**JAVA PROJECT**

**EPITA 2018**

**1: Subject Description**

This identity management project focuses on managing identities and it is built to handle all the operations to manipulate the list of identities. This project is an console application with no graphical interface it facilitate reading and writing of character from console.

Possible operations to manipulate the Identity are the Create, Read, Update and Delete and to perform these operations user must be authorize to login to application to access, modify, delete or create new Identities .Hence the application is designed such that all

Operations are restricted only to the authorize users.

**2: Subject Analysis**

**Major features**

Main features of this application is to create identity, delete identity, update identity, and to search an identity and all these features are only available to privileges user

**Application Feasibility**

The application is feasible as it is easily possible to create a login system and then permit access to modify certain records in the database with the help of java. It can be use as API with other systems.

**Data Description**

Data being manipulated on identity object user has three basic data field on which all operation will be performed

Identity\_Displayname

Identity Email

Identity UId

**Expected Result**

Easily manipulate list of record in Database without any problem and can check the state of objects persisted in database.

**Algorithm Study**

I did not use any special algorithm for this application. This application is simple and can handle exceptions during the execution and authentication of user.

**Scope of the application**

Application scoping protects applications by identifying and restricting access to application files and data. Administrators can specify what parts of an application are accessible to other applications from application record and application table record. This application have a private scope. The evolution in this application can be done in the form of graphical interface and by improving login part

**3: Conception**

**Chosen Algorithm**

I did not use any complicated algorithm for this application the algorithm choose for this application is simple and basic. Application have a login controller which logs in the user and once the login in successful we have prepared a identity Service class which will house all the operations related to the identity. The service class in turn calls the dao class to interact with the db.

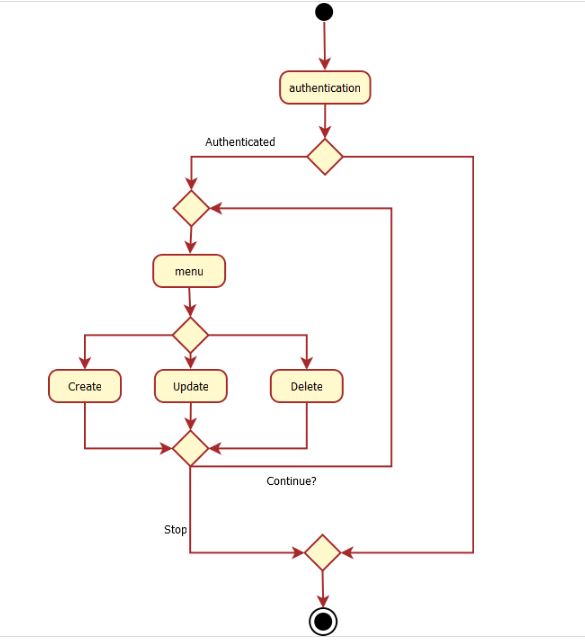
**Data structures**

The data structures used are List’s when the search operation is hit the list of Identity will be returned by the dao method via the service class.

**Global Application flow**

The first part in this application is authentication of user when user authenticate himself then he will access to the main functionality of application where he can perform following three main operation

Create , Update , Delete



**Global Schema**

The application has only one Table that is IDENTITIES

Below DDL indicate schema information

**CREATE** **TABLE** IDENTITIES

(IDENTITY\_ID **INT** **NOT** **NULL** GENERATED ALWAYS **AS** IDENTITY

**CONSTRAINT** IDENTITY\_PK **PRIMARY** **KEY**,

IDENTIY\_DISPLAYNAME **VARCHAR**(55),

IDENTITY\_EMAIL **VARCHAR**(55),

IDENTITY\_UID **VARCHAR**(55)

);

The identities table contain the id which is always generated and the display name and an email and the uid

**4: Console Operations Description:**

This console application has following operation

* Authentication

The user first authenticate himself to access the functionality. Application ask user to enter User Name and Password

* Menu

After successful login the main menu will appears to user then user can select his desired option. Menu contains five option

1. Create identity
2. Update identity
3. Delete identity
4. Search identiy
5. Quit

Application ask user to enter his selection and if user enter 1 application ask user to enter the details of identity and after giving details application create identity successfully

**5: Configuration Instruction**

You have setup Database for this application

Download apache derby client

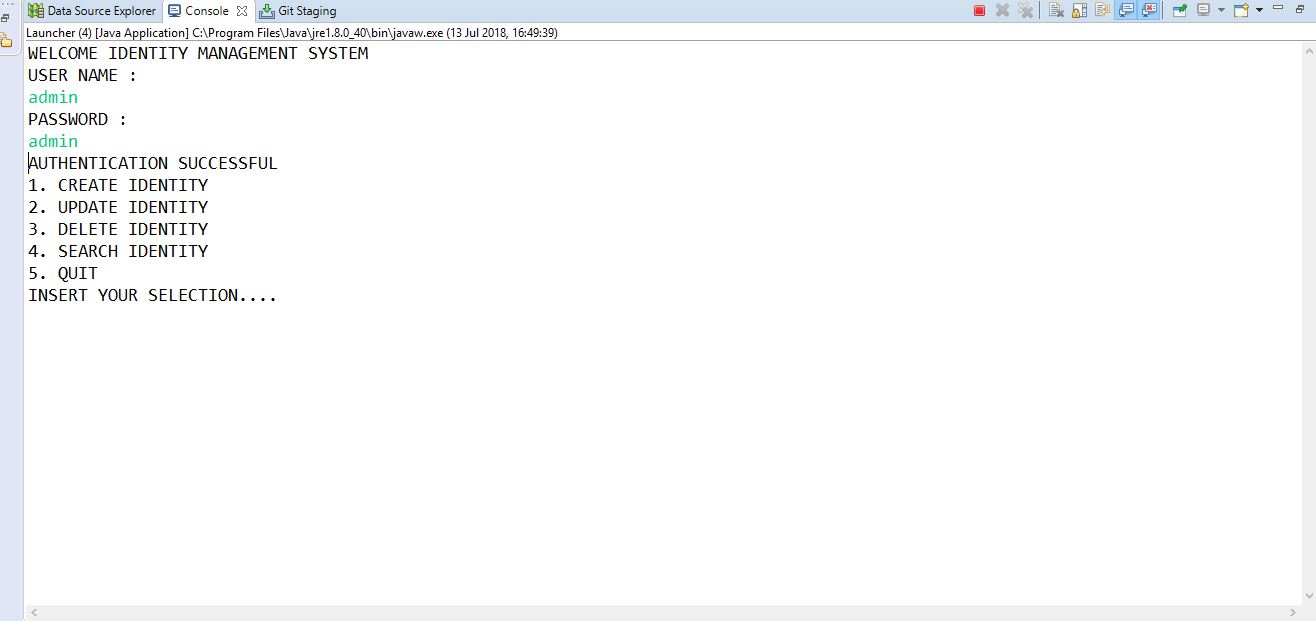
Create an instance of Derby name it as testInstance

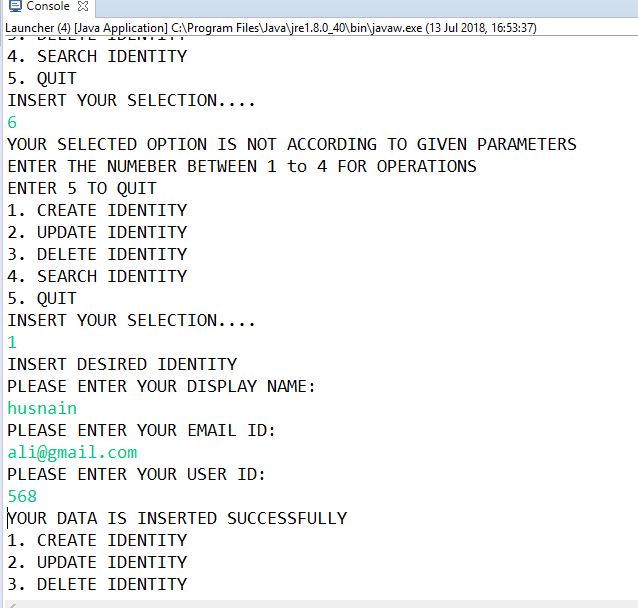
User name = “test”

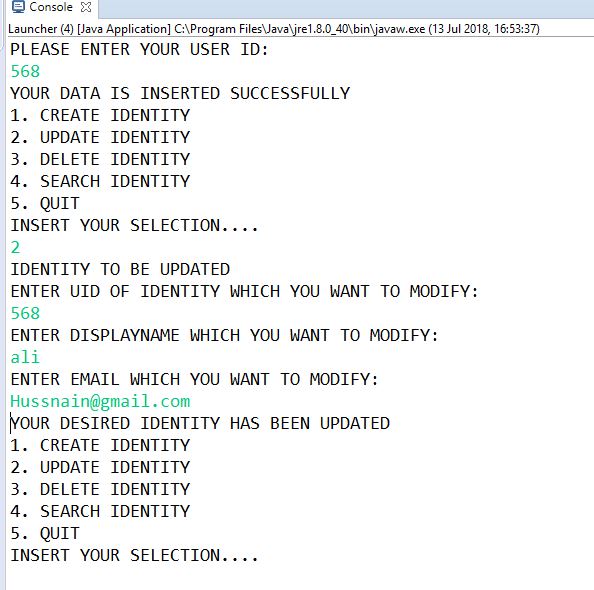
Password = “test”

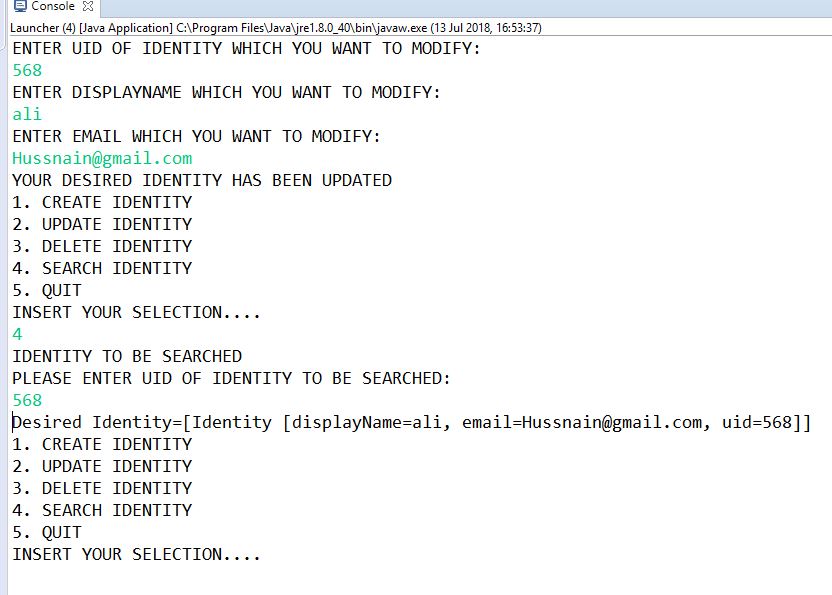
URL = "jdbc:derby://localhost:1527/testInstance;create=true";

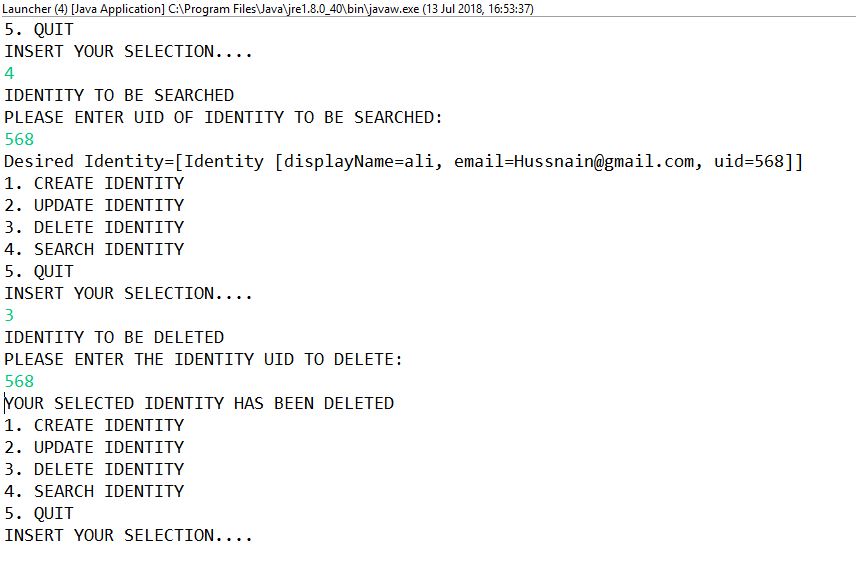
**6: Screen Shots**

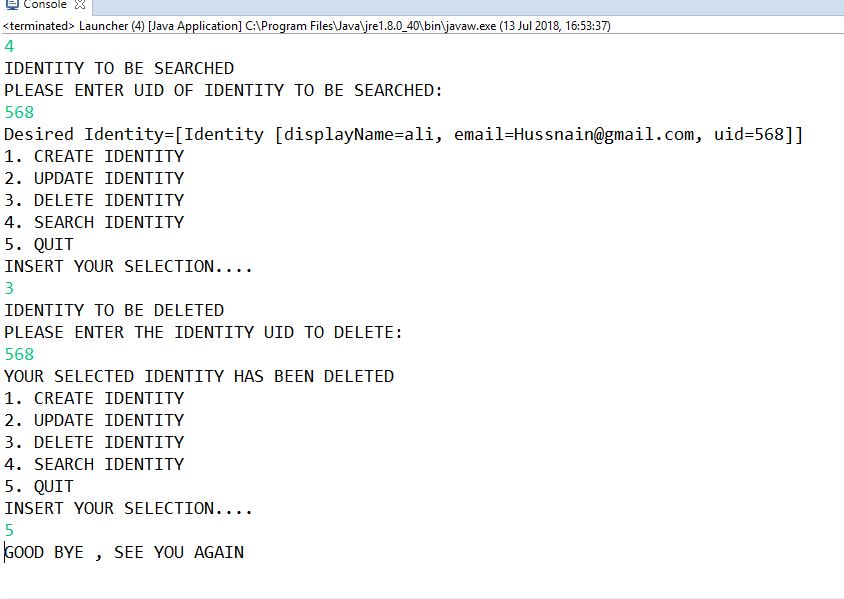
****

****

****

****

****

****

**7: Bibliography**

Just referred google and this website

<http://thomas-broussard.fr>

.