EPITA INTERNATIONAL MASTERS ADVANCED JAVA

Professor: Thomas Broussard

TECHNICAL DOCUMENTATION

Gerard BARAKAT

Le Kremilin-Bicêtre, 2016

Table of Contents

1. Subject description

This application aims to manage identities of an Information System, by creating, reading, updating and deleting (CRUD). The end user will be able to log in the system with his own password, through a GUI, and make operations with them. All data will be stored in an MySQL Database. The Project will introduce the use of Maven, Spring, and Hibernate in JAVA.

Term	Description
CRUD	An acronym for the basic functions the system has to implement. (Create, Remove, Update, Delete).
GUI	Graphics User Interface
UML	Unified Modeling Language
End User	The user which will login the system.
Identity	The entity handled by the system.

2. Subject analysis

1. Major features

- Creation, Update, and Deletion of identity.
- Search Identity by any of its attributes
- Login/Password Authentication
- SHA256 Encryption for the users
- Creation of a new user

2. Application Feasibility

Generally, for studying the feasibility of an application, we use the "TELOS" approach, which is an acronym for: Technical, Economic, Legal, Operational, and Scheduling. As this is an educational project, we will not consider the Economic and Legal parts.

- Technical feasibility:

Regarding the knowledge acquired during the Java subject and the previous experience of the group members in professional projects, this project is considered feasible.

- Operational feasibility

A system, which consists in maintenance of Identities through a basic GUI, no special training or information is needed for the use of it. It have dynamic XML Based forms, and it can create new users to access the system.

- Schedule feasibility

Considering that the project requirements were provided since the starting of the java course, and the technical expertise learned in the class, the project was planned and executed on time with no delays. The schedule of the project is smooth and not over charged.

3. Data description

The data types being handled by the system is the *Identity* and *User*.

	Class Identity	**Class User**
Description	Main class being handled by the System	Class for handling Users that login to the Application
Attributes	private String uid; private String email; private String fname; private String lname; private Date birthDate;	private int id; private String username;

4. Expected results

Is expected that the End User will be able to Login the system and manage any Identity, through any of the operations and features provided by the system. The user can create a new identity, search for an existing identity, delete an existing identity, and edit an identity. More, the user if does not have an account, he/she can create an account using the registration for and the passwords are secured with SHA256 encryption in the database.

5. Algorithms study

Apache Maven is a build automation tool for Java projects. Think of Ant, or Make, but much more powerful and easier to use. If you've ever had to deal with building a Java project with dependencies or special build requirements then you've probably gone through the frustrations that Maven aims to eliminate. Using Maven is extremely easy, once you learn a few of the main concepts. Each project contains a file called a POM (Project Object Model), which is just an XML file containing details of the project. Some of these details might include project name, version, package type, dependencies, Maven plugins, etc.

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an alternative to, replacement for, or even addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is open source. The Spring Framework has its own Aspect-oriented programming (AOP) framework that modularizes cross-cutting concerns in aspects. The motivation for creating a separate AOP framework comes from the belief that it would be possible to provide basic AOP features without too much complexity in either design, implementation, or configuration. The Spring AOP framework also takes full advantage of the Spring container.

Hibernate ORM (Hibernate in short) is an object-relational mapping framework for the Java language. It provides a framework for mapping an object-oriented domain model to a relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct, persistent database accesses with high-level object handling functions. Hibernate's primary feature is mapping from Java classes to database tables; and mapping from Java data types to SQL data types. Hibernate also provides data query and retrieval facilities. It generates SQL calls and relieves the developer from manual handling and object conversion of the result set.

6. Scope of the application (limits, evolutions)

Is scope of the system:

- Login/Logout the system
- Create, Remove, Update, Delete and View an *Identity*
- Search an *Identity* through any of its attributes.
- Create users for the System
- Use of AJAX Calls to limit page loads

Negative Scope:

The following features are **NOT** implemented, and represent an opportunity of upgrade in the future.

- User creation activation via SMS or Email or Google Authenticator...

3. Conception

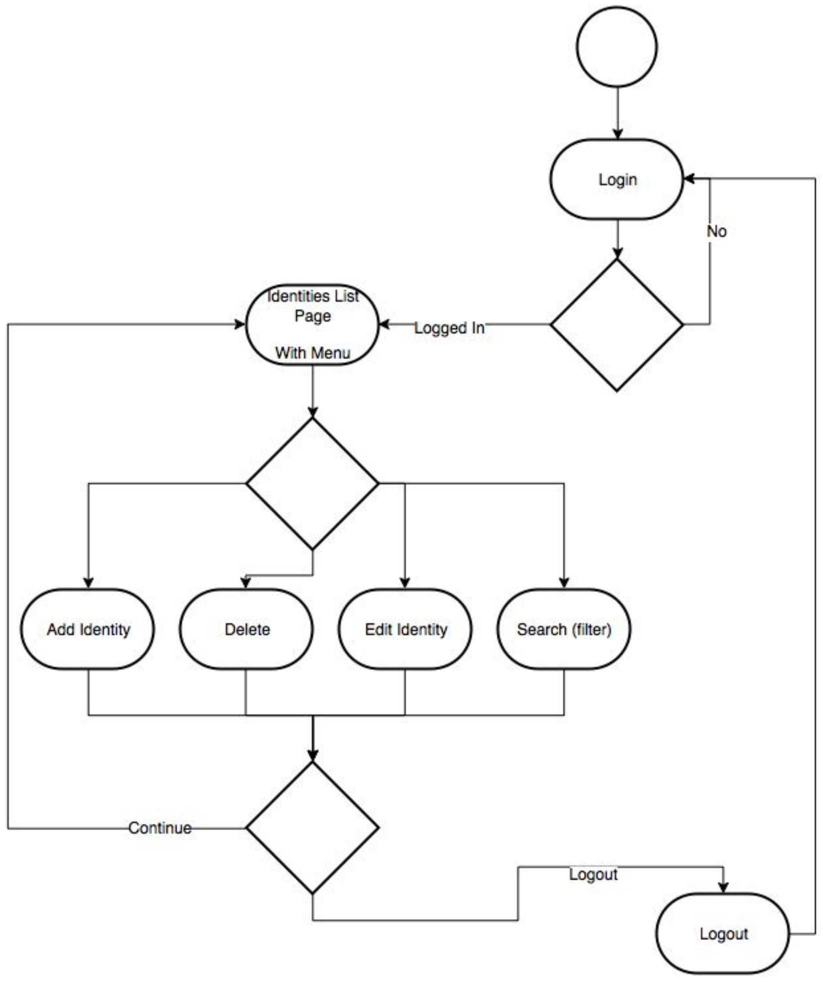
1. Chosen algorithm

We chose to use Hibernate to take care of the connection to the database using the Session Factory. An application has a single Session Factory instance and threads servicing client requests to obtain a session instance from the factory in order to make a transaction with the database (Commit a Query). Each transaction takes the criteria it needs to meet and executes the criteria. Hibernates configuration is set in the hibernate-configuration file and the database variables are in database.properties file.

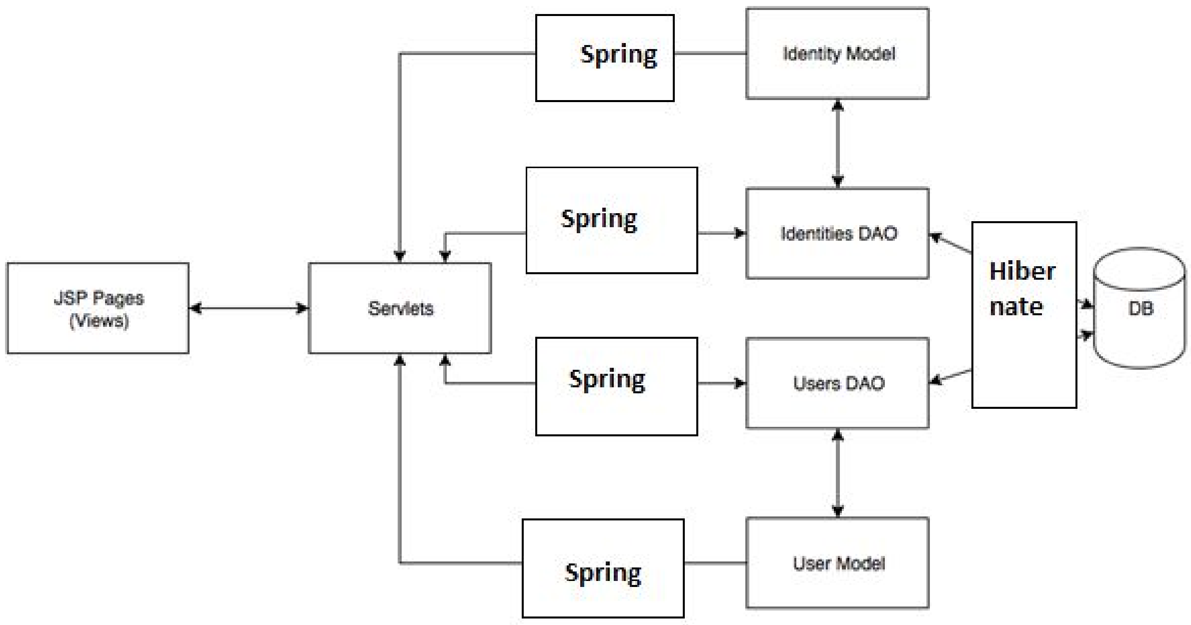
2. Data structures

For data structures, we used List and ArrayList. When we select records from the Database using the Hibernate in JAVA, we get a list. While iterating the list we create an indetity and put it in a List. In the servlet we Iterate that list to show the HTML of the view in the front-end of the application.

3. Global application flow



4. Global schema and major features schema



4. Console operations description

The user will navigate to the first page the login page of the system. If the user manages to login to the system with a valid email and password, he will be redirected to the *Identities* list page. In this page the user can add, edit, delete or filter an *Identity*.

If the user clicks on submit in the add form, and all the fields were filled, and identity is created and saved in the DB and the list of identities is reloaded.

If the user clicks on the delete of any of the *Identities* in the list, the *Identity* will be deleted and the table will be reloaded automatically.

If the user clicks on the edit button of any *Identity* in the list, the identity information will be loaded in the form of create identity, so the used can modify the entries and submit the data.

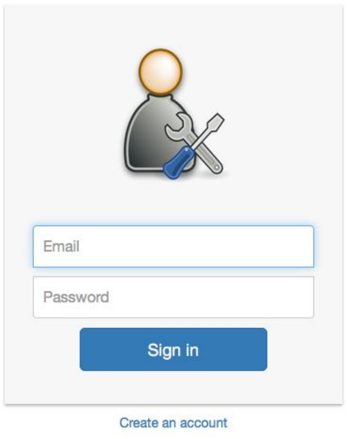
The search textbox, filters the *Identity* by its name, email or ID. The user has to enter the text in the textbox and click on search. If the textbox is empty, it will load all the identities.

5. Configuration instructions

1. Install WAMP (lamp on linux, Mamp on Mac)
2. Run WAMP (Lamp, XAMP, Mamp, etc…)
3. Install apache tomcat7 in eclipse
4. Open phpmyadmin (http://localhost/phpMyAdmin/)
5. Create a DB called “iam”
6. Import file “iam.sql” in the DB
7. Open Eclipse and import the project “IamWeb” “IamGroup” and “IamCore”
8. Click on Maven -> Install every project starting with the group
9. Start tomcat7
10. Run the file /WebContent/index.jsp
11. Login as “gerard”, passwd “123456” (as Demo and you can create your own username and password)

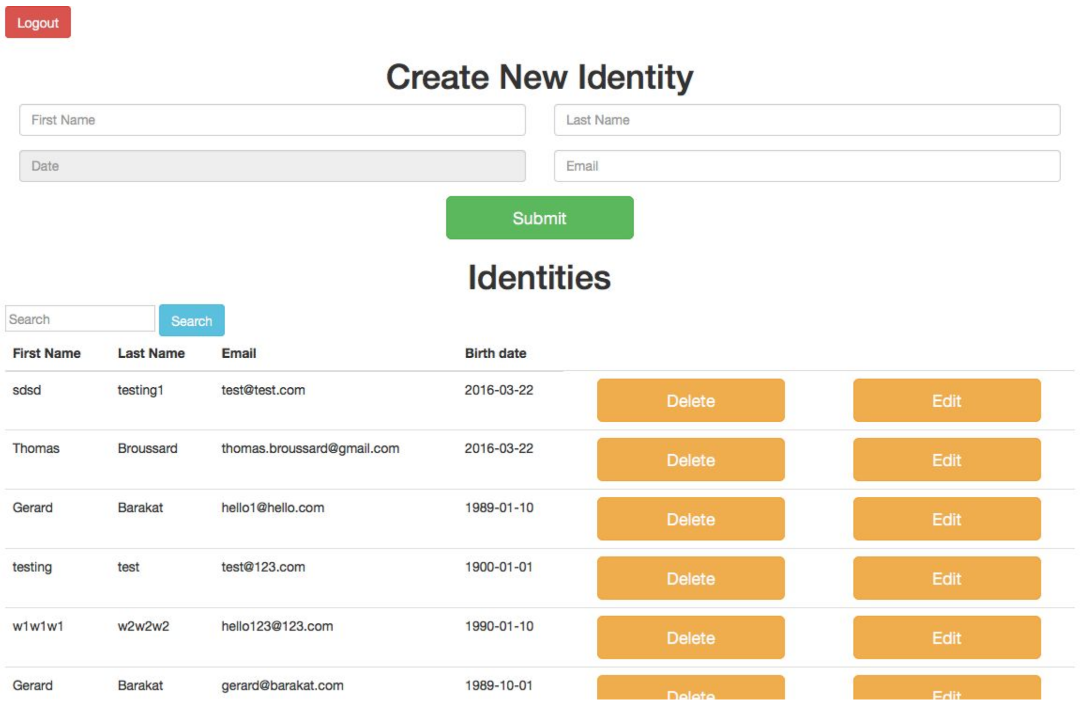
6. Commented Screenshots

1- Login Screen



This page lets the user login to the application using a username (email) and a password. If the user clicks on login with empty fields he will get an error. If the user enters a wrong username or password he will get an error, otherwise the user ID will be save in the session and he will be redirected to the identities list page. The create account button will redirect the user to the User Creation Page.

2- Identities List Page



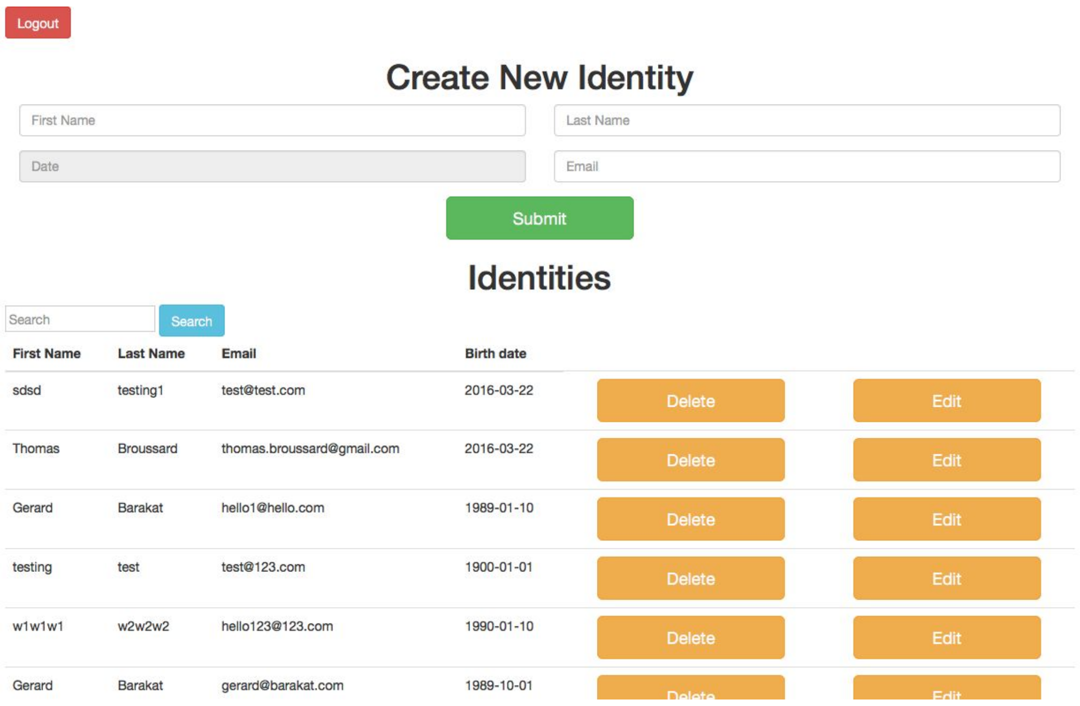
First Name	Last Name	Email	Birth date		
sdsd	testing1	test@test.com	2016-03-22	Delete	Edit
Thomas	Broussard	thomas.broussard@gmail.com	2016-03-22	Delete	Edit
Gerard	Barakat	hello1@hello.com	1989-01-10	Delete	Edit
testing	test	test@123.com	1900-01-01	Delete	Edit
w1w1w1	w2w2w2	hello123@123.com	1990-01-10	Delete	Edit
Gerard	Barakat	gerard@barakat.com	1989-10-01		

If the user clicks on submit in the add form, and all the fields were filled, and identity is created and saved in the DB and the list of identities is reloaded.

If the user clicks on the delete of any of the identities in the list, the identity will be deleted and the table will be reloaded automatically.

If the user clicks on the edit button of any *Identity* in the list, the identity information will be loaded in the form of create identity, so the used can modify the entries and submit the data.

3- Search Filter in Identities List Page



First Name	Last Name	Email	Birth date		
sdsd	testing1	test@test.com	2016-03-22	Delete	Edit
Thomas	Broussard	thomas.broussard@gmail.com	2016-03-22	Delete	Edit
Gerard	Barakat	hello1@hello.com	1989-01-10	Delete	Edit
testing	test	test@123.com	1900-01-01	Delete	Edit
w1w1w1	w2w2w2	hello123@123.com	1990-01-10	Delete	Edit
Gerard	Barakat	gerard@barakat.com	1989-10-01		

The search textbox, filters the identity by it’s name, email, user ID. The use is to enter the text in the textbox and clicking on search. If the textbox is empty, it will load all the identities.

7. Bibliography

UUID Class: https://docs.oracle.com/javase/7/docs/api/java/util/UUID.html

HttpSession: http://docs.oracle.com/javaee/5/api/javax/servlet/http/HttpSession.html

Maven:
https://docs.oracle.com/middleware/1212/core/MAVEN/weblogic_maven.htm#MAVEN8767

Spring:
http://www.oracle.com/technetwork/developer-tools/eclipse/springtutorial-087561.html

Hibernate:
https://docs.oracle.com/cd/E13226_01/workshop/docs92/studio33/hibernate-tutorial/0-Tutorial.htm