

EPITA

IAM Project

Fundamental Java Project

Team Members:

Sohel Khan Pathan

Omkar Yadav

Table of Contents

1. Subject Description	3
2. Subject Analysis	3
2.1. Major Features	3
2.2. Application Feasibility	3
2.3. Data Description	4
2.4. Expected Results	4
2.5. Algorithm Study	5
2.6. Scope of application (limits, evolutions)	5
3. Conception	5
3.1. Chosen algorithm	5
3.2. Data structures	5
3.3. Global application flow	6
3.4. Global schema and major features schema	7
4. Console operations description	7
4.1. Authentication:	7
4.2. Identity Creation:	8
4.3. Updating an Identity:	8
4.4. Identity Deletion:	8
4.5. Identity Search:	9
5. Configuration instructions	9
5.1. Development tools:	9
5.2. Schema and Database Configuration:	9
6. Commented Screenshots	10
7. Bibliography	10

1. Subject Description

The project implements basic features of IAM (Identity and Access Management) system. The system is implemented in two major elements, authentication of user and identity management. This defines user identity life cycle within an organization. It creates, or establishes the user identity, user identity operations, and finally the destruction of the user identity within the organization.

The project is implemented using feature of JAVA language, which allows user to communicate with the database using a Console. A user will be asked a series of inputs to create, update or delete an identity.

2. Subject Analysis

2.1. Major Features

Following are the major feature of IAM system:

- **Authenticate the user** – This feature allows the system to validate the user's identity, before permitting to use to IAM system.
- **Create an Identity** – To create an identity in the system with an UID, Name and Email address.
- **Update an Identity** – To update an already existing identity with an UID with new Name and Email address.
- **Delete an Identity** – To delete an identity with an UID.
- **Search/List the identities** – To search and list the identities on console based on user's input with parameters UID, Name and Email address.

2.2. Application Feasibility

- The system allows simple way to maintain the identities by using the simple console level inputs, which allows to user to freely interact with the system.
- All of the development tools used while developing like Eclipse development kit, Derby database and JDK are easily accessible.
- Considering overall security of Identity management, a strategic approach is followed for maintaining the identities and appropriate, self-explanatory error messages are raised when user fails to follow the process.

2.3. Data Description

Identities and users are stored in the Derby Database by using the JDBC connection as a bridge between program and the database. Considering the scope of the development, users are manually maintained using a basic insert query. It is essential to maintain the users before executing the program.

On the other hand, Identities are maintained by system with the three different operations to create, update and delete in the database.

Database table are user to store both Users and Identity, following is the structure of the table:

- Users:**

Field Name	Data Type	Field length	Primary Key	Field description
USERNAME	VARCHAR	10	Yes	User name
PASSWORD	VARCHAR	10	No	Password

- Identities:**

Field Name	Data Type	Field length	Primary Key	Field description
UID	INT		Yes	Unique Identifier for an Identity
DISPLAYNAME	VARCHAR	15	No	Display name for an identity
EMAIL	VARCHAR	30	No	Email ID of the identity

2.4. Expected Results

- Authentication:** User will be allowed to access basic identity management features, after user is authenticated. System will check the user input against database table “User”, if entry exist then user is an authorized user.

- Identity Management:**

- After the user authentication, a menu is displayed with the possible operations.
- Following are the list of operations:

Operations Number	Operation
1	Create Identity
2	Update Identity
3	Delete Identity
4	Search Identity
5	Exit Program

- Appropriate error **messages** to be displayed for each operation.

2.5. Algorithm Study

- Console based application.
- UI based application
- Application on mobile platform

2.6. Scope of application (limits, evolutions)

Identity Management is the core functionality based on the requirement of the application, user management is not considered in the scope of the application and it has been assumed as a part of user management project.

Also, the Identities are only going to be managed on a database, there is no current way to select a file as storage mode for Identities, so the user has to have installed a Database manager software.

Limitation:

- The system is based on console format, an User Interface would be more appealing and easier for new user.
- Lack of encryption for the password for users and sensitive information of an identity.
- Lack of integration between user and identity, result in lack of user role assignment.
- Maintaining one identity at a time would be time-consuming job considering there are numerous number to be maintained as a part of IAM system.

Evaluation:

- Concept of user and identity can be integrated by adding functionality to assign roles to the user based on the operation.
- User Interface can be developed with the functionality to add multiple identities at same time.

3. Conception

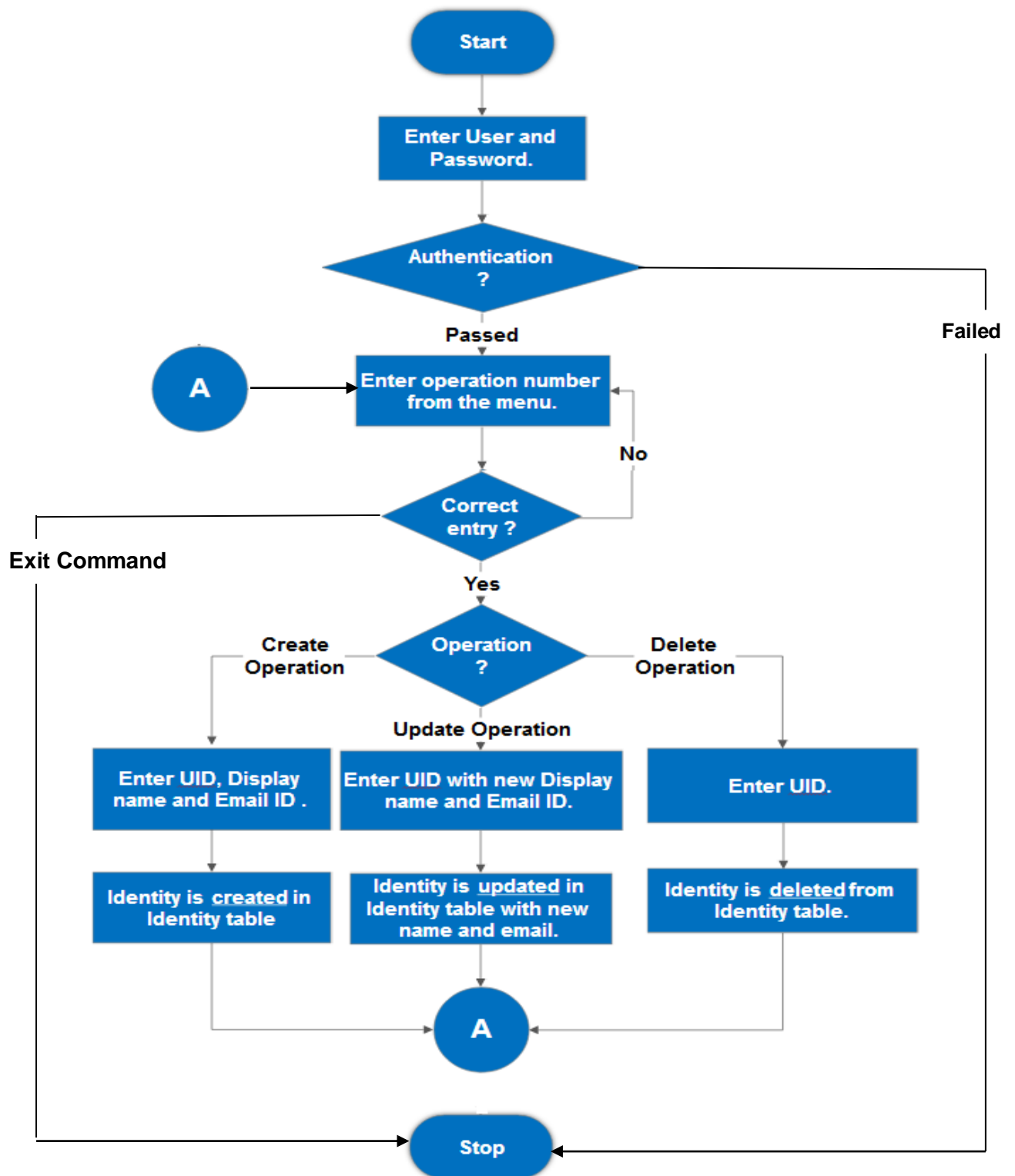
3.1. Chosen algorithm

- A console based application is used for development of the project.
- For executing the operations multiple times, a while loop is used. Based on the operation selected by the user the create, update and delete functionality will be carried out.

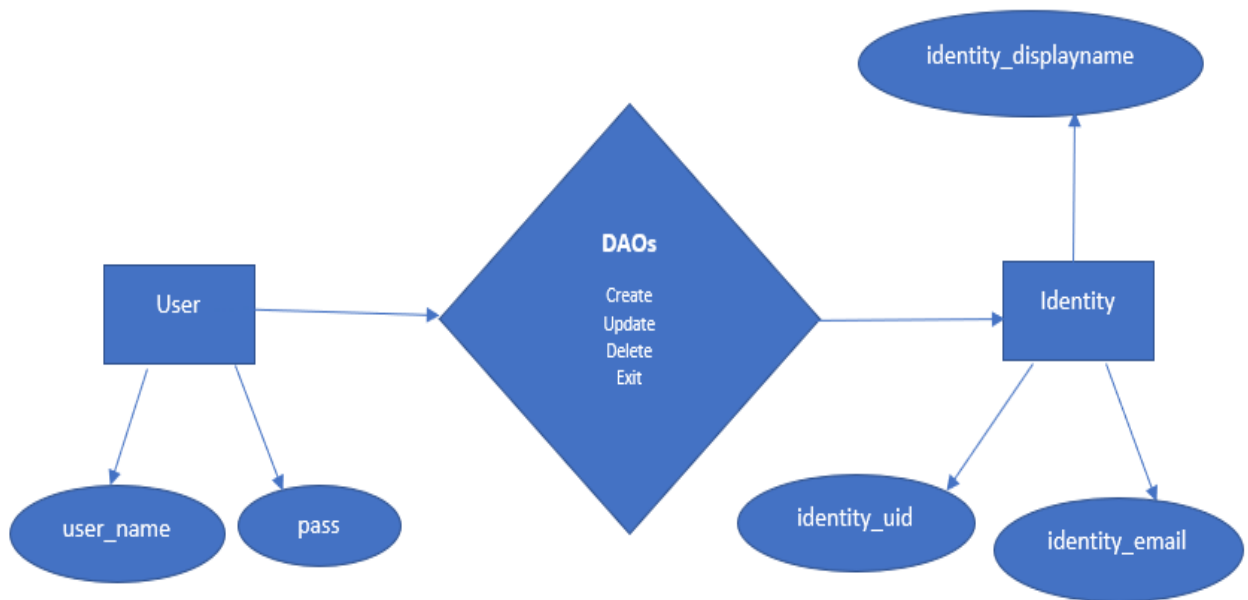
3.2. Data structures

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.

3.3. Global application flow



3.4. Global schema and major features schema



4. Console operations description

Following are the console operations implemented:

4.1. Authentication:

- This operation is responsible to authenticate the user with the User ID and Password.
- Authentication is implemented using a database table “Users”, the user is
 - authorized user, if User ID and Password entry exist in the table.
 - unauthorized user, if User ID and Password entry does not exist in the table.
- Program will be terminated, when authorization fails.
- Main program will call **authorization()** method in class “IdentityJDBCDAO”, method will return Boolean value, true for authorized user and false for un authorized user.
- Input parameters for method:

Parameter name	Type	Description
username	String	User Name
passwd	String	Password

4.2. Identity Creation:

- This operation is responsible to create an identity using UID, Display Name and Email ID.
- After the Identity creation, a new entry will be created into database table “Identities” with UID, Display Name and Email ID.
- Main program will call **identity_create()** method in class “IdentityJDBCDAO”.
- Input parameters for method:

Parameter name	Type	Description
identity	Identity (class)	Identity with UID, Display Name and Email ID.

- Custom exception “IdentityCreationException” is developed for handling exceptions for this operation.

4.3. Updating an Identity:

- This operation is responsible to update an identity using UID with new Display Name and Email ID.
- After this operation, an existing entry for the UID will be updated into database table “Identities” with new Display Name and Email ID.
- Main program will call **identity_update()** method in class “IdentityJDBCDAO”.
- Input parameters for method:

Parameter name	Type	Description
identity	Identity (class)	Identity with UID, Display Name and Email ID.

- Custom exception “IdentityUpdationException” is developed for handling exceptions for this operation.

4.4. Identity Deletion:

- This operation is responsible to delete an identity using UID, Display Name and Email ID.
- With this operation, identities will be listed on the console with respect to fields UID, Display Name and Email ID from the database table “Identities”.
- Main program will call **identity_list()** method of type list in class “IdentityJDBCDAO”.
- Input parameters for method:

Parameter name	Type	Description
criteria	Identity (class)	Identity with UID.

- Custom exception “IdentityDeletionException” is developed for handling exceptions for this operation.

4.5. Identity Search:

- This operation is responsible to search the identities using UID.
- After this operation, an existing entry for the UID will be deleted from the database table “Identities”.
- Main program will call identity_delete() method in class “IdentityJDBCDAO”.
- Input parameters for method:

Parameter name	Type	Description
identity	Identity (class)	Identity with UID, Display Name and Email ID.

- Custom exception “IdentityListException” is developed for handling exceptions for this operation.

5. Configuration instructions

5.1. Development tools:

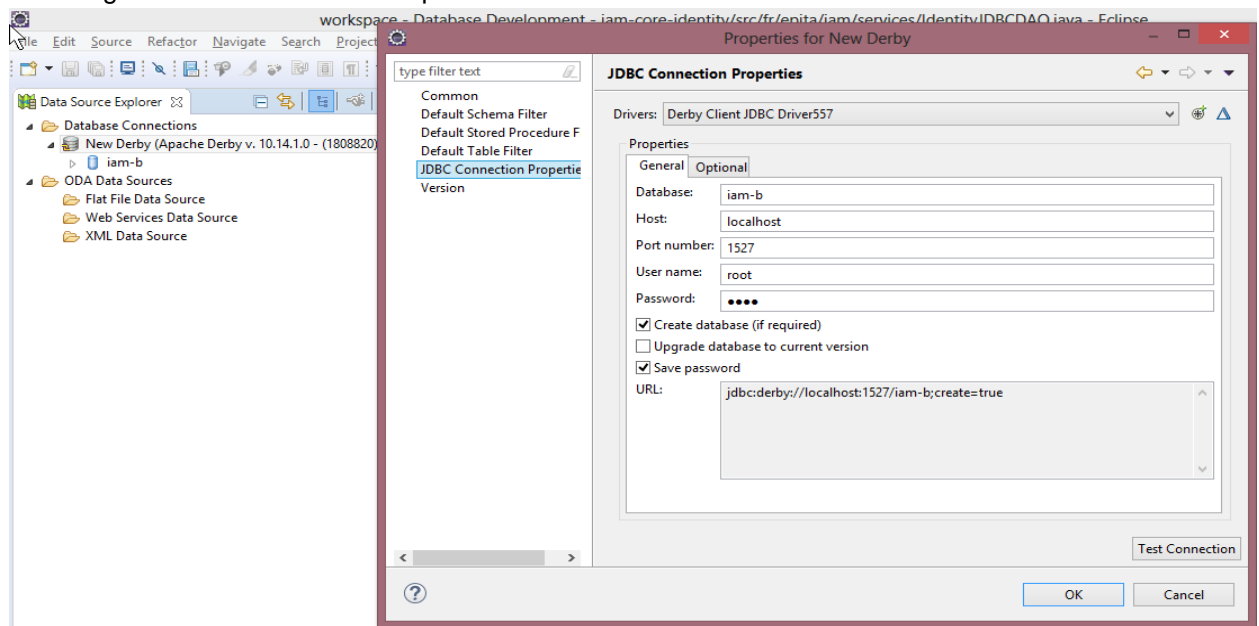
- [Eclipse Oxygen](#) eclipse Java EE IDE for Web Developers.
- [Derby Database](#).
- [Java JDK 8](#).

5.2. Schema and Database Configuration:

- For the development, we have mainly used two database tables: “Users” for user management and “Identities” for managing the identities.
- An “init.sql” file is maintained under “SQL” folder with all the required queries to create the tables.
- As mentioned earlier, we will be creating manual entries for table “Users”.
- Configuration file “testConfiguration.properties” is created by passing the connection string.
- Following are the properties:

Properties	Actual Values
Database	iam-b
Host	localhost
Port Number	1527
User Name	root
Password	root
Connection String	jdbc:derby://localhost:1527/iam-b;create=true

- Following is the screenshot with required database details:



6. Commented Screenshots

- Commented screenshot for the operations are covered as a part of User Guide document.

7. Bibliography

- Project Scope, Design patterns, OO concepts: <http://thomas-broussard.fr/>
- Working with List Data Structure: <https://docs.oracle.com/javase/tutorial/collections/interfaces/list.html>
- Overview of IAM System: <https://www.linkedin.com/pulse/major-factor-identity-access-management-iam-project-failure-mernacaj>
- Programming 1: <https://stackoverflow.com/>
- Programming 2: <https://www.w3schools.in/category/java-tutorial/>