

Subject Description

For the final project in Advanced Java, the application must perform the basic CRUD (Create, Read, Update, Delete) operations on a simple database, including frameworks we've seen throughout the semester.

Subject Analysis

Major Features

The application must allow the user to maintain, through a web servlet, a contact database with every entity being comprised of a first name, last name and a phone number. Additionally it should include a layer of security for a login process that will filter and hide pages not viewable by anonymous users.

Application Feasibility

The application has to be viewed through a web browser. The way it is to be configured would allow for it to be deployed on to a server with full capabilities of being accessible from within the network, or even online, depending on the server configuration.

Data Description

The application will use a MySQL database with a single Identity table. This table will have the following fields:

- id - An automatically generated field from within Spring and Hibernate
- firstName - String
- lastName - String
- phone - String

Additionally, the usernames and passwords were unfortunately required to be hardcoded into the security layer xml, so any additional users required should be added there.

Expected Results

The application will be done in a single web page with some jQuery functionality for ease of use and aesthetics. Therefore, the user will only have two pages in which to function: the login page, in which to input their username and password for authentication purposes; and the contacts page, where the user will be able to view, add, edit and delete the contacts from the database.

Algorithm Study

The application uses a simple MVC paradigm with a Spring Session Factory and a Spring Security Layer. When the user clicks on a command, the view sends a request to the model (through a jsp servlet). This model then turns its attention to the database in order to produce the data requested. The data is then sent to the view where it is displayed by the jsp to the user.

Scope of the Application

The application will do no more than the basic CRUD operations. Additionally the application does not currently perform checks to verify that the data is correct. For example, the database stores

only VARCHAR(45) for its first name, last name and phone columns. If it goes beyond that the data might get truncated or an error code will result.

However, the way it's set up, it's entirely possible to create additional methods for checking data

IAMWeb

Logout

Contact List

First Name	Last Name	Phone		
Kate	Beckett	419-555-9876	Edit	Delete
Peggy	Carter	419-555-8126	Edit	Delete
Harley	Quinn	419-555-2313	Edit	Delete
Nanna	Bryndis	419-555-9881	Edit	Delete
Marina	Diamondis	419-555-2773	Edit	Delete
Diana	Prince	419-555-7378	Edit	Delete
Jenny	Lewis	419-555-3333	Edit	Delete
Kara	Starbuck	419-555-0299	Edit	Delete

+ New User

Peggy	Carter	419-555-8126	Edit	Delete
Harley	Quinn	419-555-2313	Edit	Delete
Nanna	Bryndis	419-555-9881	Edit	Delete
Marina	Diamondis	419-555-2773	Edit	Delete
Diana	Prince	419-555-7378	Edit	Delete
Jenny	Lewis	419-555-3333	Edit	Delete
Kara	Starbuck	419-555-0299	Edit	Delete

+ New User

integrity as well as additional functionalities such as a search function. Everything is wired appropriately for a second version of the application.

Conception

Chosen Algorithm

The MVC model was chosen specifically because it is the simplest way to generate highly reusable code with minimal effort. A method called ModelAndView is invoked to generate the views every time one is required.

Data Structures

GUI Description

Login Page

The login page consists of a page where the user can include his username and password with a button to Sign In.

Contacts Page

The Contacts page is where everything happens.

Starting from the top we have a Logout button on the navbar to go back to the login page and kill the authentication.

Each row has an Edit and Delete button. The Edit button will generate an editable row that the user can modify to fit his needs. The Delete button removes the row and the data from the database.

At the bottom there is a New User button that the user can click to reveal a form that will allow for the inclusion of a new row in the database.

Configuration Instructions

In order for the application to work you must run the contact_manager.sql in a MySQL database for the creation of the database, tables and test data.

The configuration of the database is included within the /src/main/java/webapp/WEB-INF/classes/database.properties file and is included again here:

```
database.driver=com.mysql.jdbc.Driver
database.url=jdbc:mysql://localhost/contact_manager
database.user=root
database.password=root
hibernate.show_sql=true
```