**USER PROFILE MANAGEMENT**

Submitted by

Abdul Khader Shahid.M

Silambarasan Somasundaram.S

Master Computer Science -2018

EPITA

Professor: Thomas Broussard

**EPITA**

|  |  |
| --- | --- |
| TABLE OF CONTENTS | Page.no |
| 1. Subject Description | 03 |
| 2. Subject analysis | 03 |
| 2.1. Major features | 03 |
| 2.2. Application Feasibility | 03 |
| 2.3. Data description | 03 |
| 2.4. Expected results | 03 |
| 2.5. Scope of the Application | 04 |
| 3. Conception | 04 |
| 3.1. Database Structure | 04 |
| 3.1.1. Database Table | 04 |
| 3.1.2. Derby | 04 |
| 3.2. Global Application Flow | 04 |
| 3.2.1. Existing User Flow Description | 04 |
| 3.2.2. New User Flow Description | 04 |
| 3.2.3. Flow Chart | 05 |
| 3.3. Global Schema | 06 |
| 4. Console Operation Description | 07 |
| 4.1. User Authentication | 07 |
| 4.2. User Creation | 08 |
| 4.3. User Delete | 09 |
| 4.4. User Update | 09 |
| 4.5. User Search | 10 |
| 5. Configuration Instruction | 10-12 |
| 6. Bibliography | 10-12 |

**1. SUBJECT DESCRIPTION:**

The goal of this project to manage a user information system in secure manner to do create, access and modify operation by using database. Existing user can updater, view their information by logging the application. To create an account in this application user has to fill information like UID which has to be unique. Each user validated by UID and password.

**2. SUBJECT ANALYSIS:**

**2.1 Major Features:**

* User Authentication.
* Logging user information and Error.
* Password Matching verification
* Create user
* Update information
* Delete information
* Search User

**2.2 Application Feasibility:**

In this application, we can store unique user information to avoid the duplication value by UID. Searching mechanism will be faster through user data values indicated by single letter in program.

The result produce in a given time and it response with certain condition effectively. It throws proper error message when user made any mistake in the program and when their inputs are not matched. We provide guideline Menu to help the user to perform their action.

**2.3 Data Description:**

The data in this application it’s specific user information like email-id, user-name to manage their own profile in this program.

**2.4 Expected Result:**

When open the program it has to guide the user to do the action by displaying menu in console and user can select some options by pressing buttons.

An action or exception has to be logged in the program for future mitigation in the program. It has to verify the user by comparing the database value for authentication. Main operation has to according to program flow such as create, insert, delete and search.

**2.4 Scope of the Application:**

1. The project is for enterprise or end user to manage their own information with authentication like that we can save our personal information.
2. It gives console application to the user to perform CRUD operation with document and manual report to understand the program.
3. **Conception:**
   1. **Database Structure:**

Derby database as backend to manage their user information by derby driver. Database table contains following values.

**3.1.1Database Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **SL.NO** | **Items** | **Type** | **CONSTRAINTS** |
| 1. | IDENTITYID | Varchar | NOT NULL |
| 2. | UID | varchar | Primary Key |
| 3. | DISPLAYNAME | Varchar | NULL |
| 4. | EMAILID | varchar | NULL |
| 5. | PASSWORD | varchar | NOT NULL |

**3.1.2Derby**

Derby is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. Derby is popular choice of database for use in web application and is an open source product.

* 1. **Global Application Flow:**

Main goal of the application flow is to user has to access his information and perform the CRUD operation in effective manner and has to differentiate the user between existing user and New user.

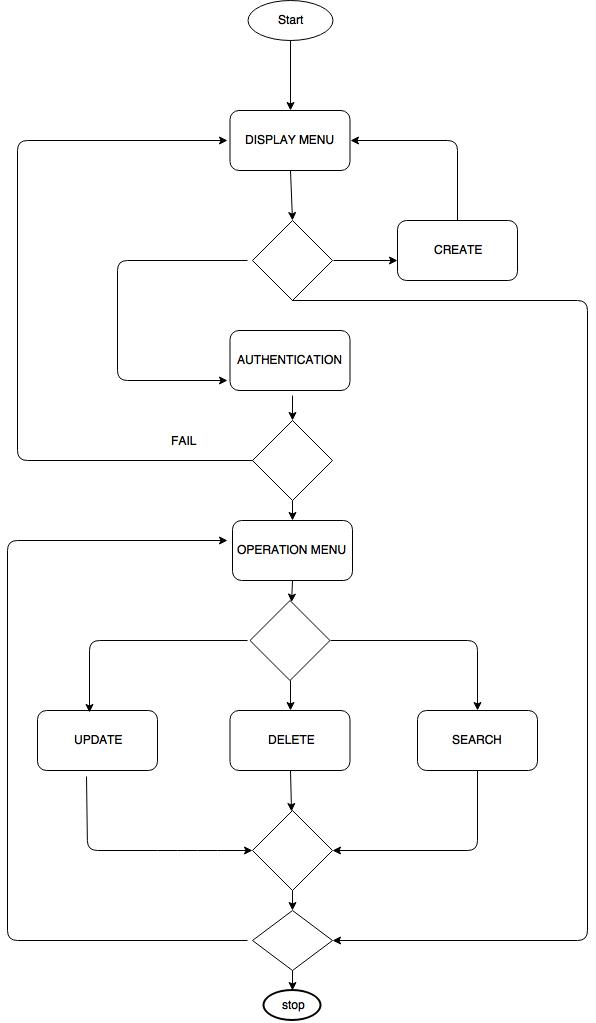
**3.2.1Existing User:**

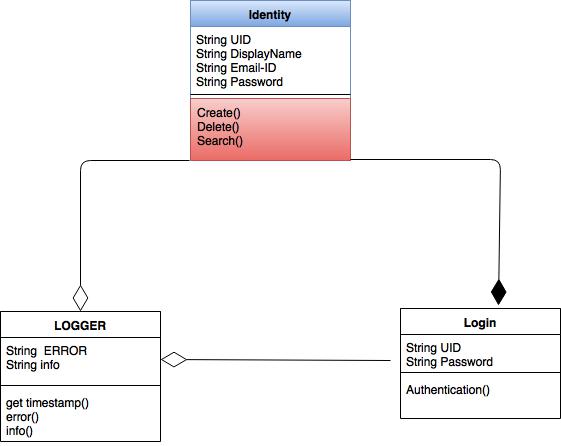
User enter into home page for authentication to access the profile if authentication fails it return to home page to perform login again in the program. If it success a user can perform the update, delete, search operation.

**3.2.2New User:**

when user successfully created in the program then it takes into authentication to verify the user to enter into the main operation in the program. It displays menu to guide the execution flow in the program.

**3.2.3. Flow Chart:**

****

**3.3 Global Schema:**

* Identity and Login class are strong aggregation if any user account deleted in identity class it affects in Login class.
* Identity class is main class in the program which consist method of CURD operation.
* Logger class is weak aggregation to store the action of identity class it will not reflect any action in identity class
* Login class consist authentication method with two objects.

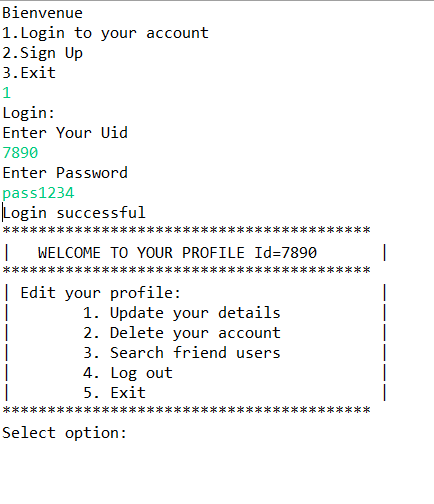
|  |  |
| --- | --- |
| **Program Variables** | **Data type** |
| UID | string |
| Email-id | string |
| display name | string |
| error | string |
| info | string |

1. **Console Operation Description:**

**4.1. User Authentication:**

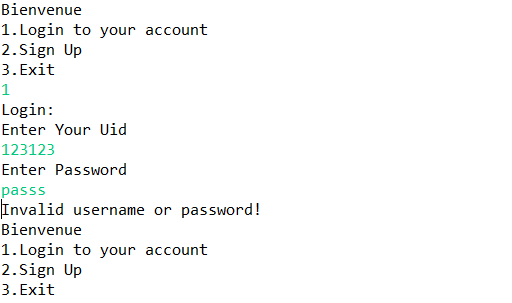
When user enter UID and password correctly following page menu will appear in console. If its failed it throws different types of errors in the program they are: Invalid user, password not matching.

**Success console**

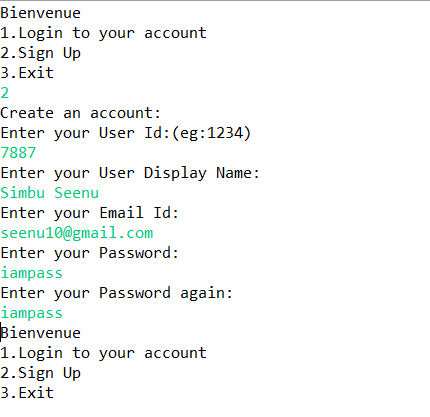
****

**`**

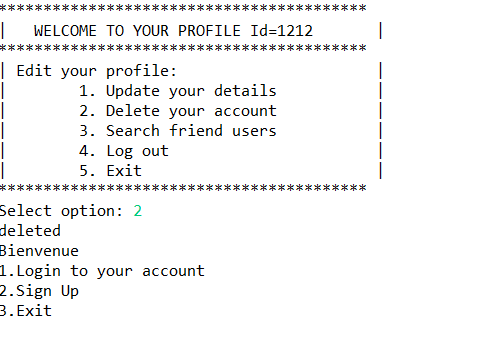
**Fail console**

****

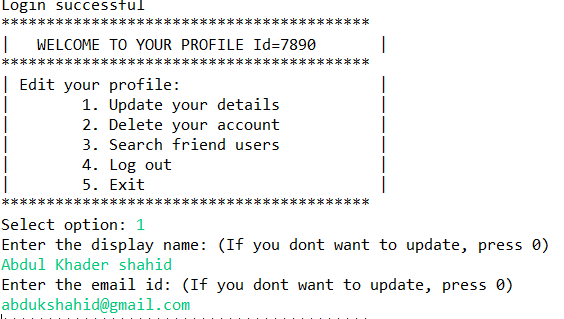
**4.2 User Creation:**

****

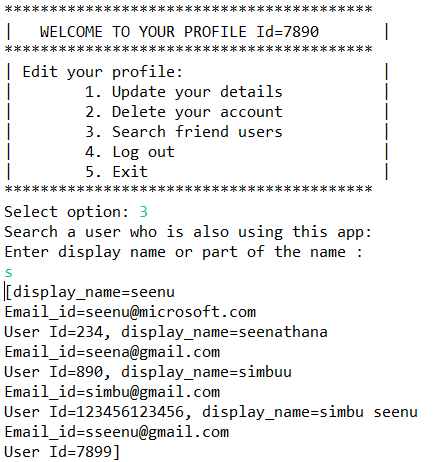
**4.3. User Delete:**

****

**4.4. User Update:**

****

**4.5. User Search:**

****

**5.Configuration Instruction:**

To user this project in client side following prerequisites are lists below.

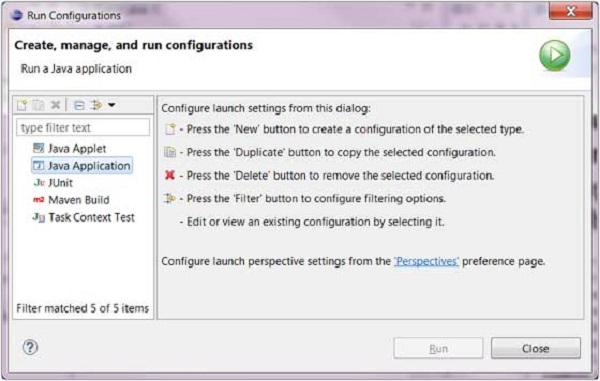
* Derby Database Configuration
* Eclipse (oxygen)

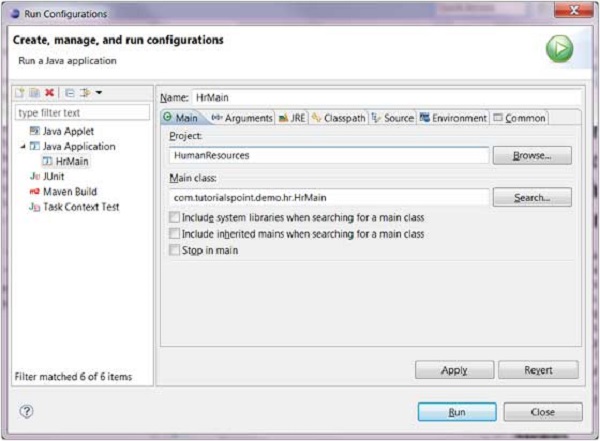
**5.1 Derby Database Configuration:**

* Create an [Eclipse Java project.](https://db.apache.org/derby/integrate/plugin_help/java_project.html)
* Add the [Apache Derby nature](https://db.apache.org/derby/integrate/plugin_help/nature.html) to your Java project.
* Start the [Derby Network Server.](https://db.apache.org/derby/integrate/plugin_help/start_server.html)
* Create a [Java application](https://db.apache.org/derby/integrate/plugin_help/derby_app.html) which accesses a Derby database.
* Shutdown the [Derby Network Server.](https://db.apache.org/derby/integrate/plugin_help/stop_server.html)
* Obtain Java and Derby information to help in troubleshooting problems using [sysinfo.](https://db.apache.org/derby/integrate/plugin_help/sysinfo.html)
* Learn more about Apache Derby from the online [documentation.](https://db.apache.org/derby/integrate/plugin_help/resources.html)

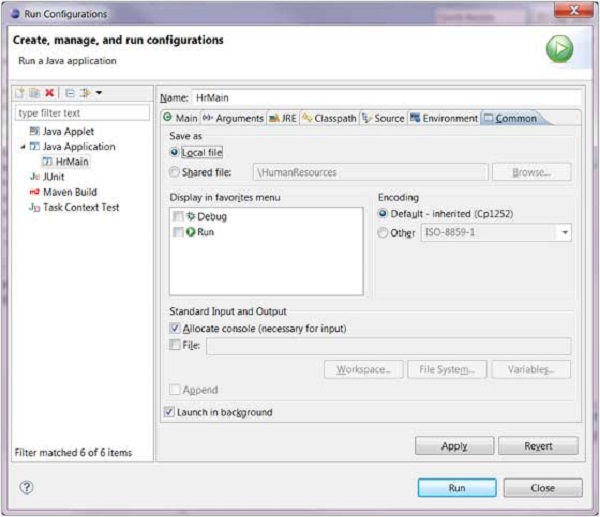
**5.2 Eclipse Run Configuration:**

The Run Configurations dialog allows you to create multiple run configurations. Each run configuration can start an application. The Run Configuration dialog can be invoked by selecting the Run Configurations menu item from the Run menu.





The Commons tab provides common options such as the ability to allocate a console for standard input and output.



To save the run configuration click on the Apply button and to launch the application click on the Run button.

1. **Bibliography:**
   1. **References:**

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

2. <https://github.com/thomasbroussard>.