**TECHNICAL DOCUMENTATION**

**“IAM-CORE”**

**JAVA PROJECT**

**MSC 2018 FALL**

**Presented by :-   
Thejus and Bhanuja Kaur**

**Table of Content**

1. Subject description 3
2. Subject analysis 3
   1. Major features
   2. Application Feasibility
   3. Data description
   4. Expected results
   5. Algorithms study
   6. Scope of the application (limits, evolutions)
3. Conception 4
   1. Global application flow
   2. Global schema
4. Console operations description 5
5. Configuration instructions 5
6. Bibliography 6

**Subject description**

“Iam-core” is a program made in JAVA language using Derby DB for performing basic CRUD(Create , Read , Update and Delete Operations) on data. It also has a feature to manage users and authentication which lets the user perform CRUD operation on user Identities. It also features a logger for logging all the activities in the application.

**Java** is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. We have used Java 8 latest version for development of our application.

**Apache Derby**  is a relational database management system (RDBMS) developed by the Apache Software Foundation that can be embedded in Java programs and used for online transaction processing.

**Subject analysis**

**1. Major Features**: iam-core has 2 core operations.

* 1. User management and authentication.
  2. Create , Read , Update and Delete operation on the DB to manage Identity details of user.

**2.Application Feasibility:** iam-core is free open source project which utilises java 8 and Derby DB which is also an open source Data base distributed under apache 2.0 licence. The project also employs a custom made logger, thus the project is completely feasible and free to use and modify.

**3.Data description:** iam-core uses various data types mainly String ,ListArray and doubles.  
Derby DB contains all the data in VARCHAR format. Storing the data as String helps in easy manipulation and retrieval of data.

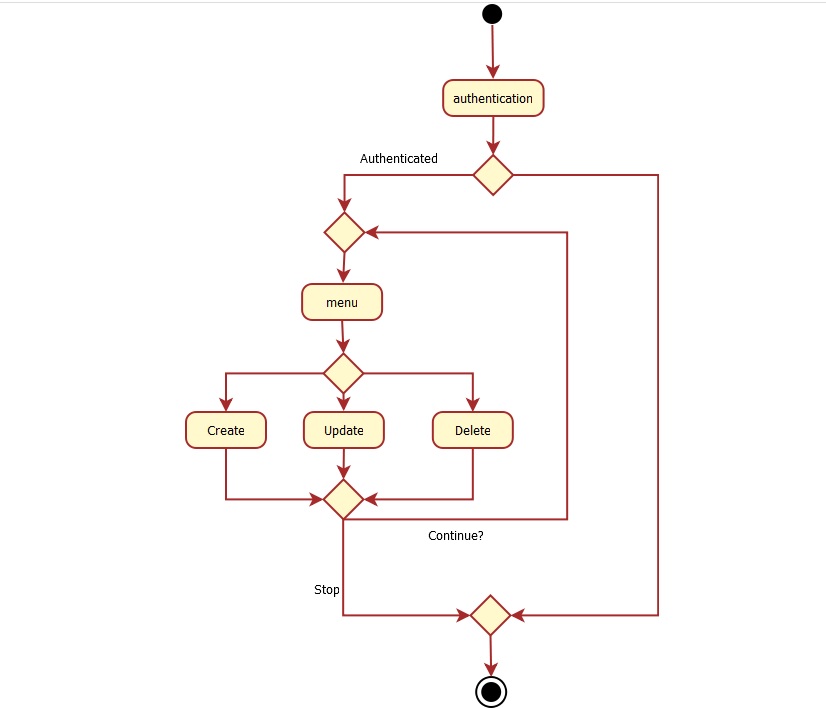
**4.Expected results:** iam-core application is expected to perform basic user account creation, login and then basic CRUD operations on user identities.  
Application is expected to create user Identity, Update them , read them and delete them as needed. Application is also expected to log all the activities in the application.log file.  
Application is expected to handle exceptions during runtime.

**5.Algorithms study:** iam-core has employed a basic interface and abstract class method to create Identity CRUD operations and interface to handle user management.  
It employed a TDD method for development of the application and also at times GWT method to perform basic testing of different modules.

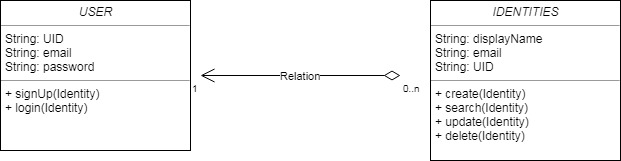
**6.Scope of the application (limits, evolutions):** Application has a few limitations mainly unavailability of a GUI i.e the application runs mainly on console operations which is less pleasing and less gives less control to the user.  
Application lets the user update only the email and display name and not the UID.  
With introduction of GUI and few security measures the application is expected to thrive well in the market to manage users and Identities.

**Conception**

**Global Application Flow:**

****

**Global Schema:**

****

**Console operations description**

**Step 1** : Configure the DB as described in the configuration Section.

**Step 2** : Run the “Main.java” file in the Launcher Package.

**Step 3** : Select whether you want to Login or Signup, enter the corresponding Menu option **Login**-> Enter UID and Password, System then searches and if a matching User profile found, then it moves to the next Step.

* **SignUp**-> Enter UID, email address and passwords all as Strings.

**Step 4** : Main selection menu appears where you can select different CRUD Operations to perform on the DB data.

**Step 5 :** Create Data-> Enter Display name , email and UID.

**Step 6** : Search-> Search for the entered data in the DB. It gives a menu to search by name, email or UID. Select respective menu option and then it displays result of the search.

**Step 7** : Update-> Updates the data in the DB. It searches the data by UID and then lets you update name and email, but not UID.

**Step 8** : Delete-> It gives you option to delete Data in the DB by name , email or UID. It then deletes the data from DB.

**Step 9** : End of all Operations.

**Step 10:** Run “TestDBOperations.java” to perform a test run of the application without console menu and static data.

**Step 11:** Run “index.html” file in the doc folder for javadoc

**Configuration instructions**

For easy use of application and easy configuration of the Derby Data Base we have exported the file containing the configuration properties and db.properties in the java project files.  
Open the db.properties file and fill in the following details with relevance to your Data Base.

DRIVER=org.apache.derby.jdbc.ClientDriver

USER=root(your root)

PASSWORD=root(your password for root user)

URL=jdbc:derby://localhost:1527/iam-b;create=true (url of the db)

**Bibliography**

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html **- JAVA SDK 8u161**

https://www.eclipse.org/ - **Eclipse Oxygen IDE**

https://www.google.fr – **Google**

https://stackoverflow.com/ - **StackOverflow**

https://www.draw.io/ - **Draw**

http://thomas-broussard.fr/work/java/courses/project/fundamental.xhtml