

Java Fundamental Project

IAMCORE

Behan Remoshan BENET RUBAN

13th July, 2018

MSc Fundamentals - EPITA

Contents

Contents	1
Subject Description	3
Subject analysis	3
Major features	3
Authentication	3
Managing Data	3
Application Feasibility	3
Data description	3
Expected results	4
Algorithms study	4
Scope of the application (limits, evolutions)	4
Conception	5
Chosen algorithm	5
Data structures	5
Global application flow	6
Authentication	7
Select operation	7
Operation	7
Next operation Loop	7
Global schema and major features schema	7
Console operations description	8
readIdentityFromConsole	8
readCriteriaFromConsole	8
displayIdentitiesInConsole	8
readUsernameFromConsole	8
readPasswordFromConsole	8
menuSelectionFromConsole	8
Configuration instructions	9
Configuration details	9

—

Subject Description

The IAM project is an Identity Management Application to manage the identities and users dynamically.

Subject analysis

Major features

Authentication

Only authenticated users can access the application and manage the data.

Managing Data

- Create Identities
- Update Identities
- Search Identities
- Delete Identities

Application Feasibility

All major features of the application are easily achieved by using a database. Major operations CREATE, UPDATE, SEARCH and DELETE are easily executed using SQL queries and the errors are managed in a efficient manner with the help of Derby Server and JDBC connection.

Console is used as the user interface, however with lots of inline instructions users can perform the needed task without difficulties.

Data description

This application manages the user identity data as follows

- Name
- Email
- UID

Additional field ID will be used to identify each unique entries in the database

Expected results

Store and manipulate identities in the database flawlessly. Provide an authentication system to allow certain users to manipulate the data. Provide an interface through console to allow users to perform tasks.

Algorithms study

- Table is created in the database using Derby
- The database is connected to the java using JDBC
- The create, search, update and delete functions are designed to manage identities.
- The authentication procedure allows users to login through console
- Logger systems and exception classes are developed to improve the program standards.

Scope of the application (limits, evolutions)

The application management is restricted to the authenticated users. It manages both the identities and the users. This is helpful for the dynamic management.

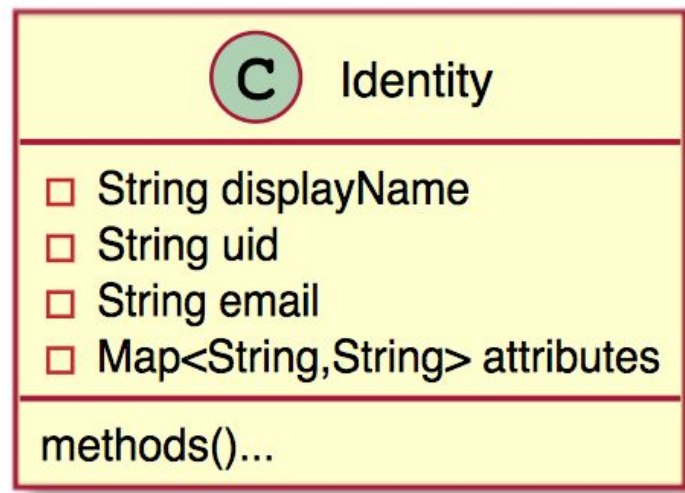
Conception

Chosen algorithm

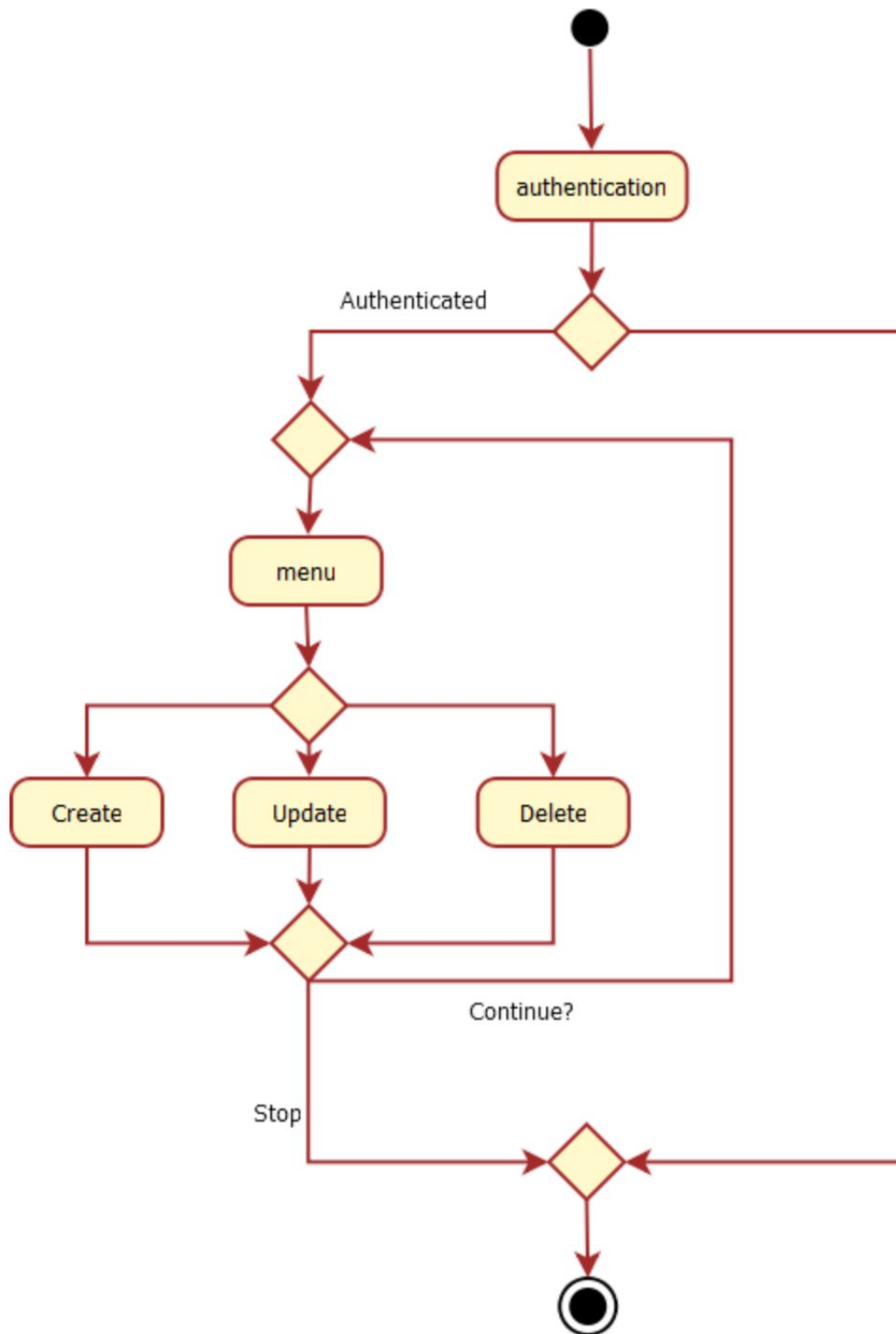
Data structures

Identities: The identities are the entities which is will be managed by the application. It has the field display name, email id and the uid, All the three field are String as per the requirement. The identity is implemented as java class which makes it more secure and manageable as the variables are private to the class.

Data can be accessed only through the implemented methods of getters and setters.



Global application flow



Authentication

As soon as the application is launched the console requests the user to enter the username followed by the password. If the user is authenticated, user gets access to the menu else the programs exits.

Select operation

Authenticated user enters to the selection menu where they can make a choice to perform specific operation.

Operation

The selected operation will be executed and the result of the operation will be displayed on the screen.

Next operation Loop

User can choose to perform another operation or exit the application.

Global schema and major features schema

The application runs with a console as the user interface. Derby powered database is used to store the identities. Database has one table that stores all identities.

```
CREATE TABLE TEST.IDENTITIES
(
  IDENTITY_ID INT NOT NULL GENERATED ALWAYS AS IDENTITY
  CONSTRAINT IDENTITY_PK PRIMARY KEY,
  IDENTITY_DISPLAYNAME VARCHAR(55),
  IDENTITY_EMAIL VARCHAR(55),
  IDENTITY_UID VARCHAR(55)
);
```




Console operations description

`readIdentityFromConsole`

Reads identity from console, asks user to enter name, followed by email and UID. Then it creates a object Identity and returns it.

`readCriteriaFromConsole`

Reads identity from console, asks user to enter name criteria, followed by email criteria. Then it creates a object Identity and returns it.

`displayIdentitiesInConsole`

Takes as a parameter list of identities and write each identity to the console.

`readUsernameFromConsole`

Asks the user for username and returns the input the application

`readPasswordFromConsole`

Asks the user for password and returns the input the application

`menuSelectionFromConsole`

Displays a menu with all available operation possible. Users input is taken and returned to backend application where the respective operation is performed.



Configuration instructions

Go to Window -> Show View -> Other -> Data Source Explorer

right click on Database Connections and click New.

From there, select Derby and create your database.

Configuration details

Database name: dbiam

User: test

password : root

Now, you have to import the derby library.

Right click on your project on the left, go to properties, Build Path, libraries, add external JAR and search in your Derby/lib folder for derbyclient.jar.

Start the derby database server

In the project explorer under sql directory run the create-schema-test.sql file to create the schema 'TEST'

Then run the file create-table.sql file to create the tables required.

Then Launch the application.

