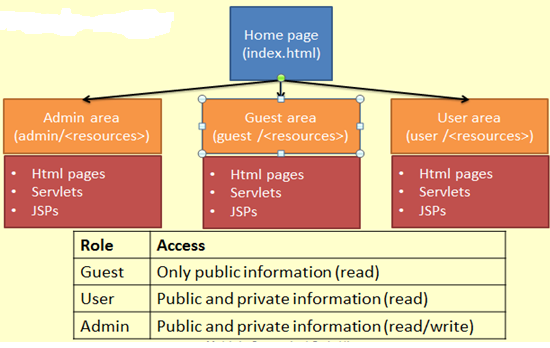
**Online Cash Register System Administration**

**1 Description**

Add authentication-authorization and encryption to our web application we built in assignment II, a self-checkout system for an online store called “eShop” using Eclipse. You need to make modifications on this web app in this assignment.

**2 Mandatory Requirements**

**2.1** Your website should be divided into three parts (3 folders: admin,user, and guest) :



* Only user and admin roles has admin privilege can access all the webpages under admins folder and move the following part in your assignment2 to the admin folder:

*Add a new HTML page called Admin.html, in which the system admin can add new items to the catalogue. User should enter the name, code, and the price of a new item as well as if the product taxable or not. Your system should validate the information (all fields are mandatory and the price should be a number) and add the new item to the catalogue. Obviously, you cannot have two items with the same code.*

* Only user has user privilege that can access all the webpages under user folder

User folder should has all the check-out pages…

* Only admin user has admin privilege that can modify all the webpages under admin and

user folders…

* Guest can only view welcome.html and index.html pages on the website.

**2.2 Database**

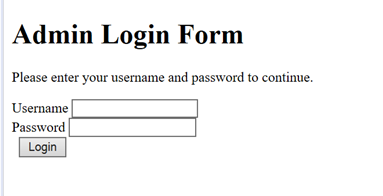
* + Use the following set of SQL statements to create the tables for login authentication using **JDBCRealm**, configure our web server to use JDBCRealm for the authentication.

|  |
| --- |
| **create table users**  **(**  **user\_name varchar(15) not null primary key,**  **user\_pass varchar(200) not null**  **);** |
| **create table user\_roles**  **(**  **user\_name varchar(15) not null,**  **role\_name varchar(15) not null,**  **primary key( user\_name, role\_name )**  **);** |
| **create table roles**  **(**  **role\_name varchar(15) not null primary key**  **);** |

**2.3 Login page**

You need to create the authentication login page using FORM-based authentication, provide the error-login.html which will pop up when login incorrect.

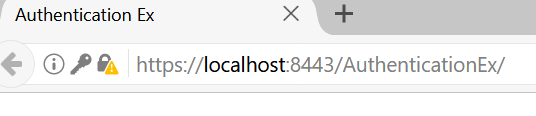
Example:



**2.4 Using TLS/SSL to run our web application**

Configure our application to use secure communication, configure the server to support TLS/SSL using port 8443 by a self-signed certificate using tomcat keytool.

Example of run web app using TLS/SSL protocol (later on top of HTTP protocol) on localhost:8443:



* 1. **submission requirement:**

**Zip the entire project and a Word file that has your servers.xml, database tables snapshots for the JDBC login tables and tomcat-users.xml and demo your work to the class and me in class.**

**Marking scheme:**

**Each section in part 2 is 10 marks, the total is 50 out of 10% of the course weight.**