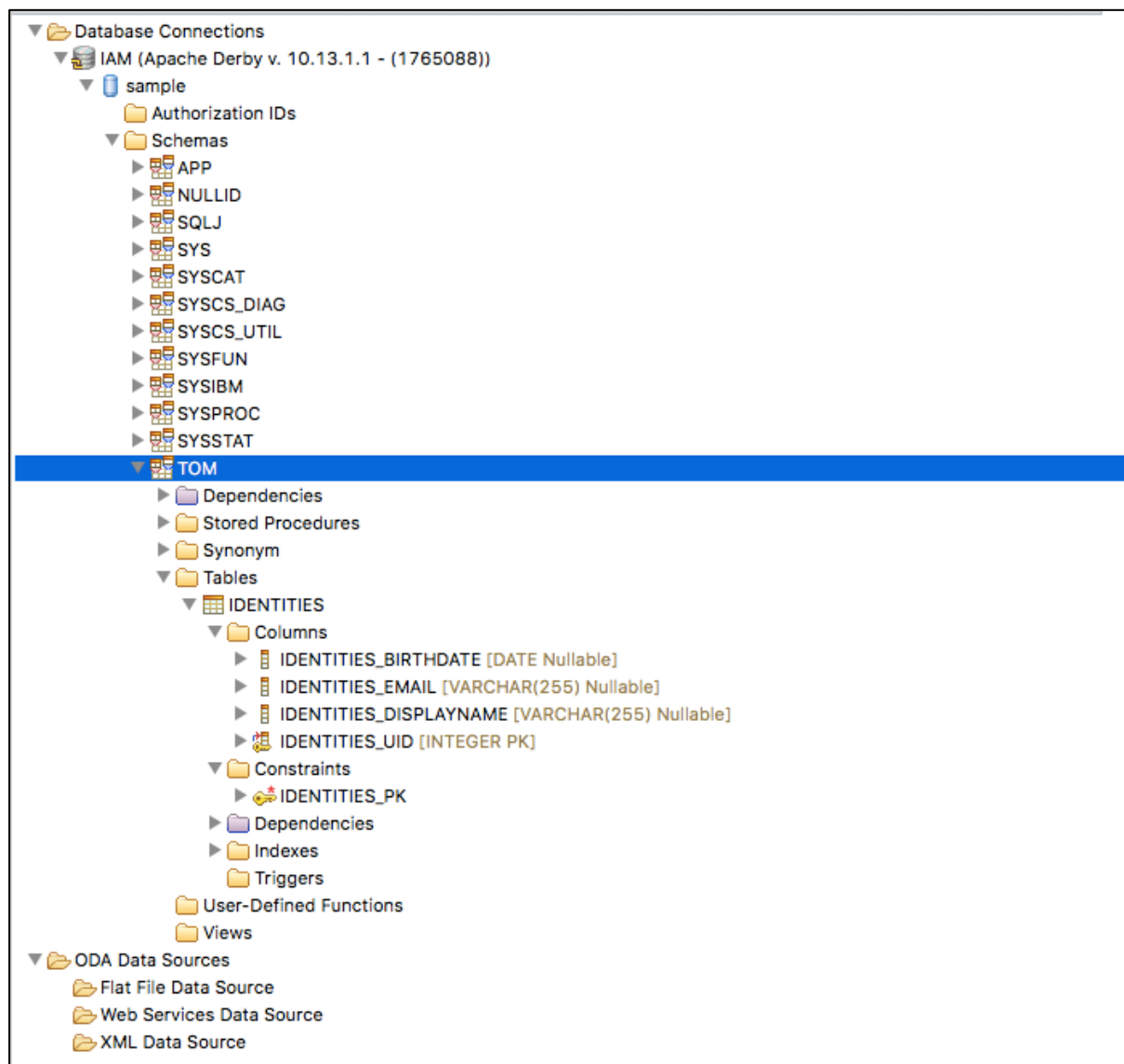The database used is *Apache Derby*

5.1 Connection Profile :

Version : Derby_10.x

Database Name: Sample

Schema Name: TOM

Table Name: TOM.IDENTITIES

5.2 Sql Query for Create Table

CREATE TABLE IDENTITIES (IDENTITIES_UID INT NOT NULL GENERATED ALWAYS AS IDENTITY CONSTRAINT IDENTITIES_PK PRIMARY KEY, IDENTITIES_DISPLAYNAME VARCHAR(255), IDENTITIES_EMAIL VARCHAR(255), IDENTITIES_BIRTHDATE DATE);
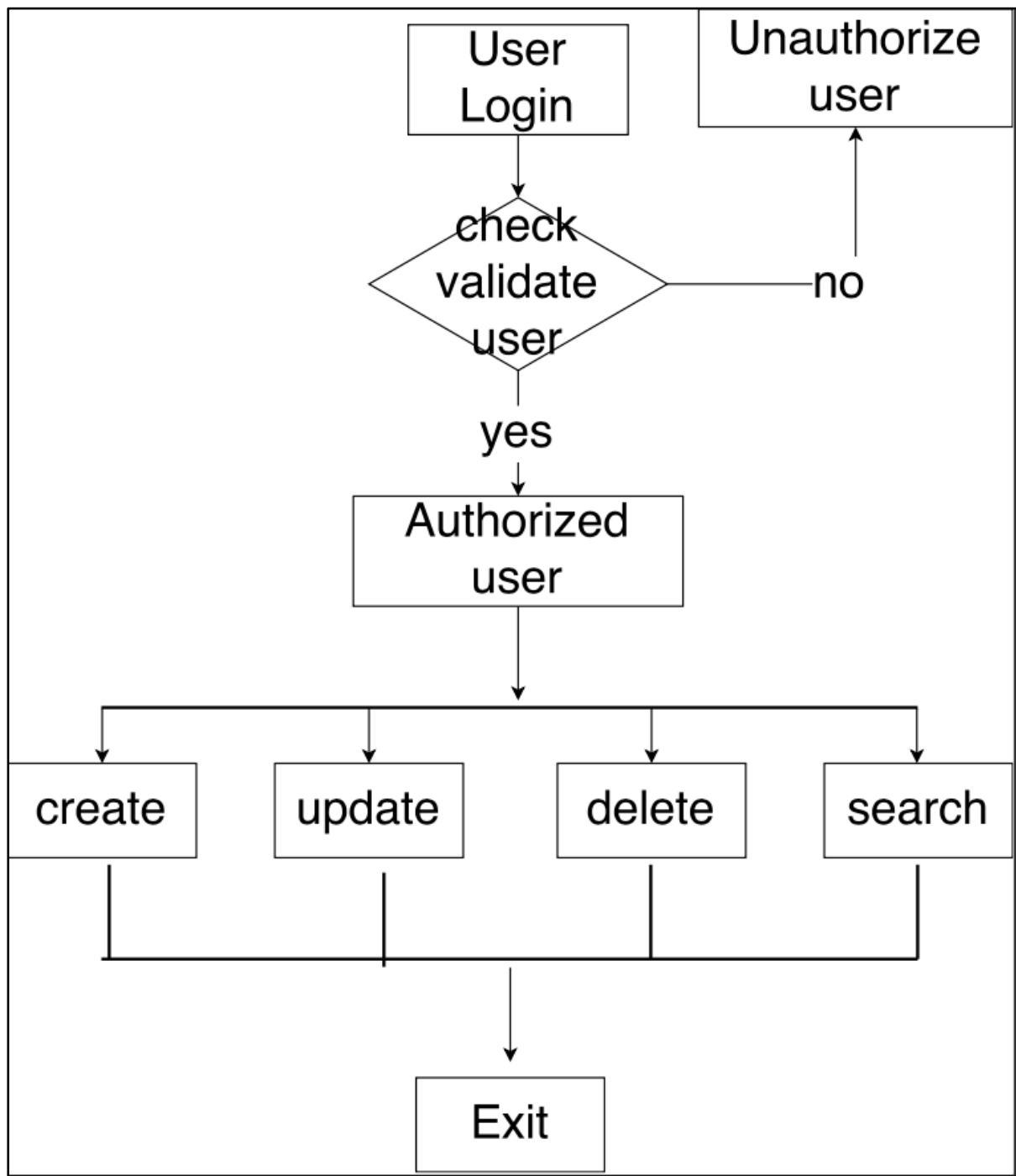
5.3 Data Manipulation Language (DML) part of SQL: -

- SELECT - extracts data from a database table
- UPDATE - updates data in a database table
- DELETE - deletes data from a database table
- INSERT INTO - inserts new data into a database table

6 Use case diagram
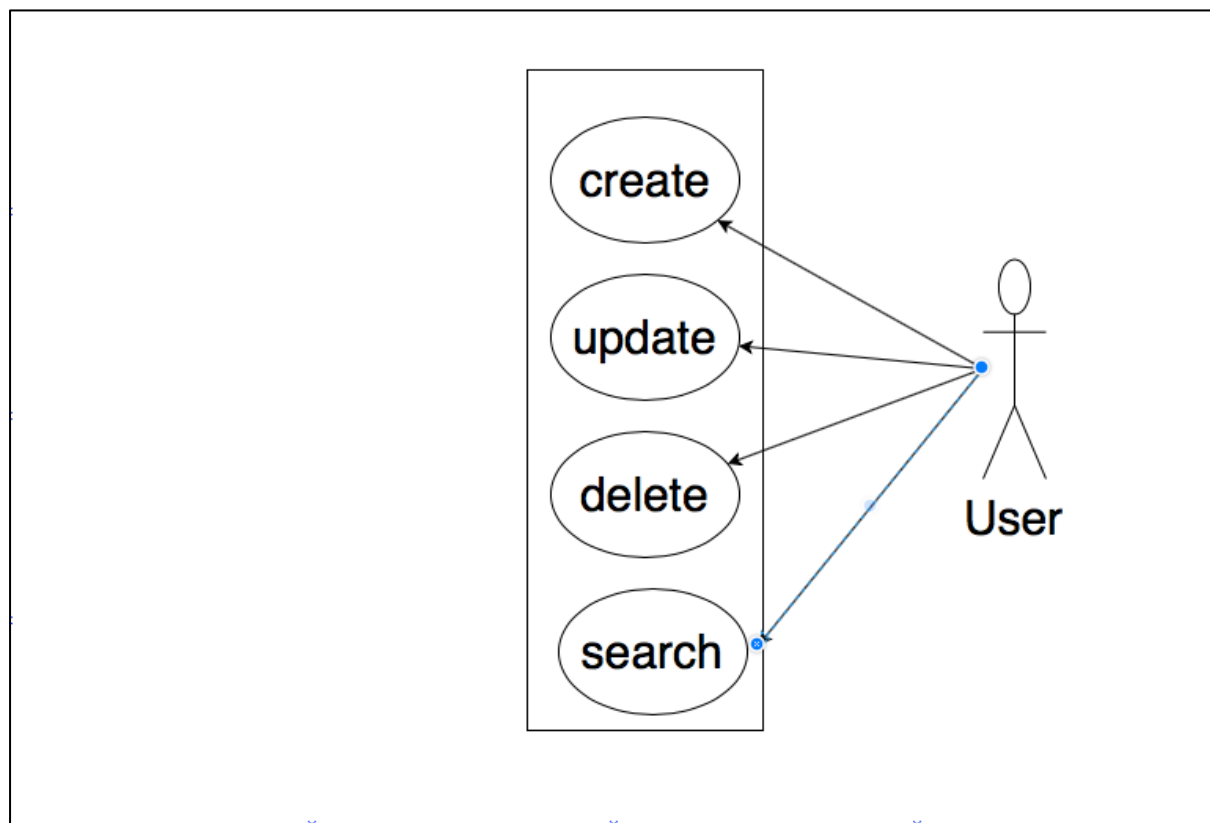
Users can create, update, delete and search the identities.
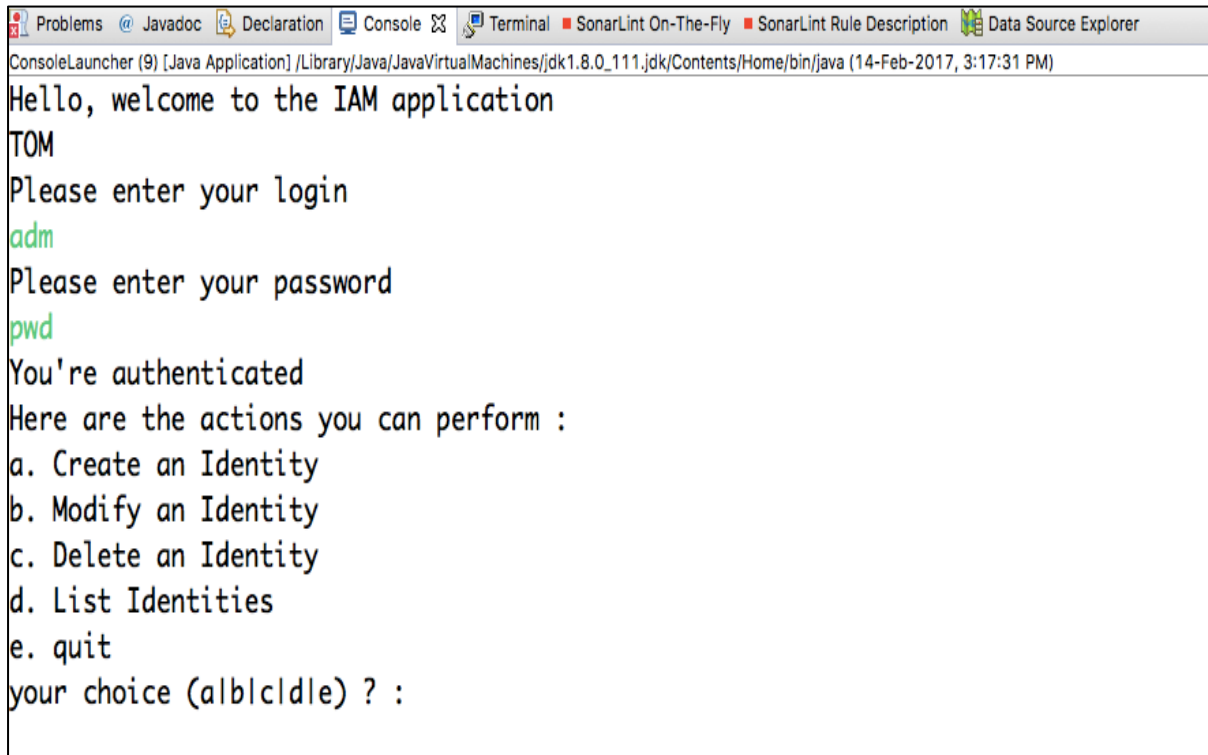
7 Data Flow Diagram:

8 SNAPSHOTS

8.1 LOGIN

Login: adm

Password: pwd

8.2 CREATE IDENTITY

Display Name: Rita

Email: rita@gmail.com


```
Problems  @ Javadoc  Declaration  Console ⊠  Terminal  ■ SonarLint On-The-Fly  ■ SonarLint Rule Description  Data Source Explorer
ConsoleLauncher (9) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_111.jdk/Contents/Home/bin/java (14-Feb-2017, 3:27:08 PM)
Hello, welcome to the IAM application
TOM
Please enter your login
adm
Please enter your password
pwd
You're authenticated
Here are the actions you can perform :
a. Create an Identity
b. Modify an Identity
c. Delete an Identity
d. List Identities
e. quit
your choice (a|b|c|d|e) ? :
a
You've selected : Identity Creation
Please enter the Identity display name
rita
Please enter the Identity email
rita@gmail.com
you succesfully created this identity :Identity [uid=null, displayName=rita, email=rita@gmail.com]
Do you want to continue Y/N
```

8.3 LIST OF IDENTITIES:

```
Problems   @ Javadoc   Declaration   Console ⊠   Terminal   ■ SonarLint On-The-Fly   ■ SonarLint Rule Description   Data Source Explorer
ConsoleLauncher (9) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_111.jdk/Contents/Home/bin/java (14-Feb-2017, 3:36:13 PM)
Hello, welcome to the IAM application
TOM
Please enter your login
adm
Please enter your password
pwd
You're authenticated
Here are the actions you can perform :
a. Create an Identity
b. Modify an Identity
c. Delete an Identity
d. List Identities
e. quit
your choice (a|b|c|d|e) ? :
d
This is the list of all identities in the system
TOM
0.Identity [uid=210, displayName=john, email=john@gmial.com]
1.Identity [uid=211, displayName=Jenika, email=jenika@gmail.com]
2.Identity [uid=212, displayName=rita, email=rita@gmail.com]
Do you want to continue Y/N
```
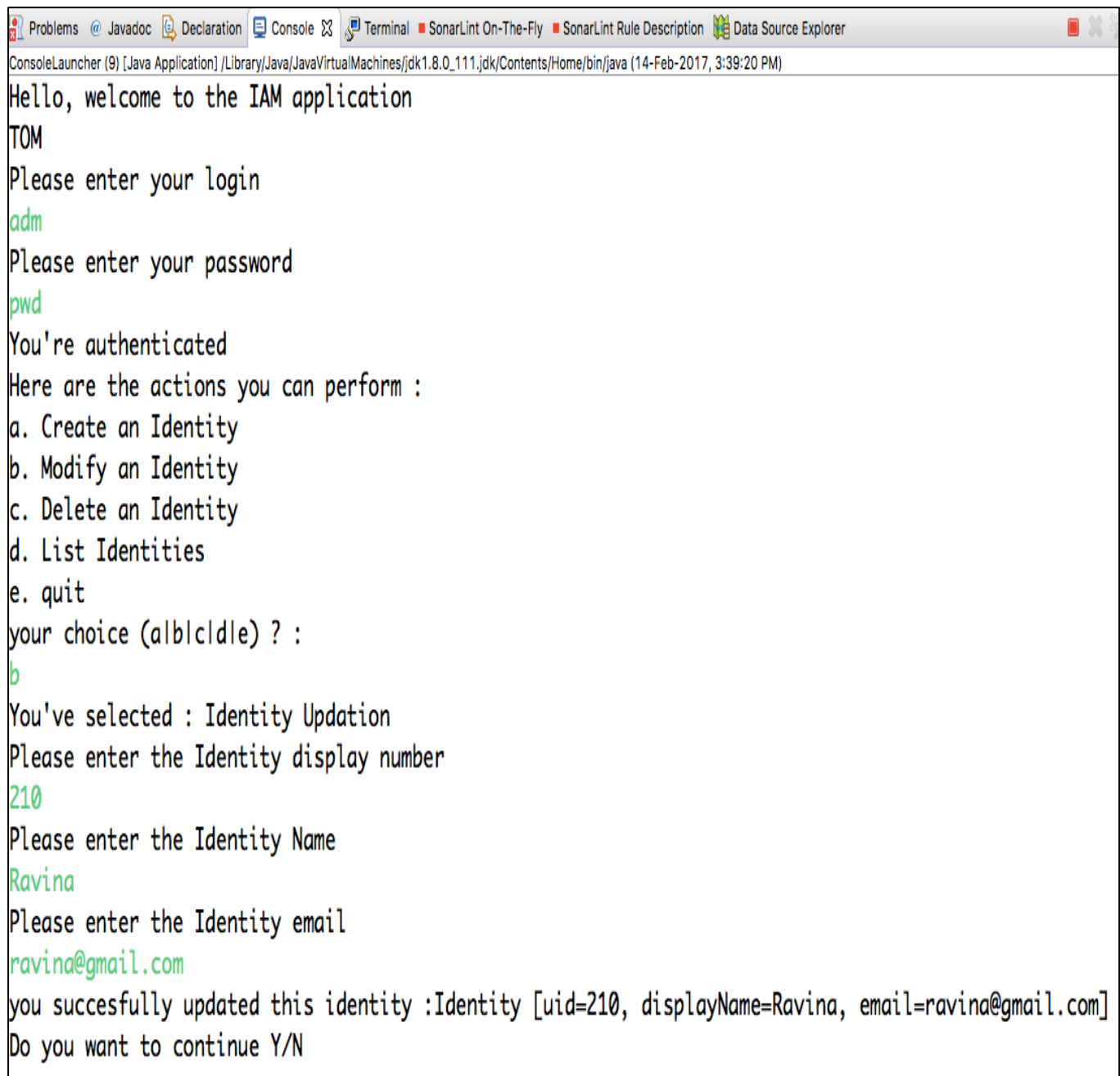
8.4 DELETE IDENTITY:

UID: 211

8.5 UPDATE IDENTITY:

UID: 210

```
Problems  @ Javadoc  Declaration  Console ⅹ  Terminal  ▪ SonarLint On-The-Fly  ▪ SonarLint Rule Description  Data Source Explorer
ConsoleLauncher (9) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_111.jdk/Contents/Home/bin/java (14-Feb-2017, 3:39:20 PM)
Hello, welcome to the IAM application
TOM
Please enter your login
adm
Please enter your password
pwd
You're authenticated
Here are the actions you can perform :
a. Create an Identity
b. Modify an Identity
c. Delete an Identity
d. List Identities
e. quit
your choice (a|b|c|d|e) ? :
b
You've selected : Identity Updation
Please enter the Identity display number
210
Please enter the Identity Name
Ravina
Please enter the Identity email
ravina@gmail.com
you succesfully updated this identity :Identity [uid=210, displayName=Ravina, email=ravina@gmail.com]
Do you want to continue Y/N
```

9   Scope of the application

This application is built to satisfy the very basic needs of Identity management. However, in order to make the identities closer to the real use, we can extend its scope. We can include alternate email-ids for a user, the user address and other details.

Evolutions:

The Identity Management System currently exhibits all DAO operations of search, create, update and delete using database (derby).

Bibliography

- https://github.com/thomasbroussard/2016s1_fundamental

- https://www.stackoverflow.com

- https://docs.oracle.com/javase/7/docs/api/