ÉCOLE D'INGÉNIEURS EN INFORMATIQUE

TECHNICAL SPECIFICATIONS

IAM APPLICATION

Submitted by

SREEDEVI BEENA

MSc COMPUTER SECURITY
FALL 2017

IAM APPLICATION

Table of Contents

1.SUBJECT DESCRIPTION.....3

2.SUBJECT ANALYSIS.....3

2.1 MAJOR FEATURES.....3

2.2 APPLICATION FEASIBILITY.....4

2.3 DATA DESCRIPTION.....4

2.4 EXPECTED RESULTS.....6

2.5 SCOPE OF THE APPLICATION (SCOPE,LIMITS, EVOLUTIONS).....6

3.CONCEPTION.....7

3.1 DATA STRUCTURES.....7

3.2 GLOBAL APPLICATION FLOW.....8

3.3 GLOBAL SCHEMA AND MAJOR FEATURES SCHEMA.....9.

4.CONSOLE OPERATIONS DESCRIPTIONS.....10

5.CONFIGURATION INSTRUCTIONS.....11

6.COMMENTED SCREENSHOTS.....11

7.BIBLIOGRAPHIES.....13

1. SUBJECT DESCRIPTION:

This Project **Identity Access Management** system is done to develop an application to manage/access the information of users in a database. This application renders a platform for the user to perform several functions on the data stored in the database and also allows retrieving the data from it. 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. This 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/**Search**

2. All the data of the identities are stored in a database with table name "**IDENTITIES**". The **GUI** makes the application user-friendlier and easily accessible. The **login details of the admin** are stored in the "**ADMIN"** table inside an **Embedded Derby Database** which is named as the **UserTable.** This database resides at:

IAMWindowApplicationBySreedeviBEENA\UserTable .

3. This contains a **CONFIGURATION** class and a **LOGGER**, which helps in implementing more security as well as to track the operations performed.

4. An **AUTHENTICATION** is performed to check whether it is the exact person using this application, which also enhances security.

5. Apart from this it is:

- Highly user-friendly
- Platform Independent
- Robust
- Good at Error detection
- Clean separation of various components
- Easy modification

2.2 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.
- Most of the components used such as the development platform, servers, and databases are open source.
- The use of **CONFIGURATION** and **LOGGER** makes the application more standard and outstanding from the available ones.
- The development of this application with the given features is quite feasible and hence, the application has been developed with a **GUI**, which made it user-friendlier.

2.3 DATA DESCRIPTION:

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

The name of the schema is ROOT.

The Schema for data is :

IDENTITIES<UID,IDENTITY_DISPLAYNAME,EMAIL>

UID	**:**	VARCHAR, Unique
IDENTITY_DISPLAYNAME	:	VARCHAR
EMAIL	:	VARCHAR

The structure of Admin table is :

ADMIN<USER_NAME, PASSWORD, PRIVILEGE >

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

- **Fr.epita.iam.LAUNCHER** : This package contains the main Class and inside there is the*main* method which drives the application.
- **Fr.epita.iam.DATAMODEL :** This package consists of the data model for the **Identity** class with the following attributes< **UID, IDENTITY_DISPLAYNAME, EMAIL>.**Also it contains the User class, which sets and gets the values for Login credentials. The attributes are **< Username, Password>.**
- **Fr.epita.iam.SERVICE :** This package contains the classes, which provides all the services including database connectivity. The classes are:
  - **AdminDAO :** This is the module, which does the database connectivity to the Admin table in the Embedded Derby database for the Admin authentication.
  - **IdentityDAO :** This module is an interface, which declares the four services. They are Create, Search, Update and Delete.
  - **IdentityJDBCDAO :** This module performs the JDBC connectivity to the IDENTITIES table.
  - **Configuration :** This class performs the configuration functionality by creating instance of Properties class and loading the properties from the *Configuration.properties* file inside our package. The url is given as *db.host*, username as *username* and password as *password*.
- **Fr.epita.iam.windowApp:** This package has two classes, which defines the two frames of the Window application**.** They are:
  - **Login &Menu :**These classes are inside the window package of the project which enables the GUI functionality for Login page and Menu page of the application.

2.4 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, display all identity, modifying an existing identity, and deleting an identity from the database.
- The database used is a DerbyClient database for the main functionality and for the Admin table it uses Embedded Derby Database.
- The tool needs to communicate with the database and return with the results in quick time in to a Graphical User Interface(Window).

2.5 SCOPE OF THE APPLICATION (SCOPE,LIMITS, EVOLUTION):

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.
- Centralization of user administration.
- Window based Graphical User Interface.

Limitations:

- Lack of multi user functionality.At present only the Admin can perform all the functions.
- Lack of added feature like in modern address book.

Evolution:

- We are working to inhance the IAM as a complete web-based software, users and login and keep details, export details, import to new system, also send SMS from there and save other information. Also to make it a Multi user application.

3. CONCEPTION:

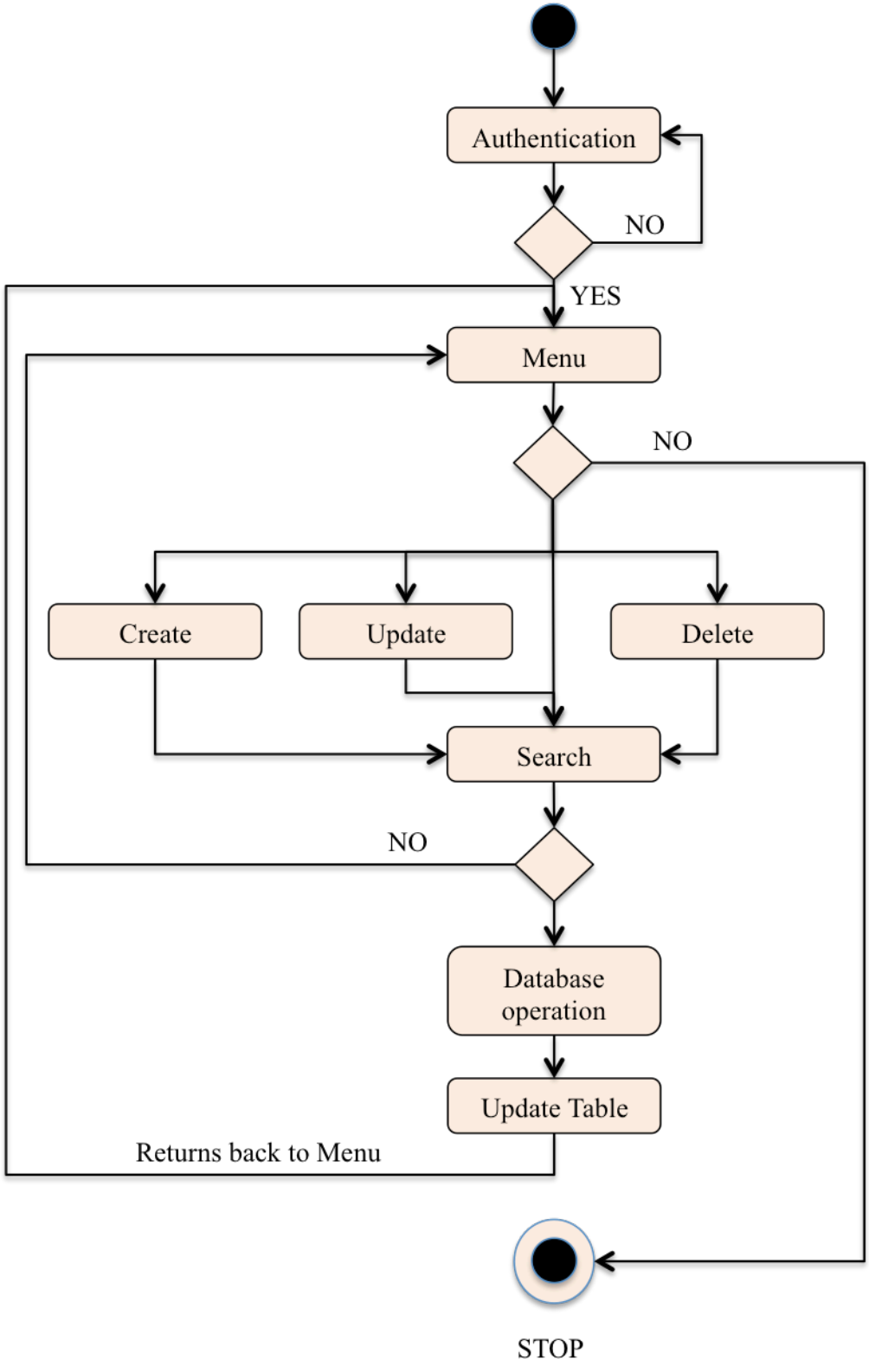
3.1 DATA STRUCTURES

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

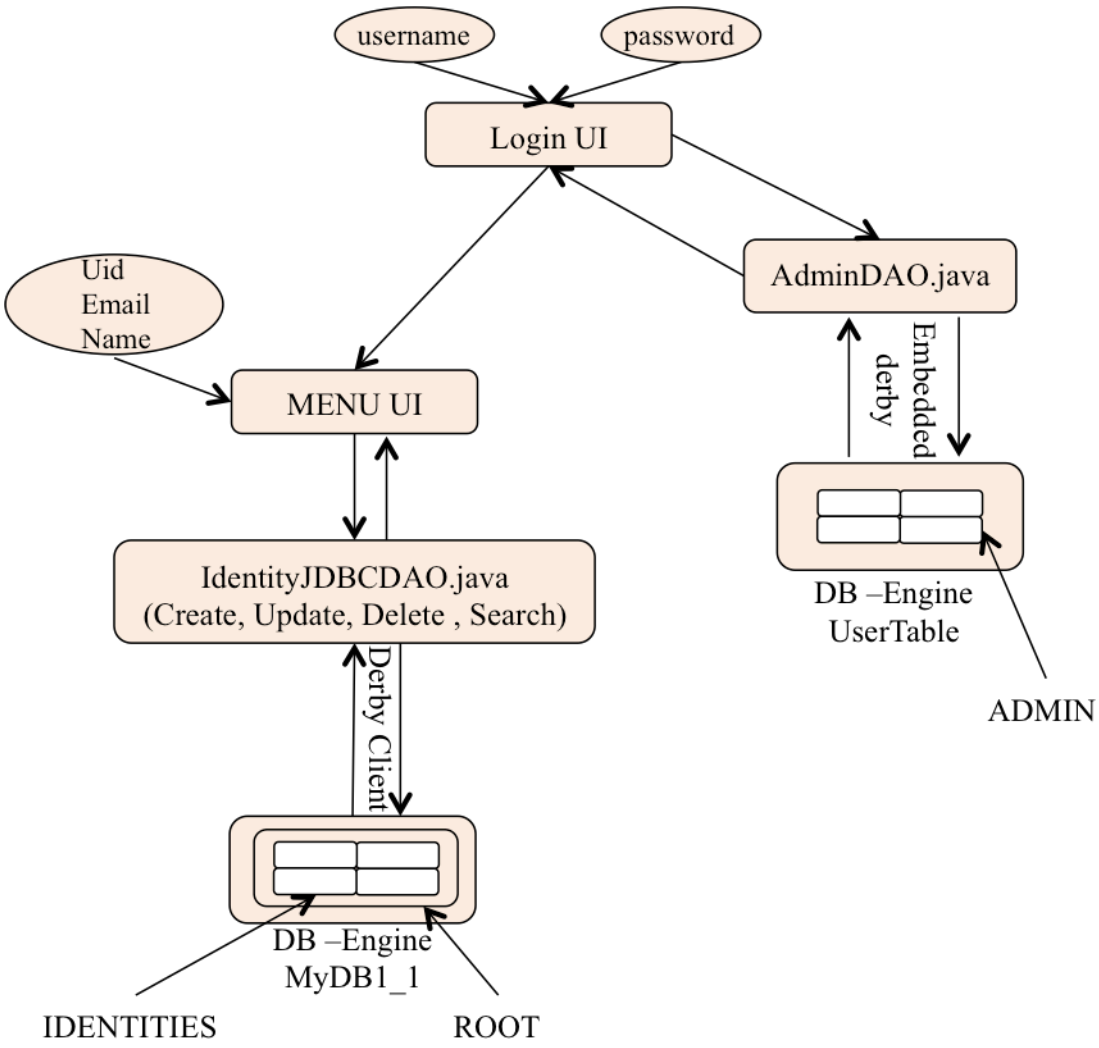
However the data structure used in this project is **LIST ,**which is a collection of elements .List is used in the project to store the collection of identities.

String is also a widely used data structure in this project.

3.2 GLOBAL APPLICATION FLOW



3.3 GLOBAL SCHEMA AND MAJOR FEATURES SCHEMA



4. CONSOLE OPERATIONS DESCRIPTION:

Console (in Window GUI) Operations Implemented in this Systemare:

1. Authentication of user
2. Create an Identity
3. Update an existing Identity
4. Delete an Identity
5. Search the record in the table and display.

Each Operation are explained below

- **Authenticate User**

  User authentication is done by an **authenticate** method in the AdminDAO that takes input as username and password, and connects to the database and authenticates the user if provided credentials are correct, else will not authenticate the user and window will pop up telling the error.

- **Create an Identity**

  This operation allows an authenticated user to create a new identity. Identity Uid, name and email are provided by the user.The method used for creating a new identity is **create** in the **IdentityJDBCDAO class.**

- **Update an Identity**

  This 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 **update.**

- **Delete an Identity**

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

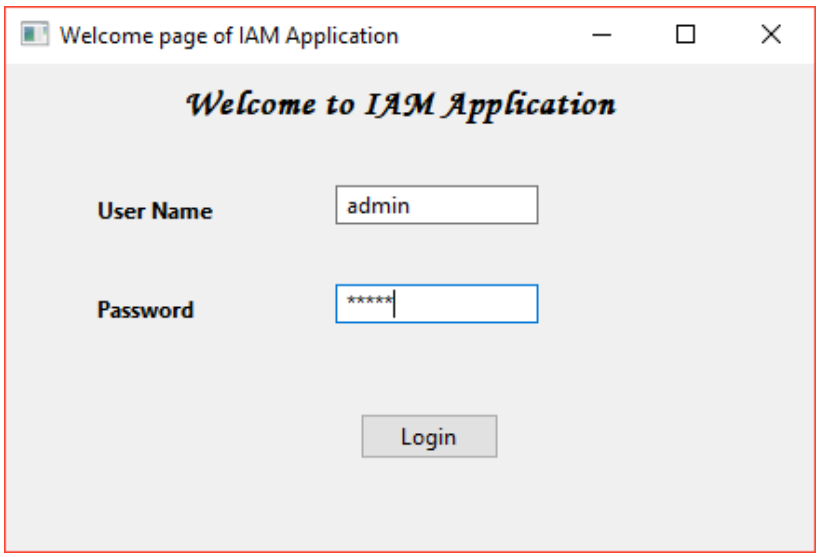
- **Search/List Identities and display**

  This operation retrieves the identities stored in the database and output them to the GUI. Methods used are **search and display.**

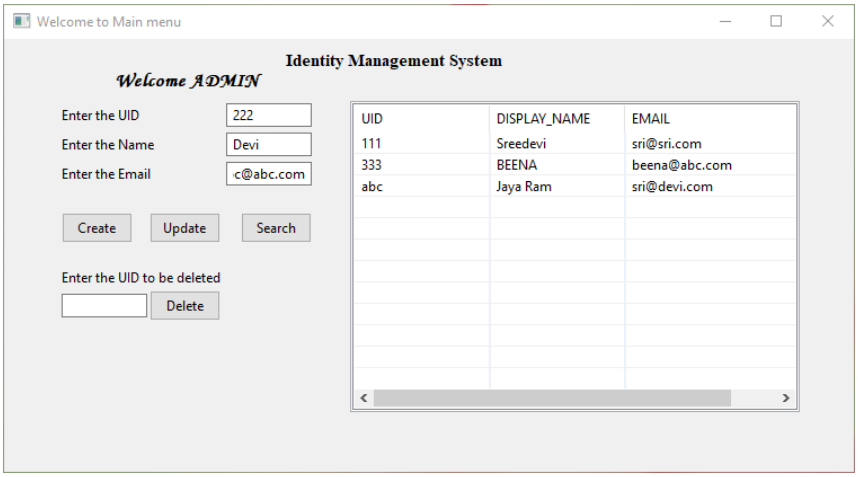
5. CONFIGURATION INSTRUCTIONS

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
3. There is an Embedded Derby database, which contains the login information of the Admin. The URL(**adminDb.host**), Username(**userDb**) and password(**passDb**) for connecting to this database is provided inside the *Configuration.properties* file. Please do not alter them.

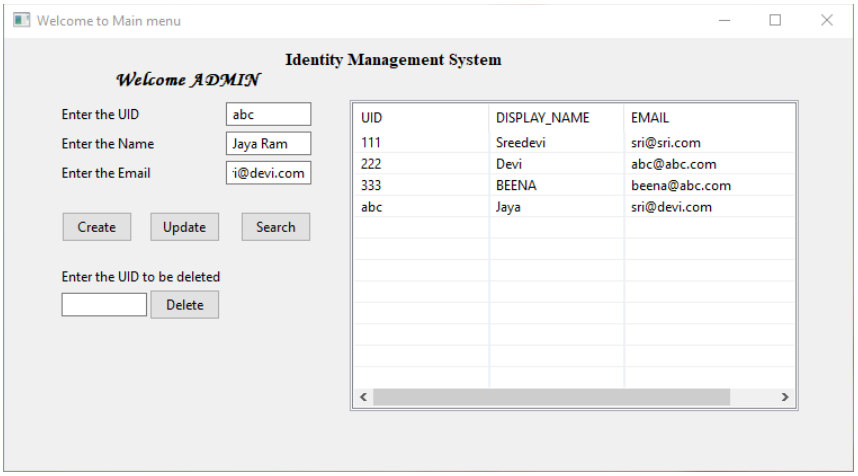
6. COMMENTED SCREENSHOTS



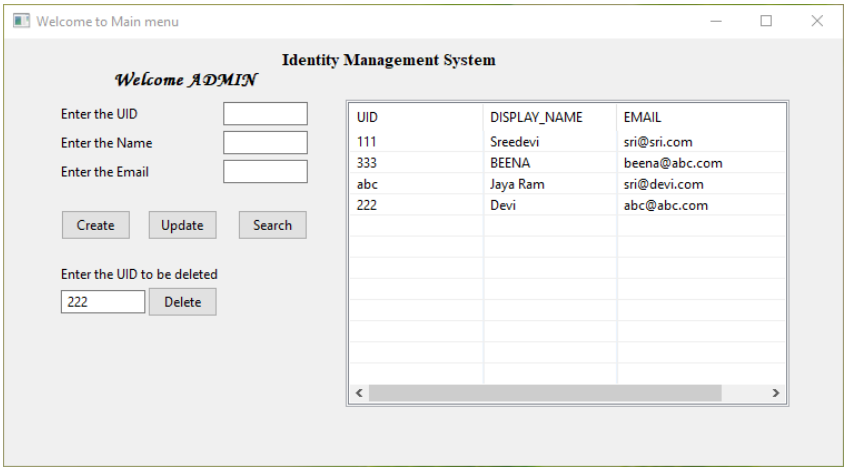
a. Screenshot of Login page of the Application



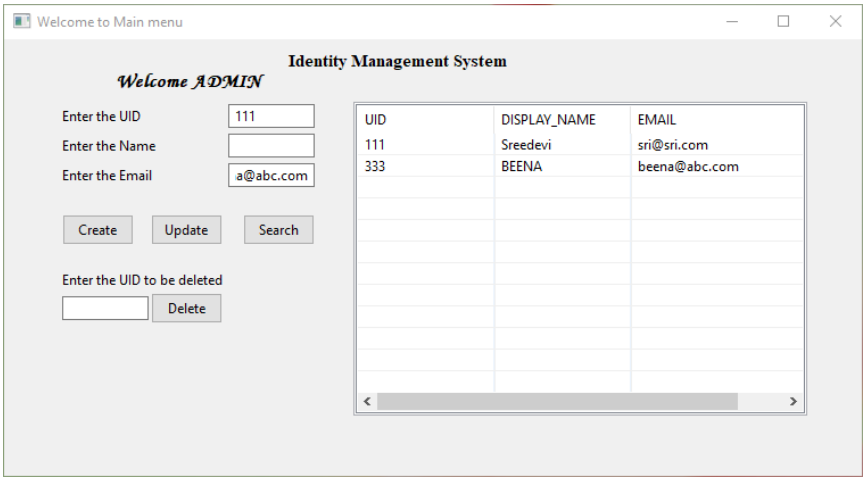
b. Screenshot of the Main Menu-User Creation



c. Screenshot of the Main Menu- User Updating



d. Screenshot of the Main Menu-User Deletion



e. Screenshot of the Main Menu-Search

7. BIBLIOGRAPHIES

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

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

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