JAVA (USER IDENTITY MANAGEMENT)

EPITA

mounika kasaraneni Vinay pavan guntur

2018

Table des matières

[1 Subject description 1](#_Toc507092459)

[2 Subject analysis 2](#_Toc507092460)

[2.1 Major features involved in the project are: 2](#_Toc507092461)

[Create 2](#_Toc507092462)

[Modify 4](#_Toc507092463)

[Delete 5](#_Toc507092464)

[List 5](#_Toc507092465)

[Exit 6](#_Toc507092466)

[2.2 Application Feasibility 6](#_Toc507092467)

[2.3 Data description 6](#_Toc507092468)

[2.4 Expected results 6](#_Toc507092469)

[2.5 Algorithms study(logic) 6](#_Toc507092470)

[2.6 Scope of the application (limits, evolutions) 6](#_Toc507092471)

[3 Conception 6](#_Toc507092472)

[3.1 Chosen algorithm 6](#_Toc507092473)

[3.2 Data structures 6](#_Toc507092474)

[3.3 Global application flow 7](#_Toc507092475)

[3.4 Global schema and major features schema 7](#_Toc507092476)

[4 Console operations description 7](#_Toc507092477)

[5 Configuration instructions 7](#_Toc507092478)

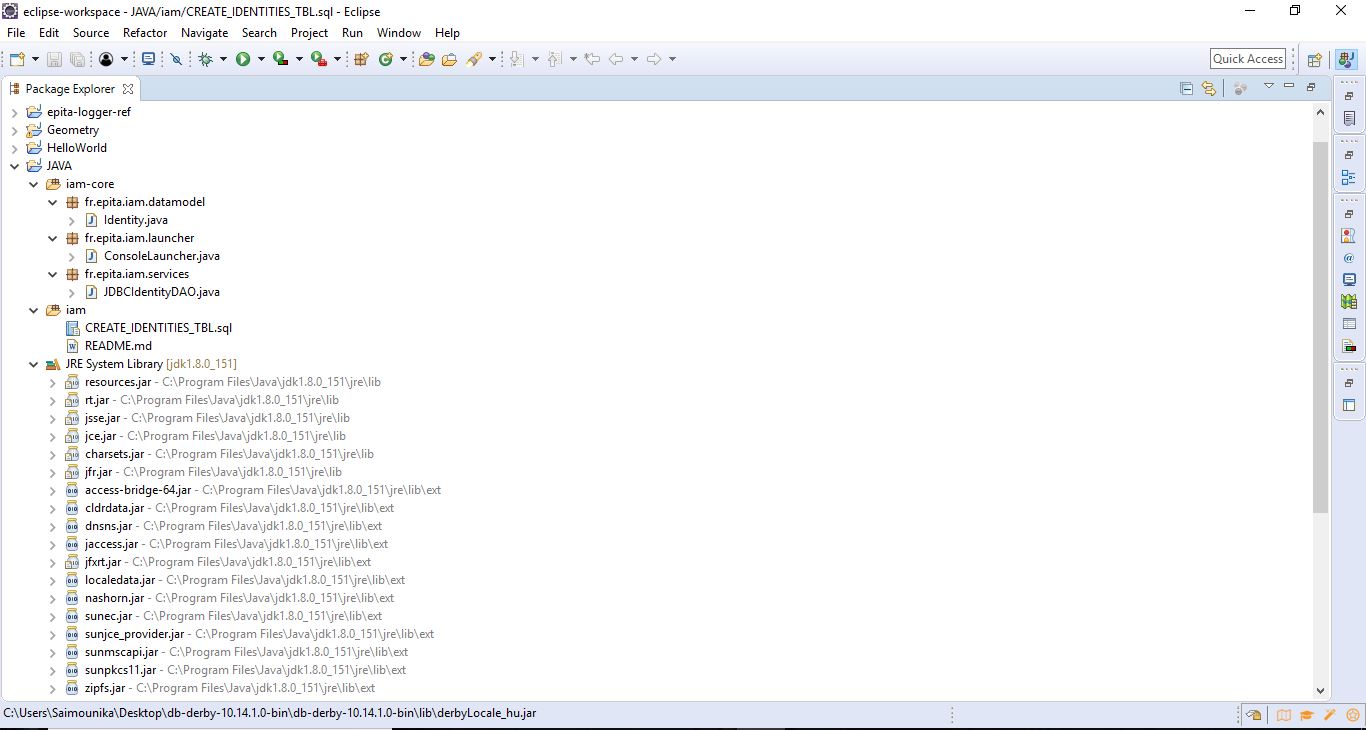
[6 Commented Screenshots 7](#_Toc507092479)

[7 Bibliography 7](#_Toc507092480)

# Subject description

The goal of this project is to make you discover how to program in Java. This project has a good covering in the nowadays Java language usage, you will learn basic things, like creating and using the Java Developer Kit APIs, but also to persist data in databases, or executing an application on a Java application Server. The subject of Identity Management has been chosen because it will put trainees in the Business application world, which is what they will probably do in the next years.

The major purpose of the documentation is to figure out the end to end applications of the project **JAVA (USER IDENTITY MANAGEMENT)**



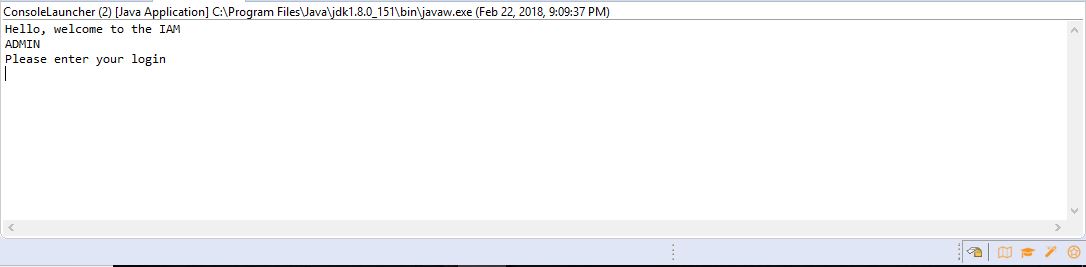


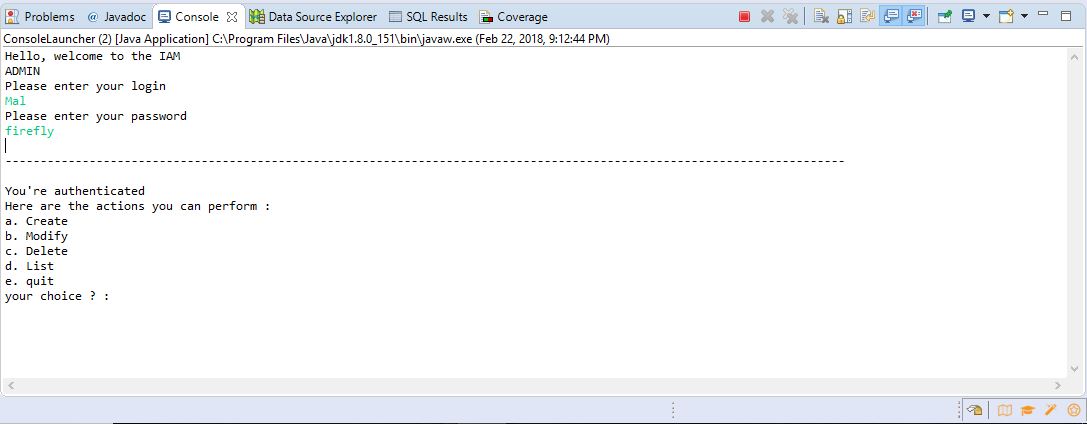
# Subject analysis

## Major features involved in the project are:

The Login details are:

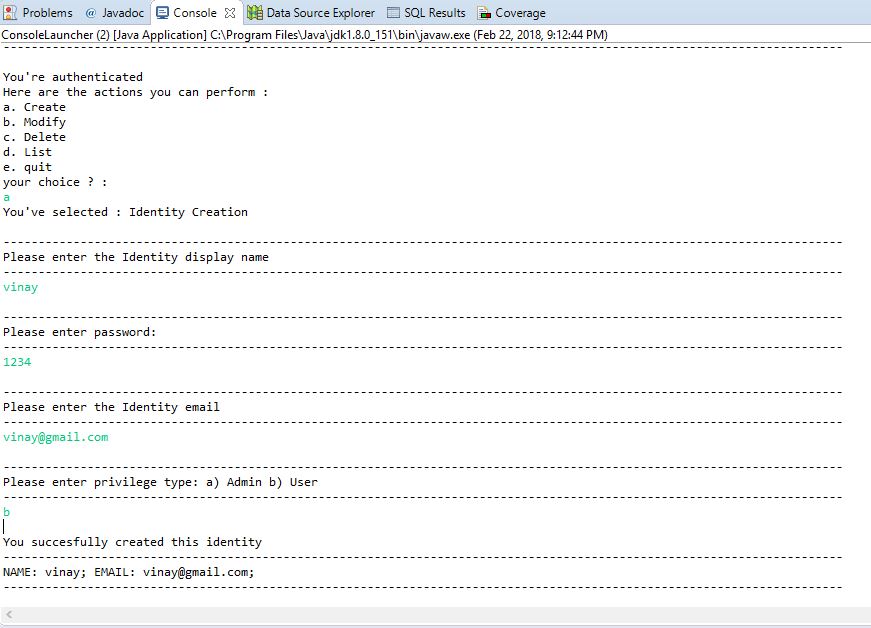
**Login**: Mal (M-capital)  
**Password**: firefly

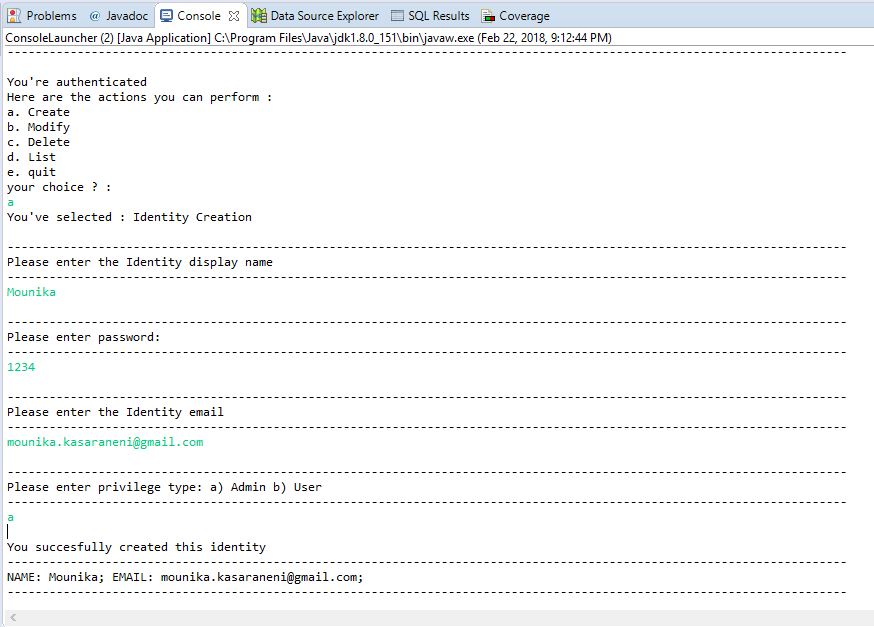




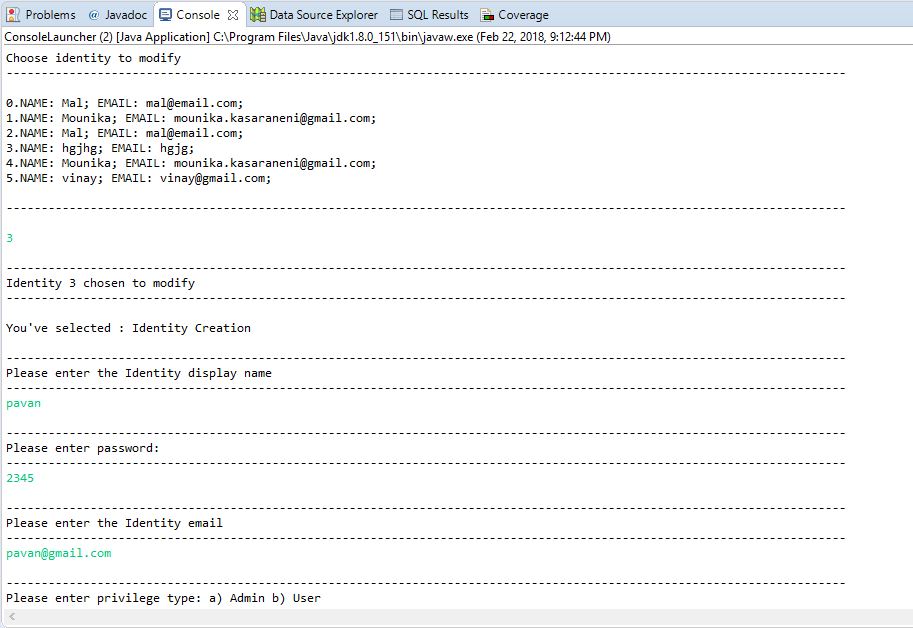
Create: where we can create the user or admin account with user name and password along with the email id.

Here we have created 2 accounts one is for admin and the another one is user.



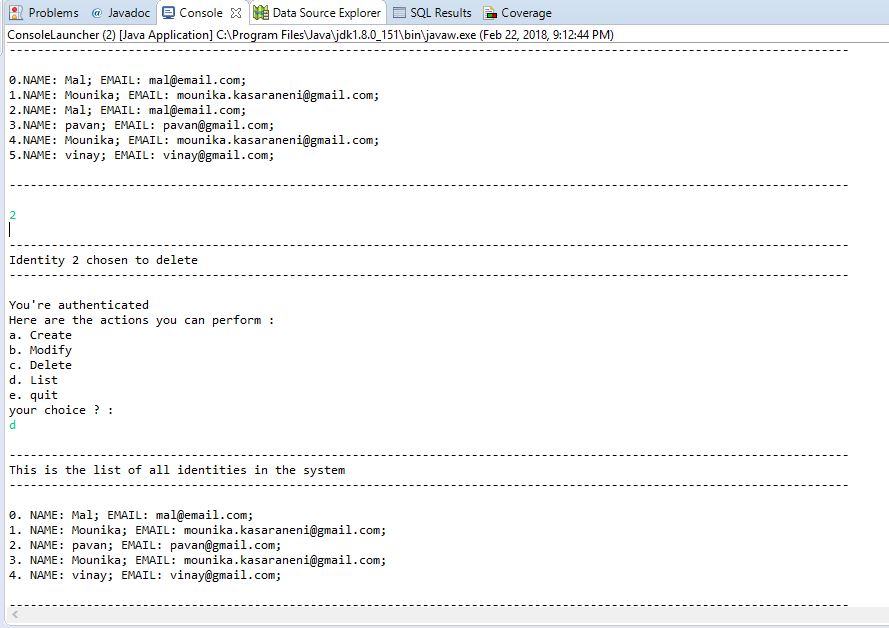


Modify**:** Here in the modify section we can modify the account details.

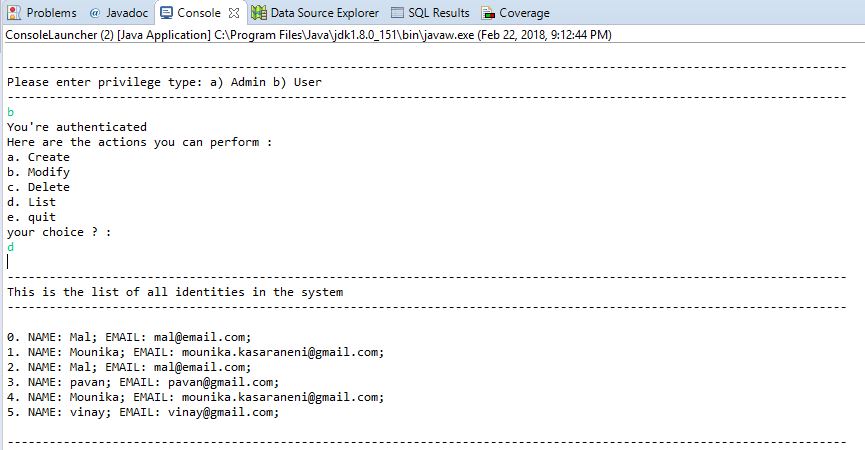


Here we have modified the account details and the result has shown in the **list** column regarding this modified details.

Delete**:** In the delete section we can delete the entire section of the user account or admin account.



List**:** Where we can see the entire details list of the accounts



Exit**:** we can exit from the console



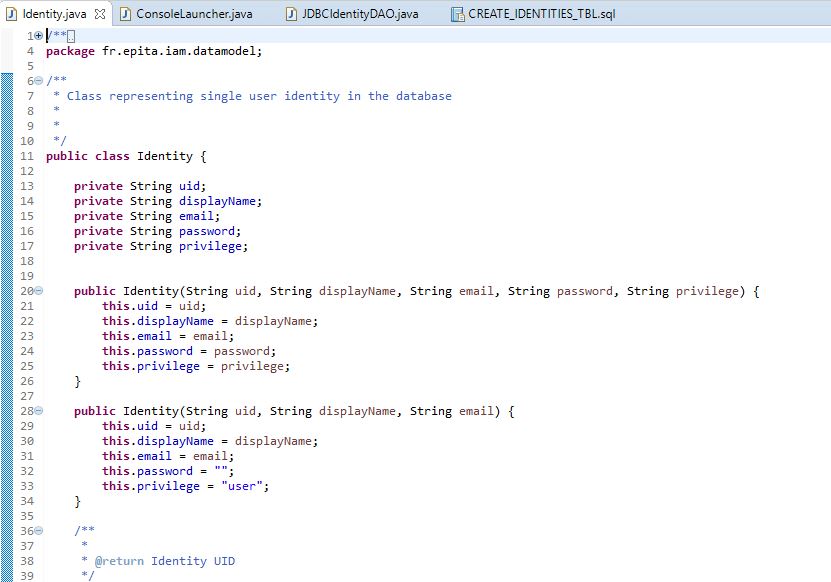
## Application Feasibility

Here the application feasibility states that it easily possible to create a login system and then it gives the permit access to modify the certain records in the database with the help of the java.

## Data description

Data Description being manipulated on the identity object.

* The object contains a string  
    
  Display Name  
  Email  
  Password  
  Privilege



## Expected results

As we are getting the results in the backend by using all the operations with the connectivity of the database without any errors.

## Algorithms study(logic)

We have very minimal knowledge in the algorithm, so we haven’t used any specific algorithm cases in the project.

## Scope of the application (limits, evolutions)

Here in this project we categorize the user into two categories   
user and admin

# Conception

## Chosen algorithm

The reason for not using algorithm is, here we haven’t use any searching techniques’, so that’s the reason we have not use the algorithm.

## Data structures

We have usen the Boolean operations in the ConsoleLauncher.java

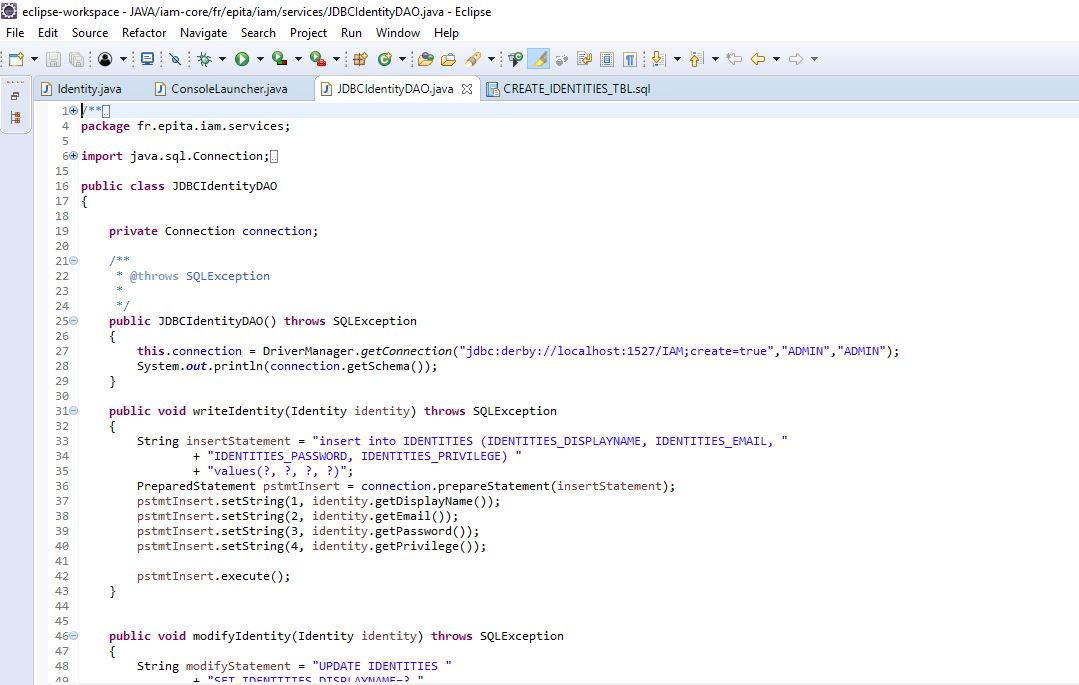
boolean shouldClose = false;  
while(!shouldClose) {  
String answer = menu(scanner);  
switch (answer) {  
case "a":  
// Creation  
createIdentity(scanner);  
break;  
case "b":  
// Modification  
modifyIdentity(scanner);  
break;  
// Deletion  
case "c":  
deleteIdentity(scanner);  
break;  
// Listing  
case "d":  
listIdentities();  
break;  
// Exiting  
case "e":  
shouldClose = true;  
break;  
default:  
System.out.println("This option is not recognized");  
break;

## Global application flow

{HELLO, WELCOME TO THE I AM}

## Global schema and major features schema

JDBC CONNECTIVITY



IDENTITY CONNECTIVITY

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\_PASSWORD VARCHAR(255),

    IDENTITIES\_PRIVILEGE VARCHAR(255)

    );

INSERT INTO IDENTITIES (IDENTITIES\_DISPLAYNAME, IDENTITIES\_EMAIL, IDENTITIES\_PASSWORD, IDENTITIES\_PRIVILEGE)

VALUES ('Mal', '[mal@email.com](mailto:mal@email.com)', 'firefly', 'admin')

# Console operations description

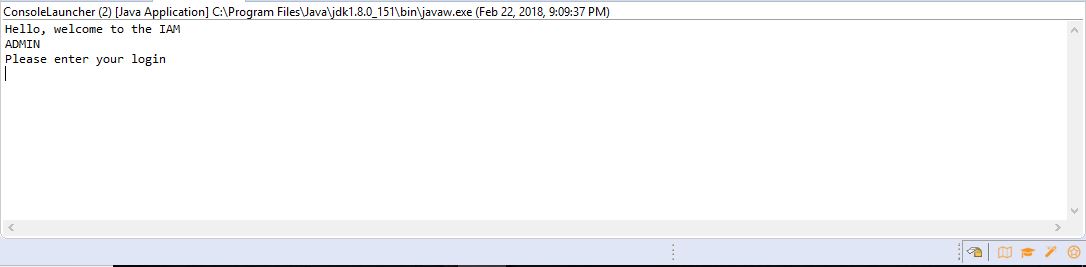
We have provided all the screenshots in detailed, along with the working process.

# Configuration instructions

|  |  |
| --- | --- |
| Driver | org.apache.derby.jdbc.ClientDriver |
| User | ADMIN |
| Password | ADMIN |
| URL | jdbc:derby://localhost:1527/IAM;create=true |

# Commented Screenshots

As of now we have done only the backend, so here we are providing the screenshot for the db.



# Bibliography

We have referred the document which was provided during the semester.