Create a simple Java web application. In this application there should be a page where a user can register an account. The information required for an account should include an email address (this will be the users login name), password and some additional basic fields (your choice, look at some popular online services). The forms in the application should include fields validation (notify users of mistakes made). Check for duplicate email address when registering an account. After a user has registered, send the user an email message congratulating him/her on successful registration.

Create a login form where a user can logon to the system. This form should also include a link for a user that has forgotten his/her password. The forgotten password form should allow a user to receive a password change email that will include a link. By pressing the link the user can change the password.

The user should be able to logout from the system.

The main function of the application is the following: the user may manage lists of ToDo items. It is up to you how you choose to implement it. It should be very basic, nothing complicated.

Try to separate java classes into logical packages, for example: controller, entity, service, util, validator.

When sending emails make use of Spring framework's JavaMail supporting classes. SMTP settings should be in a configuration file that we can change before building a WAR archive.

Use of AJAX technology is not required, simple request / response model is sufficient. However, you can use it if you wish.

The SQL schema should be in a SQL text file, in the project directory (see if you can find a good place for it). The schema will be executed by us against an empty database.

Make use of the following technologies:

\*) Apache Maven - you can use the webapp archetype as a basis for the project.

\*) Servlet container: Tomcat 7.

\*) Database: PostgreSQL.

\*) Hibernate ORM for data persistence. Use JPA annotations when mapping entities (don't use XML mapping).

\*) Spring framework and Spring MVC. Optionally you can also use Spring Data JPA if you're able to.

\*) Use Java Server Pages (JSP) as a view technology.

The result of the work should be a Maven project that can be built, this will produce a WAR archive. We should be able to deploy this WAR archive to a Tomcat servlet container. The servlet container will run on Linux, so supporting other platforms is not necessary. Don't include a JDBC driver with the WAR archive, our servlet container will have it in the lib directory.

You must use any Java version starting from 6 (1.6) and Bootstrap 3 (http://getbootstrap.com/).