**DETAILED COMPREHENSIVE REPORT**

**ON**

**JAVA CRUD OPERATION PROJECT APPLICATION**

**By**

**SAURABHKUMAR GAJJAR**

******

**Master of Computer Science(Software Engineering)**

**14-16 Rue Voltaire, 94270 le Kremlin-BicêtreDETAILED COMPREHENSIVE REPORT**

**PROJECT**

Guided By

**THOMAS BROUSSARD**

**Product management at NatSystem**

****

**CONTAINS**

1. **SUBJECT DESCRIPTION…………………………………………………………..3**
2. **SUBJECT ANALYSIS………………………………………………………………..4**
   1. **Major Features……………………………………………………………...4**
   2. **Application Feasibility…………………………………………………….4**
   3. **Data Description.…………………………………………………………...4**
   4. **Expected Results……………………………………………………………4**
   5. **Algorithm study……………………………………………………………..5**
   6. **Scope of the Application………………………………………………….8**
3. **CONCEPTION………………………………………………………………………….8**
4. **Chosen Algorithm…………………………………………………………………..8**
5. **Data Structure……………………………………………………………………..10**
6. **Global Application view…………………………………………………………10**
7. **CONSOLE OPERATIONS DESCRIPTION…………………………………....11**
8. **CONFIGURATION INSTRUCTION…………………………………………….14**
9. **Subject Description**

This Project is giving the overall behavior and understanding about most intense java concepts. For example, Interfaces, Inheritance, Java environment, DAO(data access object). This project is created on Eclipsce-jee-oxygen by using JDK1.8.0\_151. This project is also using derby database for Database operations. This Project is basically calling 4 methods Create, Update, Search(Read), Delete operations. Means to read and store information about identities. This feature can be achieved in several ways (these means are listed from the simplest to the hardest).

The main goal is to manage input which is entered by users by Identity in data model package.It creates, or establishes the user identity, user identity operations, and finally the destruction of the user identity within the organization. Identity management controls these operations using an approval process that is required by the business. Manages access control for various resources (systems and applications) within the enterprise. Access management also includes user account management. The goal of this project is to develop an application using Java which can automatically connect to a database and store information in the database and to retrieve or fetch, modify and delete the information in the database.

* Create an Identity
* Update an Identity
* Delete an Identity
* Read an Identity(Search).
* Authentication

Here we have the Environment Development Details

Platform : Windows 10

Language : Core Java

IDE : Eclipse-JEE-Oxygen

Database : Apache Derby 10.14.1.0

**II. Subject Analysis**

**Here, we see the analysis of “DatabaseOperations 2” and limitation of this project.**

**A. Major Features**

The user should be able to

* Create,
* Update,
* Delete
* Search Identities from the application.

The sub features we can say user - friendly, Easy-to-use, Data entry restricted to avoid errors. It also contains Authentication so it gives security to handle the data.

**B. Application Feasibility**

If we see cost feasibility every tool and software is open source. So it is feasible within the estimated cost. It is also technically feasible once you use you evaluate it. There is no dependency on GUI, so it take less time to run and give expected results.

**C. Expected Results**

This instrument is fit for verifying the client, making another character, refreshing a current personality, and erasing a personality from the database. The database utilized is a derby database. The apparatus needs to speak with the database and come back with the outcomes in speedy time. It is able to run four main operation and give output which we have exactly entered.

**D. Algorithm studies**

It is very simple and Console level application.

We are taking the choice from user what he/she to do in CRUD operation.

while(true){

Login(); // authentication part

System.out.print(“Enter choice:”);

if(choice.isEmpty()){

System.out.println("ENTER A CORRECT NUMBER!!!!");

}

switch(choice) {

case "1":

{clr(); // to clear console

getalldata(); // take data from user and create data

clr();

break;}

case "2": clr();

searchdata(); // take choice from which enetity he/she wants to search from database

break;

case "3": {clr();

updatedata(); // take input and update it.

clr();

showdata();

break;}

case "4": {

clr();

deletedata(); //Delete a data based on Userid

clr();

showdata(); break;}

default: break;

}

}

we created IdentityDAO as a interface class so we can use it as by inheritance like it contains four operations or four methods. So we can use them to create File(IdentityFile), Database(IdentityJDBCDAO), xml(IdentityXML), ext. So it is the best example for inheritance concept.

By DAO concept we are using object of Identity.java file and use it in this three with the same arguments.

Here, we can see one of the method from Main class.

private static void searchdata() throws IdentityCreationException{

final Identity id1 = new Identity();

String rowname;

System.out.println("By which Identity do you want to search: ");

rowname = sc.next();

if(rowname.contentEquals("displayname")) {

System.out.println("Enter DisplayName for search: ");

id1.setDisplayName(sc.next());

}

else if(rowname.contentEquals("emailid")) {

System.out.println("Enter Email Id for search: ");

id1.setEmail(sc.next());

}

else if(rowname.contentEquals("uid") ) {

System.out.println("Enter User id for search: ");

id1.setUid(sc.next());

}

dao.search(id1);

sc.close();

}

So we can see that we are calling search() file from the IdentityJDBCDAO by creating dao object. It is passing the data by Identity Object which is id1 in IdentityJDBCDAO file.

This was the overview of DatabaseOperation2 project now we study connection, preparedstatement and SQL query to do CRUD operation.

public void create(Identity identity) throws IdentityCreationException {

Connection connection = null;

try {

connection = getConnection();

final PreparedStatement preparedStatement = connection.prepareStatement("INSERT INTO IDENTITIES(UID, DISPLAY\_NAME, EMAIL) values (?,?,?) ");

preparedStatement.setString(1, identity.getUid());

preparedStatement.setString(2, identity.getDisplayName(); preparedStatement.setString(3, identity.getEmail());

preparedStatement.execute();

} catch (ClassNotFoundException | SQLException e) {

LOGGER.error("error in create method :" + e.getMessage());

final IdentityCreationException businessException = new IdentityCreationException(identity, e);

throw businessException;

} finally {

try {

if (connection != null) {

connection.close();

}

} catch (final SQLException e) {

System.out.println(e.getMessage());

e.printStackTrace();

}

}

}

PreparedStatement is helping to execute SQL query and we have call getConnection() method which is static for IdentityJDBCDAO.java. We can see that DatabaseOperation have a source Project which is epita-logger.

This project is helping to Configuration.java File to Configure url, username and password of Derby database. It is created for when we use this project on other machine is will search url, username, password on that machine with path. Configuration file is taking string input and set the propertyKey to load it on database.

It is calling from testConfiguration.properties file where we can have these three data so we can set it as per other machine where we are running our Project.

For that first we have to pass the VM arguments in Run Configuration like

-Dconf = testConfiguration.properties.

VM arguments are typically values that change the behaviour of the Java Virtual Machine (JVM).

Logger.java is for good coding purpose. It gives info, Debug message and also creates application.log where we can find time where we just run the Project.

**F. Scope of the Application**

For User Interface, we can have the GUI(Graphic User Interface) in next modification.

For Privacy and Security we can add it.

As expressed over, the application is constrained to just oversee Identities and not to deal with the Users. This is obviously one of the initial steps to be enhanced in a next modification.

Additionally, the Identities are just going to be overseen on a database, there is no present method to choose a document as capacity mode for Identities, so the client needs to have introduced a Database director programming.

**Limits**

Lack of client GUI/web interface which could bother first time clients.

Possibility of decoding the framework secret key.

Does not open the framework to oversee consents and qualities of clients.

Only one user created this project.

**Evolution**

I am targeting to add GUI and spring it with my Project and also we can make other users to register in it.

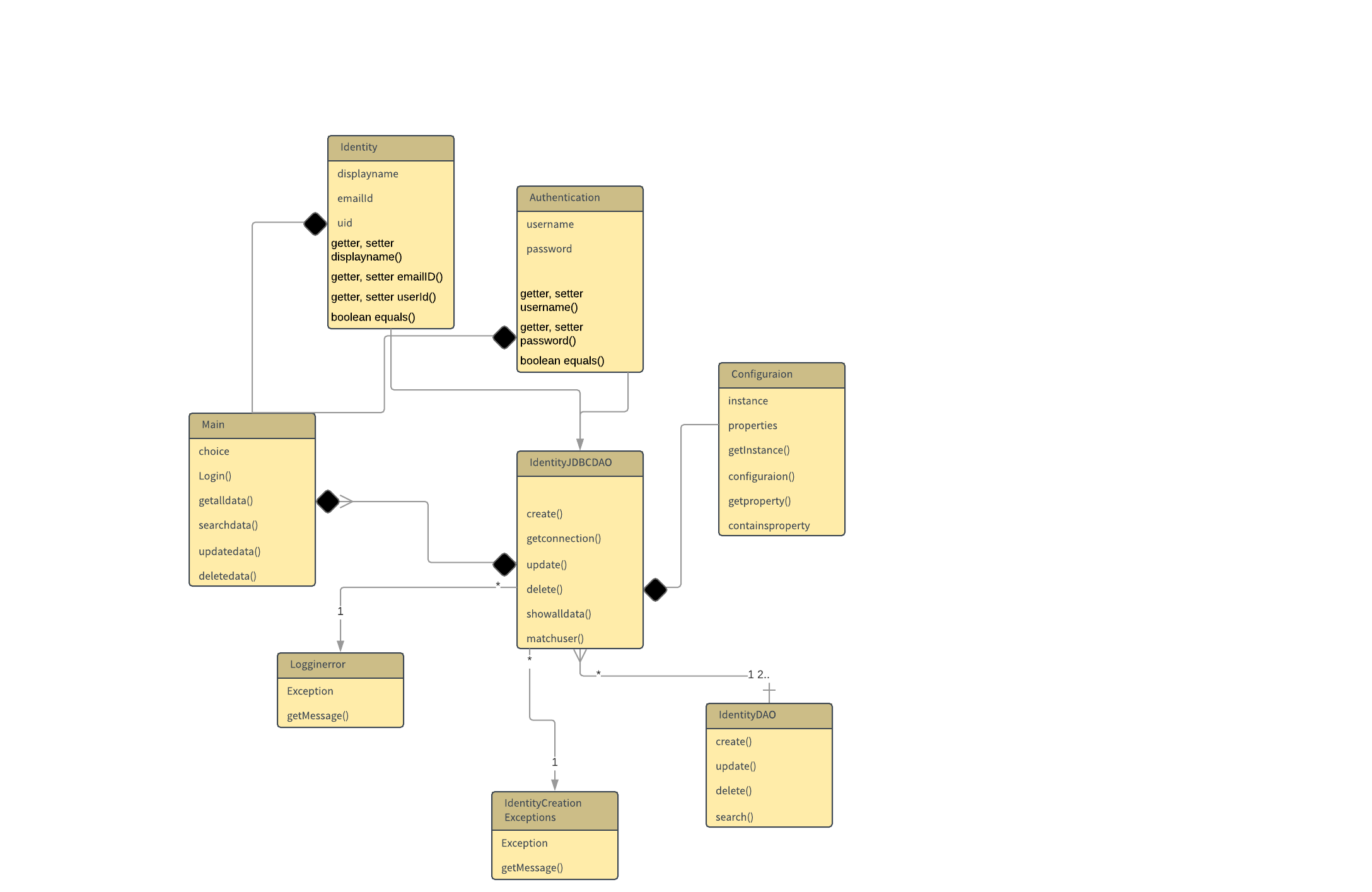
**III. Conceptions**

**A. Chosen Algorithm**

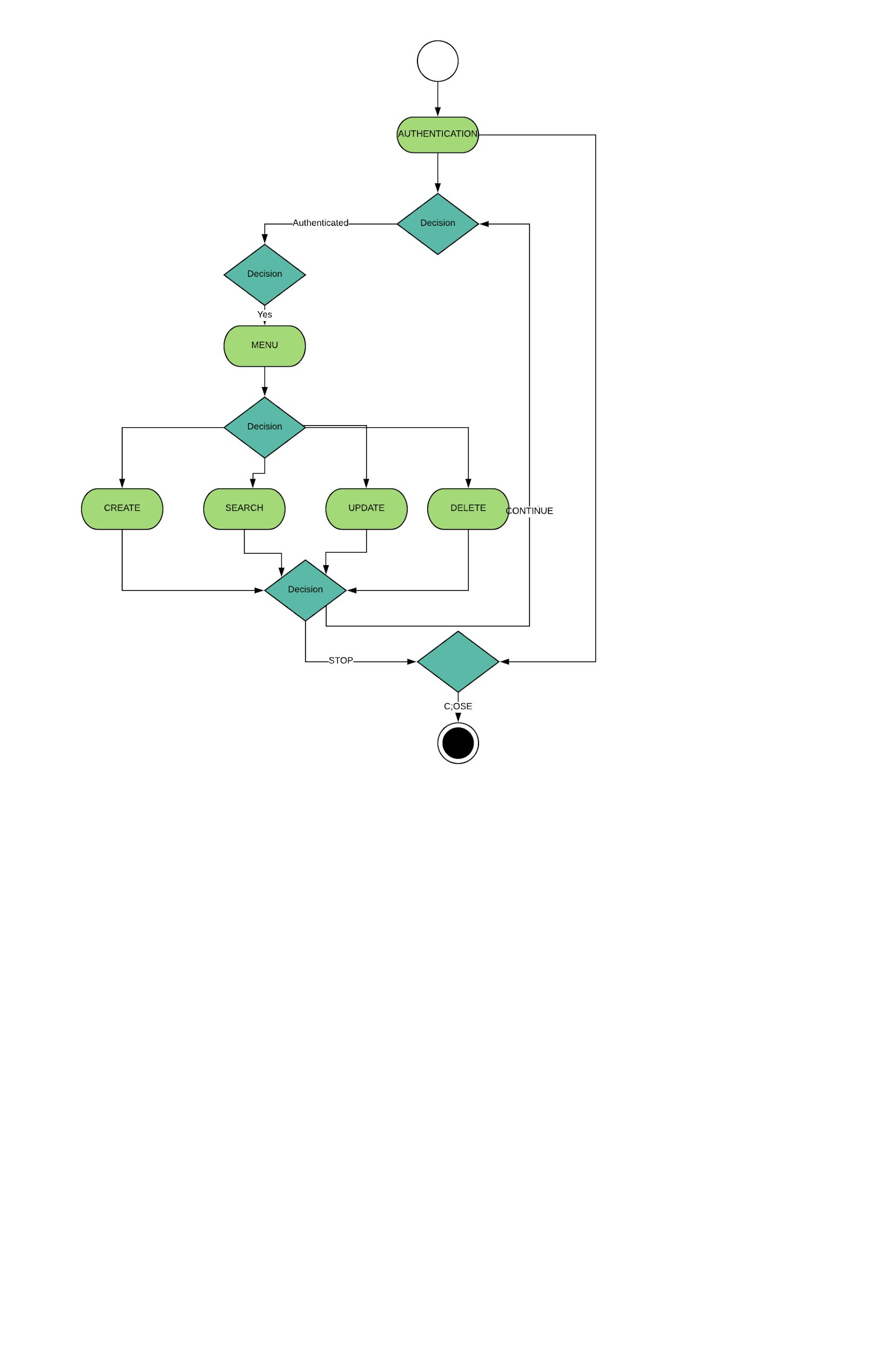
In DatabaseOperation2, there are four packages in source folder. Each contains utility as their name. For example,

* fr.epita.DatabaseOperation2.launcher is containing Main file which is run or launch for run.
* fr.epita.DatabaseOperation2.datamodel is containing Identity, Authentication which are used to take a data or set a data.
* fr.epita.DatabaseOperation2.services is providing services files like IdentityDAO, IdentityJDBCDAO, Configuration.
* fr.epita.DatabaseOperation2.exceptions is providing Exception files which I have created we can say that user defined.

**B. Data Structure(Class Diagram)**



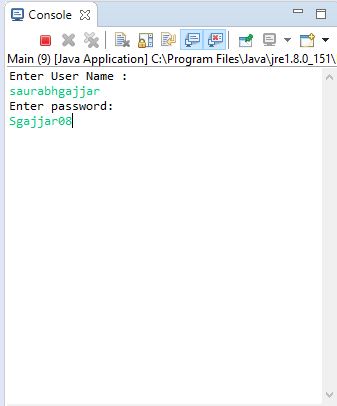
**C. Global application flow**

****

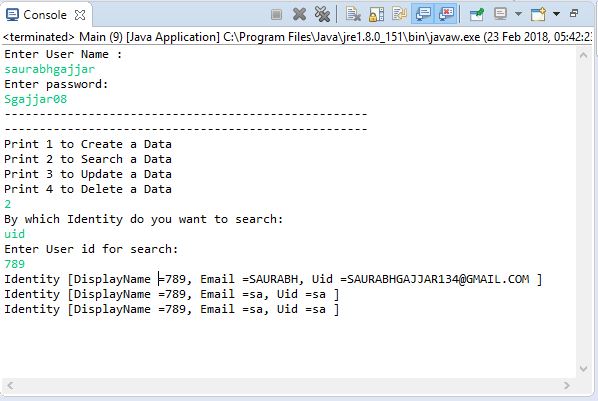
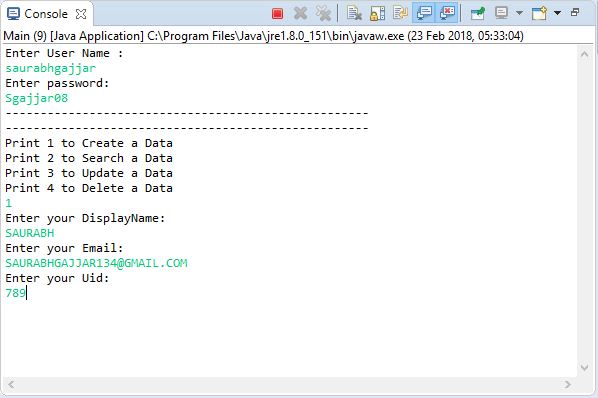
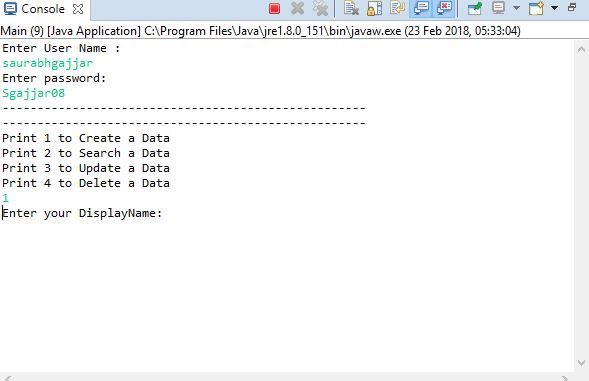
**IV. Console Operation Description**

Here, I am sharing some console screenshots. To Run this Project go to Run → Run Configuration → Arguments→ type -Dconf = testConfiguration.properties → Apply→ Run.

Now you can see console part



I just enter user-name and password now i choose 1 to create data.



here we can see that data identity has been created.

**V. Configuration Instruction**

Our Project is taking the database connection string check it is same url or not. if not then you probably create new database with the same host , url(data name), username and password and check it is now okay or not.

Check out the project path also if it is not same then it might get problem.

Check VM arguments in Run Configuration. It should be like -Dconf = testConfiguration.properties.

At Console Enter username = saurabhgajjar and password = Sgajjar08.