Gyakorlat 4 óra feladatok

**JSF alapok**

1. Importáljuk a lab\_4 projektet (Import/Existing Maven project/Mappa kiválasztása, ahol a pom file található).
2. Állítsuk be a web.xml-ben a diákon található minta alapján, hogy az alkalmazás használja a FacesServlet-et.
3. Jelenítsük meg a felhasználók szerepkörét is a felhasználókat listázó oldalon. Ehhez a customer/view.xhtml oldal **h:datatable** tag-jét kell szerkesztenünk. Vizsgáljuk meg, hogy a többi mező (vezetéknév, keresztnév, felhasználónév), hogyan van megjelenítve, és a minta alapján jelenítsük meg a szerepkört is.
4. Egészítsük ki a felhasználó létrehozása oldalt (customer/create.xhtml), hogy jelszót és szerepkört is meg lehessen adni. A jelszó mezőt a többi szövegbeviteli mező alapján készíthetjük el (keresztnév, vezetéknév), annyi különbséggel, hogy a h:inputText beviteli mező helyett a **h:inputSecret** mezőt használjuk.

A szerepkör mezőnél jelenleg egy értékből választhatunk. Egészítsük ki két további értékkel:

vendég (GUEST), illetve vevő (CUSTOMER).

1. Végezzük el a 4.-es pontban leírtakat a felhasználó szerkesztése oldalon is (customer/edit.xhtml).
2. A felhasználó létrehozása oldalon a Mentés gomb jelenleg nem csinál semmit. Kössük be ide a customerBean#addCustomer() metódust úgy, hogy a gomb megnyomása utána visszairányítson minket a felhasználók listája oldalra.
3. Kössük be a felhasználó szerkesztése oldalon a mentés gombra a customerBean#updateCustomer() metódust.

**Template-elés, komponensek építése, validátorok**

1. A resources/templates/menu.xhtml file-ba kiemeltük a menü kódját. Használjuk ezt a menüt a pageTemplate.xhtml-ben az ott beégetett menü helyett (ui:insert).
2. A order/view.xhtml oldalnál állítsuk be, hogy a resources/templates/pageTemplate.xhtml-t használja template-ként.
3. A felhasználó létrehozása és módosítását oldalak ugyanazokat a módosítandó mezőket tartalmazták. Ezeket összevontuk egy közös komponensbe (editCustomerComp.xhtml). Ezt a komponenst fogjuk használni a customer/edit.xhtml illetve a customer/create.xhtml oldalakon.

* Adjuk hozzá a komponenst a két oldalhoz, és töltsük ki a kötelező paramétereket a customerBean megfelelő mezőivel.
* Amíg nem adjuk hozzá az oldalhoz a megfelelő namespace-t, nem ismeri föl a komponenst. Adjuk hozzá a namespace-t , amely a komponensünket tartalmazza(órai diák alapján).

1. Szerkesszük meg az editCustomerComp komponens beviteli mezőit, hogy felhasználónév és jelszó mező kitöltése kötelező legyen. Ehhez használjuk az f:validateRequired beépített validátort.
2. Jelenleg hiányzik a hibaüzenetet megjelenítő mező a jelszó komponenshez. Adjuk hozzá a h:message mezőt, amely a jelszó beviteli mezőhöz kapcsolódó hibákat jeleníti meg.
3. Egészítsük ki a PasswordValidator osztályt:

* Adjuk hozzá a megfelelő annotációt, hogy valóban validátorként lehessen használni.
* Implementáljuk a validate metódust. Ha a megadott jelszó rövidebb, mint 6 karakter vagy hosszabb, mint 12 karakter, akkor a következő hibaüzenetet jelenítse meg: a jelszó hossza nem megfelelő.
* Helyezzük el a validátort a jelszó mezőn.