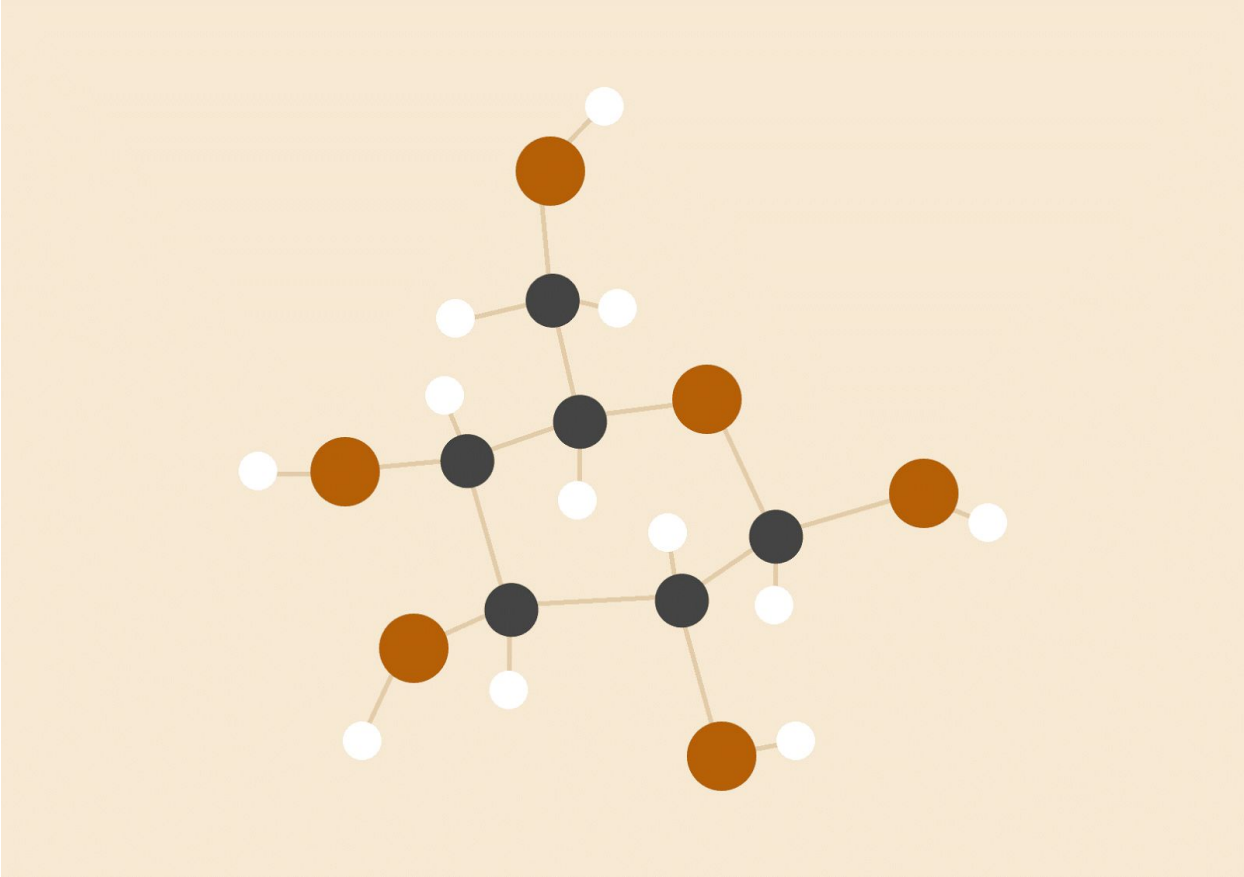
The IAM Web Application

IAM Project - Advanced Java



Lin LI

05.12.2016
EPITA International Masters
Advanced Java Students
Le Kremlin-Bicêtre, 94043

Technical Specification

Table of Contents

Subject Description

The project is developed using MVC based on JSP + Servlet + JavaBean. IAM web application allows users to login with username and password stored in usertable of MySQL database. Once logged in, users can display/create/update/delete identities stored in another table successfully to manage the information of identities in the identityTable of database.

Subject Analysis

Major Features

It's a web application where user can perform below functions:

- Create an identity
- Update an identity
- Delete an identity
- Display all the identities

To enforce security regarding user information and to avoid the access of user information by an unauthenticated personnel, we have implemented "User Authentication" concept where the user has to login as the first step which will enable the user to perform further steps related to identity management.

The user will have to authenticate him/her with a username and password stored in the usertable database.

Apache Build Manager Maven is used to make it possible to build the project in different environments.

Application Feasibility

The development of this application with the given features is quite feasible and hence, the application has been developed. Three methods are possible to create the same web application.

1. JSP+JavaBean

2. JSP+JavaBean+Servlet(MVC)
3. Struts+Spring+Hibernate(SSH)

In our project we are using the second method. Using MVC, makes the code cleaner and easier to maintain than the first method.

Data description

Below is the types of data we created to make this application achieve its given features:

UserTable:

Table of users who are authorized to log in to the identities management system.

Attributes: username, password

IdentityTable:

Table of the identites to be managed.

Attributes: uid(automatically incremented), firstName, lastName, email, sex, birthdate.

Expected results

When we run the web application on Tomcat server, we would be first directed to the login page. Wrong username/password, would keep users stay in the login page.

Successfully logging in would take user to the main page, where on the left side users can find menus of features that can be operated on the identitytable, eg. create/delete/update/display identities.

Logging out would take user back to the login page.

Algorithms study

Three algorithms can be used to complete this project:

1. JSP+JavaBean
2. JSP+JavaBean+Servlet(MVC)
3. Struts+Spring+Hibernate(SSH)

Scope of the application (limits, evolutions)

Limits:

Using JDBC, the program can get complex if the scale of the project gets bigger.

Evolutions:

Using SSH - Struts, Spring, Hibernate Structure instead would be more practical and cleaner in case the project gets bigger.

Conception

Chosen algorithm

JSP+JavaBean+Servlet (MVC)

Data structures

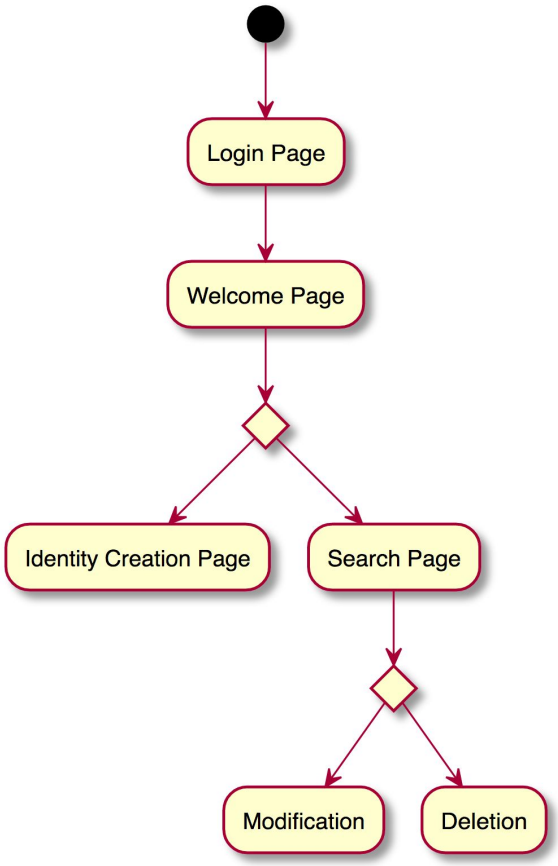
JSPs: The View of our algorithm. Deals with the front-end where users interacts with the system.

Servlets: Controller of the algorithm. It gets requests from Views and delegate the requests to corresponding Services, and then send the response back to the Views.

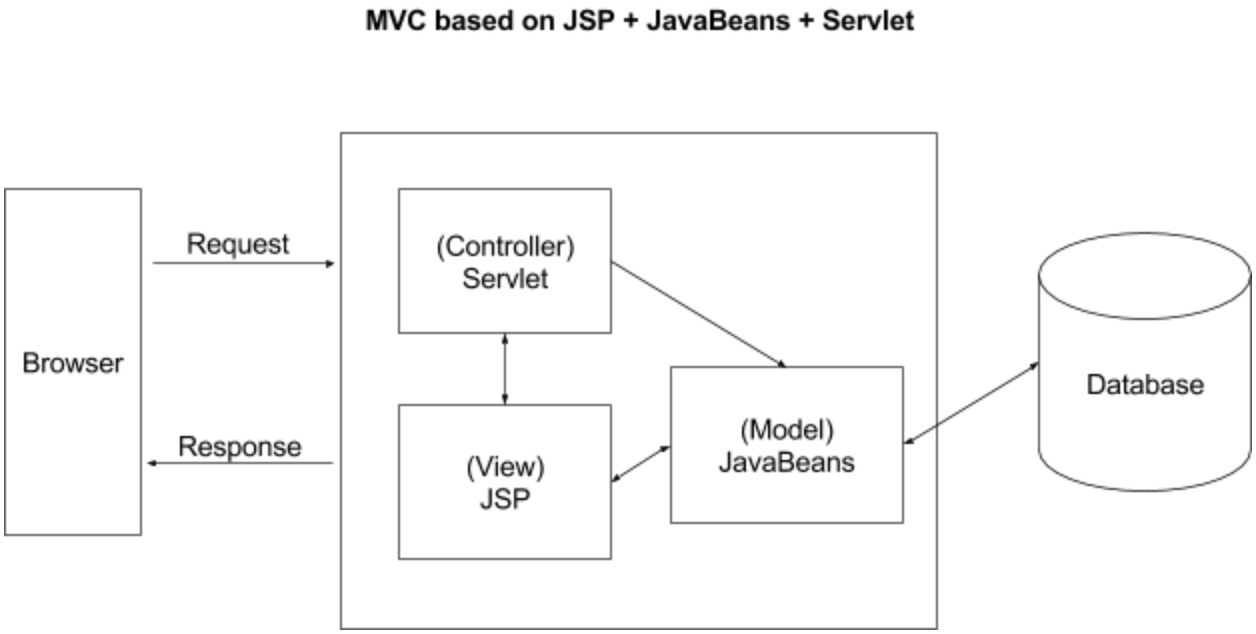
Services: Deals with the communication between the database and the system. Send user requests and get results from database, in this case, MySQL.

Datamodels: Data is encapsulated in JavaBeans to send requests and responses in between server and databases.

Global application flow



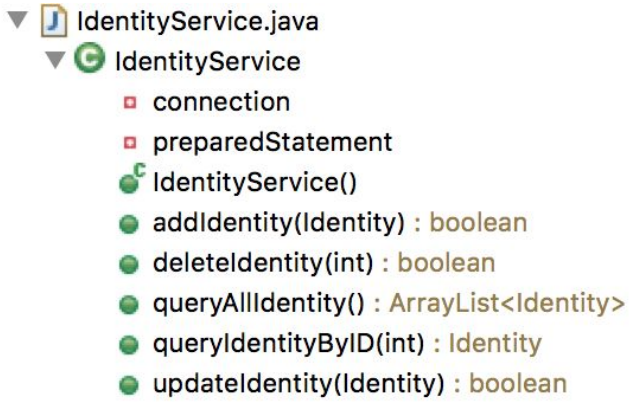
Global schema and major features schema



- Identity.java
  - Identity
    - birthDate
    - email
    - firstName
    - lastName
    - sex
    - UID
    - getBirthDate() : String
    - getEmail() : String
    - getFirstName() : String
    - getLastName() : String
    - getSex() : byte
    - getUID() : int
    - setBirthDate(String) : void
    - setEmail(String) : void
    - setFirstName(String) : void
    - setLastName(String) : void
    - setSex(byte) : void
    - setUID(int) : void

Identity class with getters and setters of all the table attributes from database.

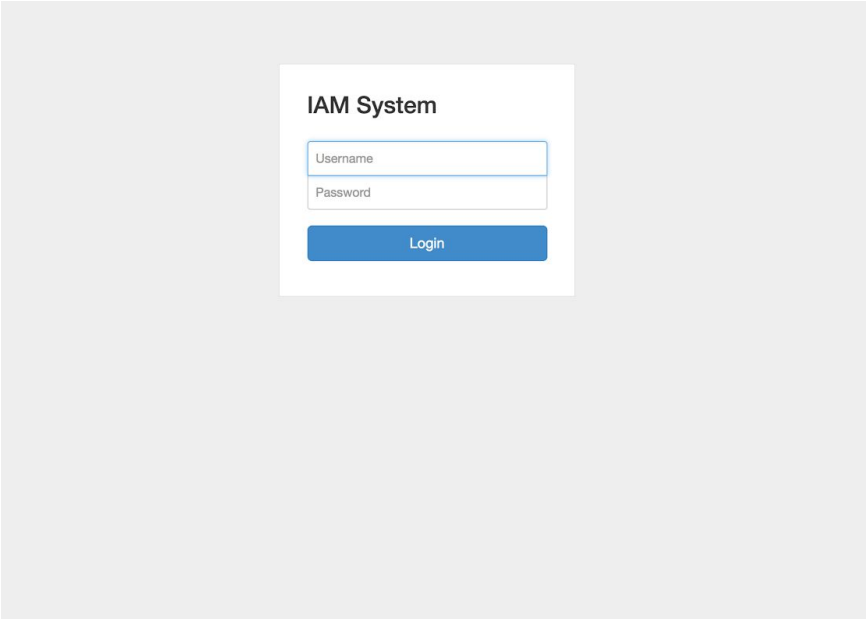
Usertable class with setters and getters of all the table attributes from database.



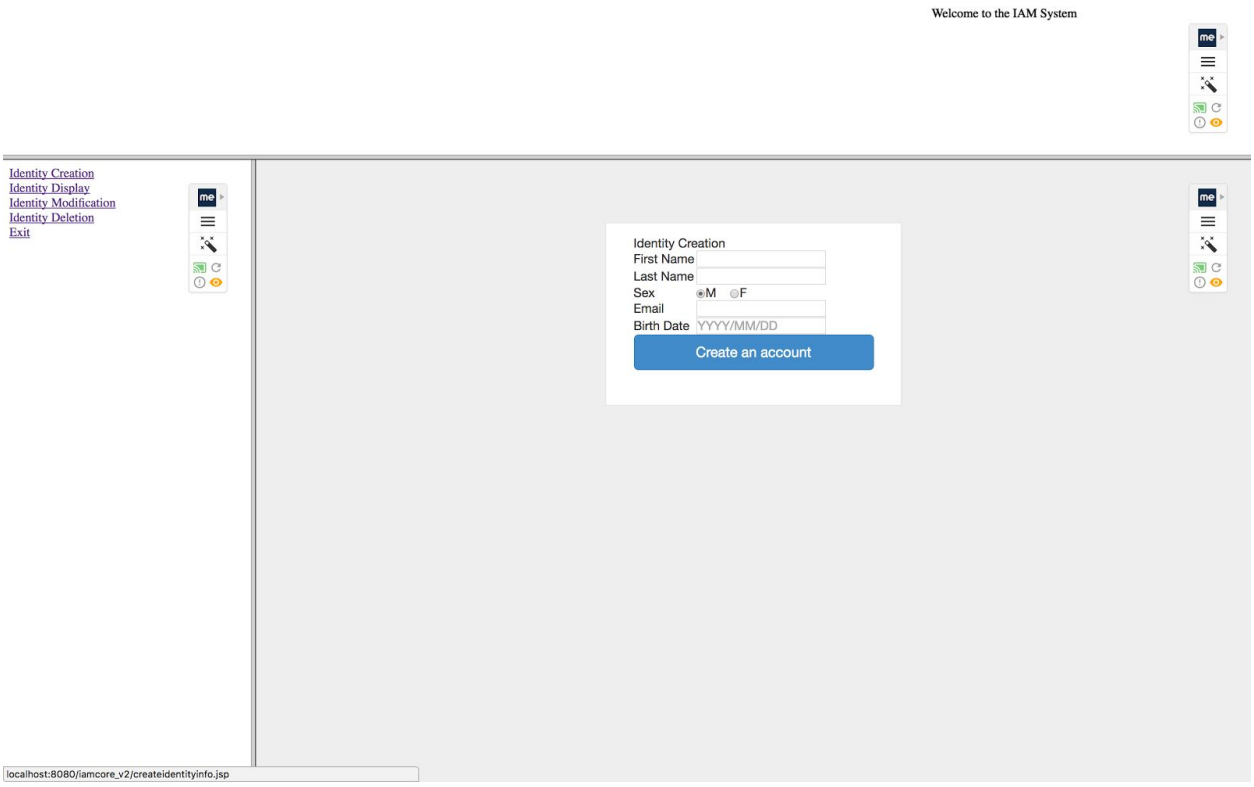
Service class for Identity, which deal with communication with the operations on the identitytable from requests of users.

Service class to deal with requests from users to communicate with usertable database.

GUI description

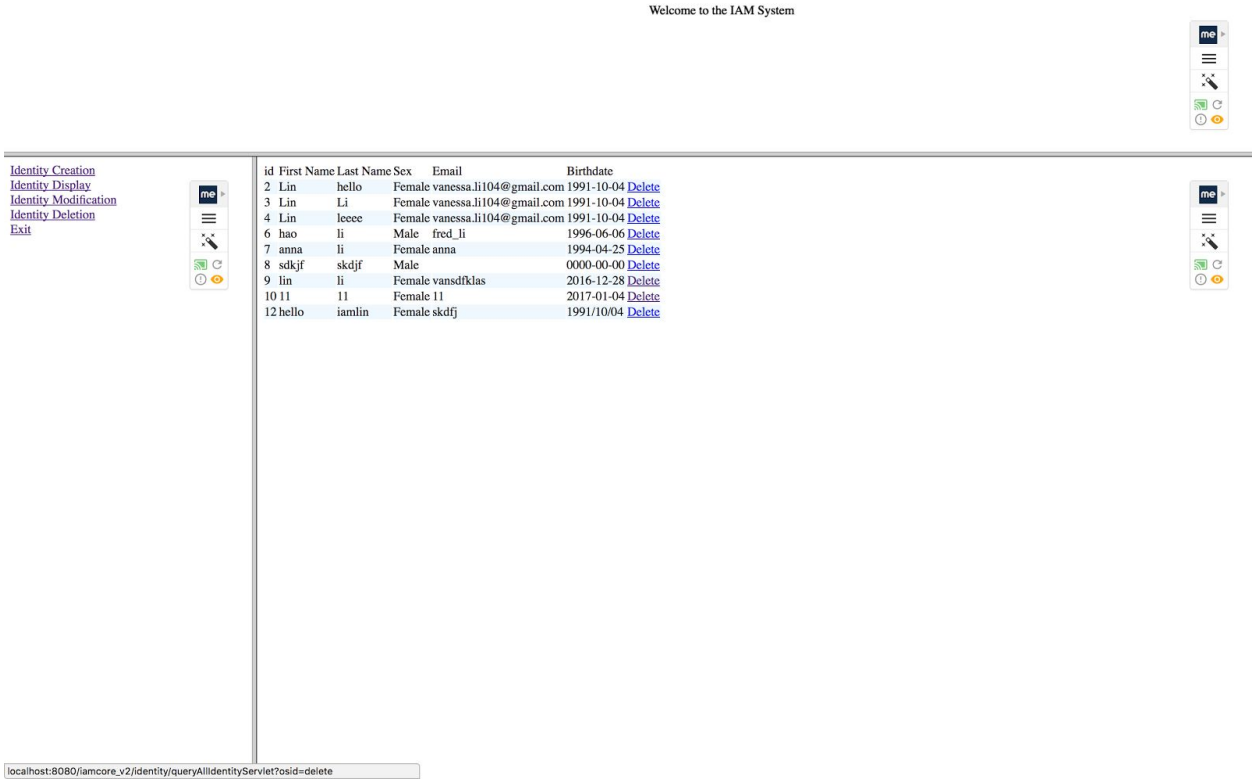


Index.jsp: Login page

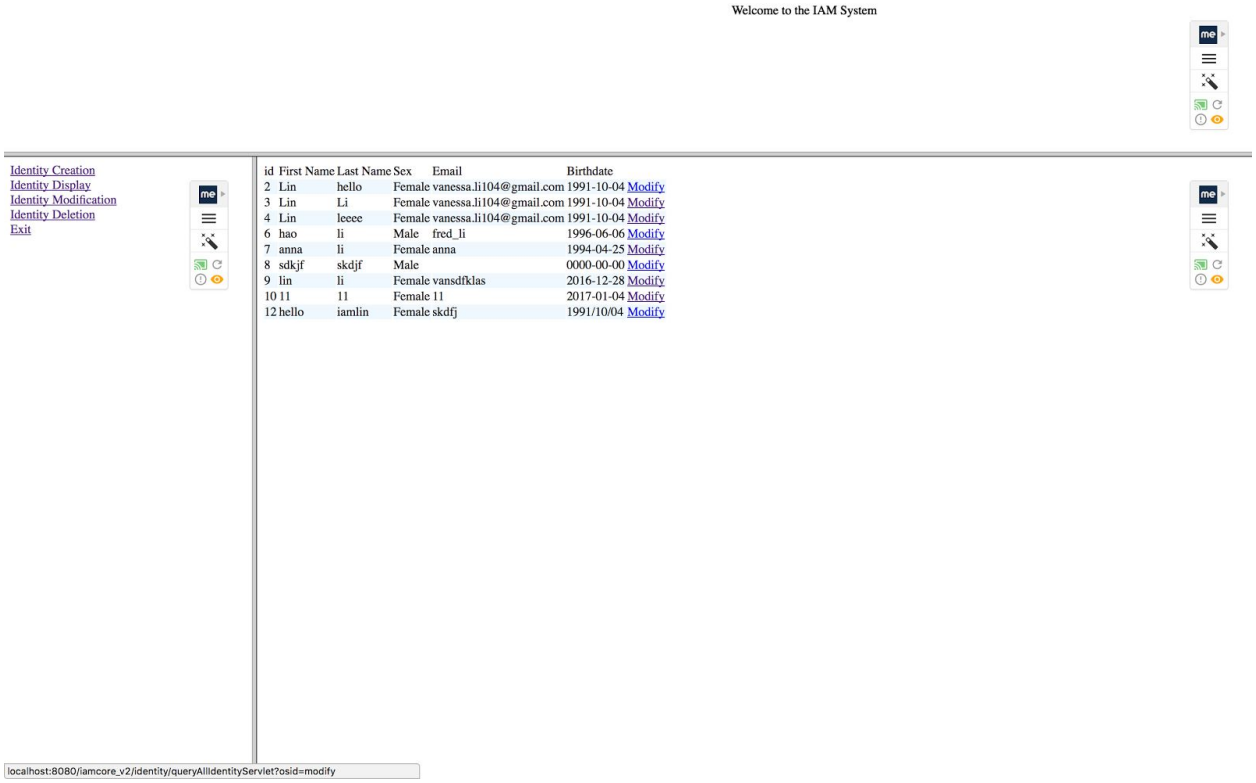


createidentityinfo.jsp: identity creation page

displayidentityinfo.jsp: Identity Display page



deleteidentityinfo.jsp: Identity deletion page



Modifyidentityinfo.jsp: Identity Modification page

modifyOneIdentity.jsp: Modification of one identity page

Configuration instructions

Using maven build manager, we could implement this project easily on any environments. But below is the list of systems which were used to develop and test the application:

IDE: MyEclipse 2016 CI

Tomcat: Apache Tomcat 8.5.6

MySQL: MySQL server 5.0 or higher

Bibliography

1. Stack Overflow - Using beans in servlets: https://goo.gl/o6bv0R
2. Mkyong - How to create a web application project with maven: https://goo.gl/4jCYFX