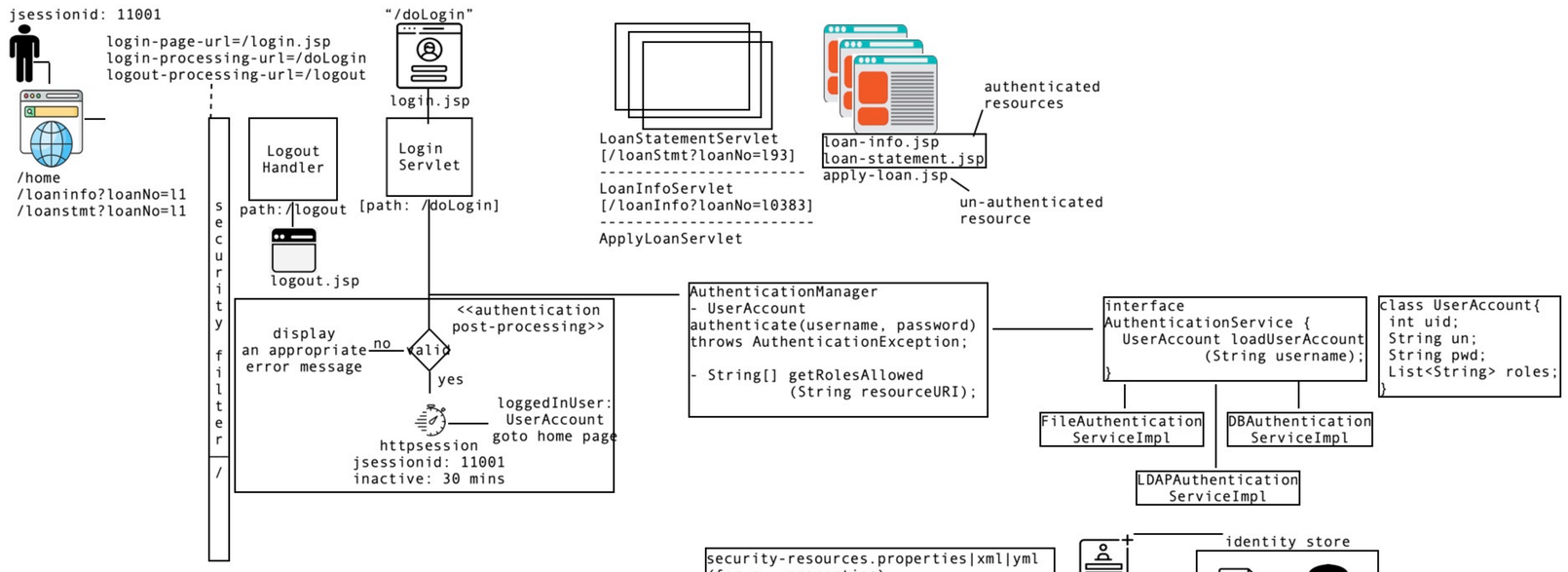

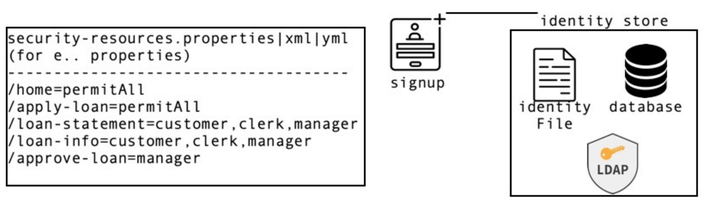

Security Filter:
every request will be receiver by the SecurityFilter because of url-pattern "/". upon receiving the request, the filter checks for
1. is the resource is authenticated resource or not
requestURL - should be lookedup into security-resources.properties to determine which type of resource it is.

if the resource is public or un-authenticated resource "permitAll", then the security filter can allow the resource to be directly accessing without login/authentication

2. if the resource is authenticated resource and a specific role of user is only allowed to access, then
2.1 check whether the current user is logged-in or not
  - if the user is not logged in, the redirect the user to login page
2.2 if user is already logged-in to the system, then
  - get the loggedInUser info from session object
  - now take the roles of the user and verify whether the user poses the role for accessing the resource or not
  2.2.1 if the user poses the right role, grant the access to the resource just by calling doFilter(req, resp);
  2.2.2 if the user doesnt poses the required role in access the resource throw an exception indicating he is not authorized in accessing the resource by reporting ForbiddenException. HTTP:403 error

AuthenticationException
|-CredentialsMisMatchException
|-AccountLockedException
|-AccountNotActiveException
|-AccountDisabledException

What information will be stored aspart of the identity store?
We store the identity of the user along with the roles of the user.
For eg.. if the identity store is a database, then we capture the it in db tables as below

roles
role_id role_cd

users
user_id username password status

A = active
R = registered
L = locked
D = disabled/deleted

user_roles
user_id role_id

depends on the application requirement, we might choose to have a user in only #1 role or in multiple roles

LDAP stands for light weight directory access protocol, similar to how we have tomcat/weblogic servers for deploying and exposing the application.
LDAP is also an directory server in which we store identity information of the users of the organization centrally.

What are the challenges in implementing custom security in securing an web application?
1. we need to pour lot of time in designing and architecting the system inorder to implement security
2. we need to write lot of code and complexity involved in implementing the security is very high and takes huge amount of time
3. proper testing needs to be done inorder to ensure there are no security breaches in allowing the access to the application. We need specialized security test engineers who can understand the security aspects/technics inorder to test and certify the application from security standpoint of view. this requires huge amount of time in testing and cost of certifying the application is very high