sstProje2

Java Server Faces ile Stajyer Arama

Ege Üniversitesi Bilgisayar Mühendisliği
Sunucu Yazılım Teknolojileri
Proje-2

Umut BENZER 05-06-7670 http://www.ubenzer.com/

İçindekiler

Tasarım Aşaması

Genel Tanıtım

Bu projede Java Server Faces teknolojisi kullanılarak basit bir stajyer öğrenci ekleme / arama / seçme sitesi geliştirilmiştir.

Bu proje diğer yaptığım birçok projenin aksine, gerçekten yayımlanacak kadar gelişmiş bir proje değildir. Proje yapılırken JSF teknolojisini öğrenmek amaç edinilmiş olup, kullanıma sokulabilecek kadar özellik eklenmemiştir.

Proje sadece proje metninde bizden istenilenleri gerçekleştirmekte olup, gerek arayüzü gerek altyapısı ile çok temel özellikler ve sadece istenen özellikleri sunmaktadır. Bunun temel nedeni olarak tez konusunda yoğunlaşmam gösterilebilir.

Platform ve Dil

Proje Eclipse 3.6 Helios ortamında geliştirilmiştir. Bu proje üç yıl aradan sonra tekrar Eclipse'i kullanmaya geri dönmem ile yaptığım ilk proje olup, aynı zamanda IDE'yi öğrenme aşamasını da kapsamıştır.

Proje Apache Tomcat 7.0 sucusunda çalıştırılmıştır ancak teknik olarak diğer destekleyen sunucularda da gerekli konfigürasyonlar yapılarak çalıştırılabilir.

Projede JSF 2.0 ve Facelet'ler kullanılmıştır.

Raporun Kapsamı

Rapor proje hakkında teknik bilgiler ve çalışan halinin tanıtımını içermektedir.

Teknik Bilgiler

MVC

Bu projede mümkün ölçüde MVC Design Pattern kullanılmaya çalışılmıştır. Paketler ve sınıfların işleyişi bunlara göre düzenlenmiş olup, logic view ve model birbirinden ayrılmıştır. Projenin küçüklüğü nedeni ile domaine ait, ancak veri katamanı ile alakalı olmayan nesneler (domain) ve altyapı nesneleri (utility) model kısmıyla birlikte tasarlanmıştır.

Paketler ve Sınıflar

Projede toplam iki adet paket kullanılmıştır. Bunar yanında .xhtml ve .jsp uzantılı kısımlar *(Web Content içeriği)* yapısı gereği üçüncü bir paket olarak düşünülebilir.

Paketler ve paketlerde bulunan sınıflar aşağıda belirtilmiştir. Sınıfların ne işe yaradığı ve metotlarının neler yaptığı konusundaki ayrıntılı bilgiler JavaDoc olarak kaynak kodu ile beraber sunulduğundan buraya bir kopyası daha eklenmemiştir.

com.ubenzer.sst.proje2.beans

Bu paket MVC patterninin "Controller" kısmına denk gelmektedir. Buradaki sınıfların temel amacı kullanıcıdan (arayüzden, istemciden) gelen bilgilerin alınmasını ve istenen süre boyunca (request, session, application) tutulmasını sağlamaktır.

OgrenciBean

Bu sınıf, sitenin öğrenciler ile ilgili kısımlarından gelen bilgilerin tutulmasından sorumludur. Kısacası, sitenin "öğrenci ekleme/düzenleme/login" sayfalarından biri kullanılıyorsa, bu sayfalardan alınan bilgiler bu bean aracılığı ile sisteme ulaşmaktadır.

Öğrenci bilgilerinin ekrana yansıtılması, öğrencinin login stateinin tutulması gibi işler bu bean aracılığı ile yapılmaktadır.

SirketBean

Bu sınıf, sitenin şirket ile ilgili kısımlarından gelen bilgilerin tutulması ve uygun çıktılar yapılmasından sorumludur. Eğer "şirket yetkilisi login, öğrenci arama, seçme" işlemlerinden birisi yapılıyorsa gelen bilgiler bu bean aracılığı ile sisteme ulaşmaktadır.

com.ubenzer.sst.proje2.model

Bu paket yazılımın model kısmını oluşturmaktadır. Yazılımın küçüklüğü nedeniyle, çok amaçlı veya projeyle doğrudan alakası olmayan utiliy sınıfları ve domaindeki nesneleri temsil eden sınıflar da bu paket altına alınmıştır.

DAL

Data Access Layer sınıfı yazılımın veritabanı ile bağlantı kurmasını sağlayaca veritabanı bağlantı kodlarını içermektedir. Bu sınıfın değiştmesi ile projenin bağlancağı veritabanı motoru (DBMS) ve veritabanı (connection string) kolayca değiştirilebilmektedir.

Herhangi başka bir sınıf VT bağlantısı kurmak istediklerinde bağlantı nesnesini bu sınıftan isterler.

Ogrenci

Bu sınıf domain için geçerli bir öğrenci nesnesi yaratmak için kullanılabilir. Yani araması yapılıp döndürülen her bir öğrenici bu sınıfın bir instancesi olabilir. Oturum açmış bir öğrencinin bilgisi de yine bir Öğrenci nesnesini referans alınarak tutulur.

Bu sınıf aynı zamanda statik metotlar da sunmaktadır. Bu statik metotlar yeni öğrenci eklemek ve belirli parametrelere göre öğrenci araması yapmak için kullanılmaktadır.

Bu sınıf veritabanına erişmek için DAL sınıfından bağlantı nesnesi alır.

SecilebilirOgrenci

Şirket yetkilisinin öğrencileri arayıp bazılarını seçmesi durumunda öğrencilerin “seçilip” “seçilmemesi” sözkonusudur. Ancak bu öğrencilerinin adı, ilgi alanları gibi bir öz değerleri değildir, şirket yetkilisinin istediği ile belirlenmiş, sistemde sadece geçici olarak saklanması gereken bir durumdur.

Bu özelliği sağlamak için içerisinde seçililik durumunu tutuan boolean bir değer ve bir Öğrenci nesnesi taşıyan bir SeçilebilirÖğrenci sınıfı vardır.

Bu sınıf sadece bilgi tutma amaçlıdır. Başka bir şey yapmamaktadır. Getter ve setter dışında metodu bulunmamaktadır.

Sirket

Bu sınıf domain için geçerli bir şirket nesnesi yaratmak için kullanılabilir. Oturum açmış bir şirket yetkilisi bir Şirket nesnesi referans alınarak tutulur.

Bu sınıf aynı zamanda şirket sorgulaması yapmaya imkân sunan statik metotlar da sunmaktadır. Proje metninde istenmediği için ve zaman kısıtlarından dolayı şirket ekleme gibi özellikler projeye eklenmemiştir.

Bu sınıf veri tabanına erişmek için DAL sınıfından bağlantı nesnesi alır.

com.ubenzer.sst.proje2.view

Bu olduğu varsayılan sanal bir pakettir. Projede doğrudan görülmemektedir. MVC patterninin "View" kısmına denk gelmektedir. Temel olarak "WebContent" klasörünün içerisinde yer alan .xhtml ve .jsp dosyaları bu pakete ait sayılabilir.

index.jsp

Sitenin ilk giriş dosyasıdır. Bu dosyanın tek amacı ziyaretçiyi JSF'li başlangıç sayfası olan `/faces/roleSelection.xhtml` konumuna yönlendirmektir.

loginOgrenci.xhtml

Bu sayfa öğrencinin oturum açmasını sağlayan arayüzü oluşturmaktadır. İş mantığının bu sayfalarda yer almadığı (bu sayfaların uzantısının bile .xhtml ile bittiği) dikkat edilmesi gereken bir noktadır.

Eğer oturum açmaya çalışan öğrenci sistemde yoksa bu öğrenci sisteme eklenir. Eğer sistemde olan bir öğrenci oturum açmaya çalışıyorsa şifresi kontrol edilir ve bilgi güncelleme sayfasına yollanır. Eğer öğrencinin şifresi yanlış ise bu bir uyarı mesajı ile bildirilir, şifresi projenin kontrolünü kolaylaştırmak amaçlı olarak "hatırlatılır".

loginSirket.xhtml

Bu sayfa şirket yetkilisinin oturum açmasını sağlayan arayüzü oluşturmaktadır. Bu sayfa aracılığı ile kullanıcı adı ve şifre alınır, doğruysa öğrenci arama sayfasına yönlendirlir. Hatalı şifre gireb yetkiliye proje icabı şifre "hatırlatılır". Eğer şirket yetkilisi zaten oturum açmışsa bu sayfada "Giriş yapmışsınız." uyarısı görüntülenir.

ogrenci.xhtml

Bu sayfada yeni kayıt edilen/bilgileri düzenlenen öğrencilerin bilgileri görünmektedir. Bilgiler girildikten sonra değişiklikler buton aracılığı ile kaydedilebilir. Dikkat edilmesi gereken nokta "şifre"nin her sayfa yüklenişinde silinmesidir. Bu güvenlik amaçlı olup JSF tarafından gerçekleştirilmektedir. Eğer bu istenmeyen bir durumsa aşağıdaki değişiklik yapılarak bu önlenebilir:

```
<h:inputSecret value="#{OgrenciBean.password}"></h:inputSecret>
```

yerine

```
<h:inputText value="#{OgrenciBean.password}"></h:inputText>
```

yazılacak.

Bu sayfada girilen veriler için veri tipi kontrolü **yapılmaktadır.** Ancak hatalı veriler sadece sisteme eklenmemektedir, ayrıca hatalı girilen bilgilerden dolayı kullanıcı uyarılmamaktadır. Bu, proje metninde istenmediğ için zaman kısıtları nedeniyle eklenmemiştir.

roleSelection.xhtml

Bu sayfa, JSF için giriş sayfasıdır. Burada yapılmak istenen seçilmelidir. Öğrenci olarak giriş yapılabilir veya şirket yetkilisi olarak giriş yapılarak öğrenci araması yapılabilir.

secililer.xhtml

Şirket yetkilisinin seçmiş olduğu öğrencilerin listelendiği sayfadır. Yapılan seçim session boyunca saklanmaktadır. Yani, bu sayfadan ayrılıp başka işler yapılması halinde bile, session timeout olmadığı ve tarayıcı kapatılmadığı sürece aynı şirket yetkilisi bilgilere tekrardan erişebilecektir.

sirket.xhtml

Şirket yetkilisinin ad ve ilgi alanları parametrelerine göre arama yapabileceği, arama yapmadan tüm öğrenci listesini görebileceği ve öğrencilerin solundaki kutucuğu tikleyip aşağıdaki güncelleme butonu aracılığı ile öğrencileri seçebileceği bir sayfadır.

Öğrencilerin adı veya ilgi alanına göre arama, ilgili kutucuklardan birisi doldurularak yapılabilir. İçerisinde yazılan kelimeler geçen tüm öğrenciler listelenecektir. Eğer tüm öğrencilerin listelenmesi isteniyorsa boşluk “” aratılabilir.

Seçilmiş öğrenciler görülebilir ve istenirse bu liste boşaltılabilir.

Dikkat: Oturum açmadan önce bu sayfanın adresini doğrudan adres satrına yazarak ulaşırsanız hata mesajı alacaksınız. Sistemdeki oturum açma işlemleri **gerçekten olmaktadır.** Sadece basit bir sayfa yönlendirilmesinden ibaret olmayıp bu konuda güvenlik ciddiye alınmıştır.

Bir pakete girmeyen diğer dosyalar

model.ucls

Otomatik olarak yaratılmış sınıf diyagramı dosyasıdır. Bu dosya kodlarda değişiklik yapıldığında otomatik olarak güncellenmektedir.

web.xml

Tüm JAVA web projeleride bulunması gereken, gelen isteklerin hangi sınıf(lar)a gönderileceğini belirleyen yapılandırma dosyasıdır. Bizim yapılandırmamızda tüm istekler `javax.faces.webapp.FacesServlet` tarafından karşılanmakta olup, sistemdeki .xhtml dosyalarının JSF ile render edilerek gönderilebilmesi –kısacası sistemin çalışması için- tüm dosyalar `/faces/*` patternine uygun olarak çağırılmalıdır.

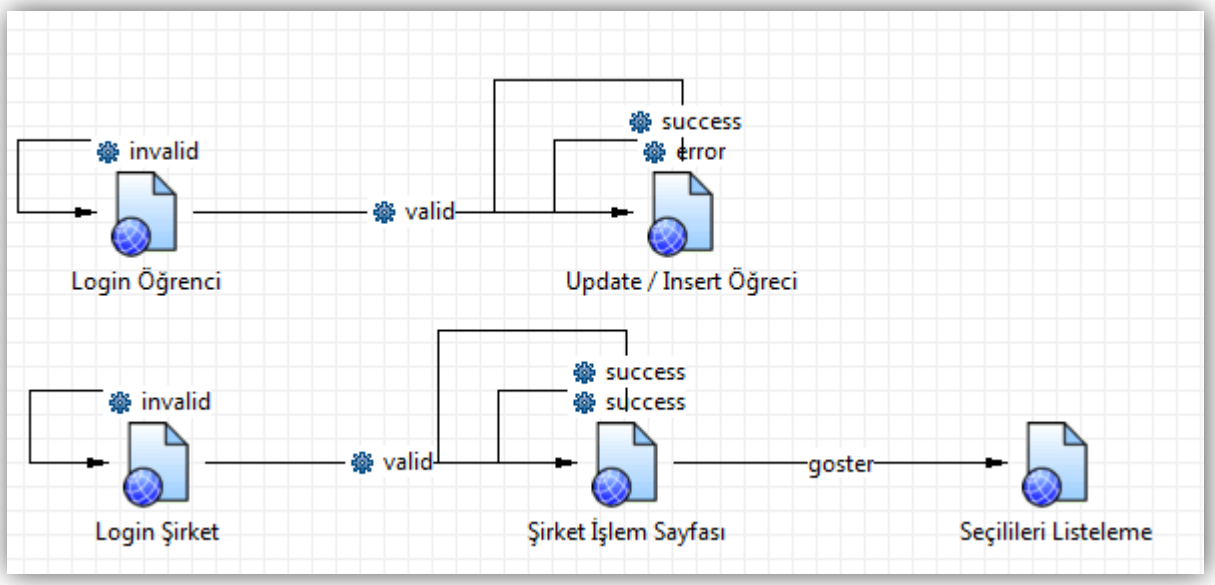
Bu yapılandırma dosyası Eclipse tarafından otomatik olarak yaratılmıştır.

faces-config.xml

JSF’nin çalımasını sağlayan, genellikle sayfalar arası yönlendirmeler ve “Bean”ların bilgileri tutma süresi/yaşam süresi yapılandırmalarının yapıldığı JSF’nin temel yapılandırma dosyasıdır.

Oturum açma ve seçilmiş öğrenci gibi bilgilerin sadece request bazında değil, kullanıcı sitede dolaştığı sürece tutulmasını sağlamak amacıyla tüm beanlar session scopeunda tutulmuştur. Bununla beraber proje daha büyük olsaydı, request ve application scopeunda beanlar da olabilirdi.

Faces-config.xml Eclipse tarafından görselleştirilebilmektedir. Bu görselleştirme projenin işleyişini anlamak açısından yararlı olabilir. İlerleyen sayfada bu görselleştirme bulunmaktadır:



Temalar

Bu projede aynı zamanda JSF 2.0 teknolojisi ile birlikte arayüz katmanında kullanılması önerilen Facelets'lerin kullanımına dair örnekler de bulunmaktadır. Temalar Facelets'in bir özelliğidir. ASP.Net'teki Master Page'lere benzemektedirler. Sayfaları belirli parçalara bölerler ve belirli kısımalarının başka yerlerde doldurulmasına olanak tanırlar.

StandartTema.xhtml

Standart tema, sitede kullanılan tüm arayüzün oluşturulduğu temadır. Bu tema kendi içerisinde üç bloktan oluşur:

- Header
- Content
- Footer

Content kısmı, sitenin işleyişine göre gerekli şekilde dinamik olarak doldurulurken, diğer parçalar statiktir.

header.xhtml

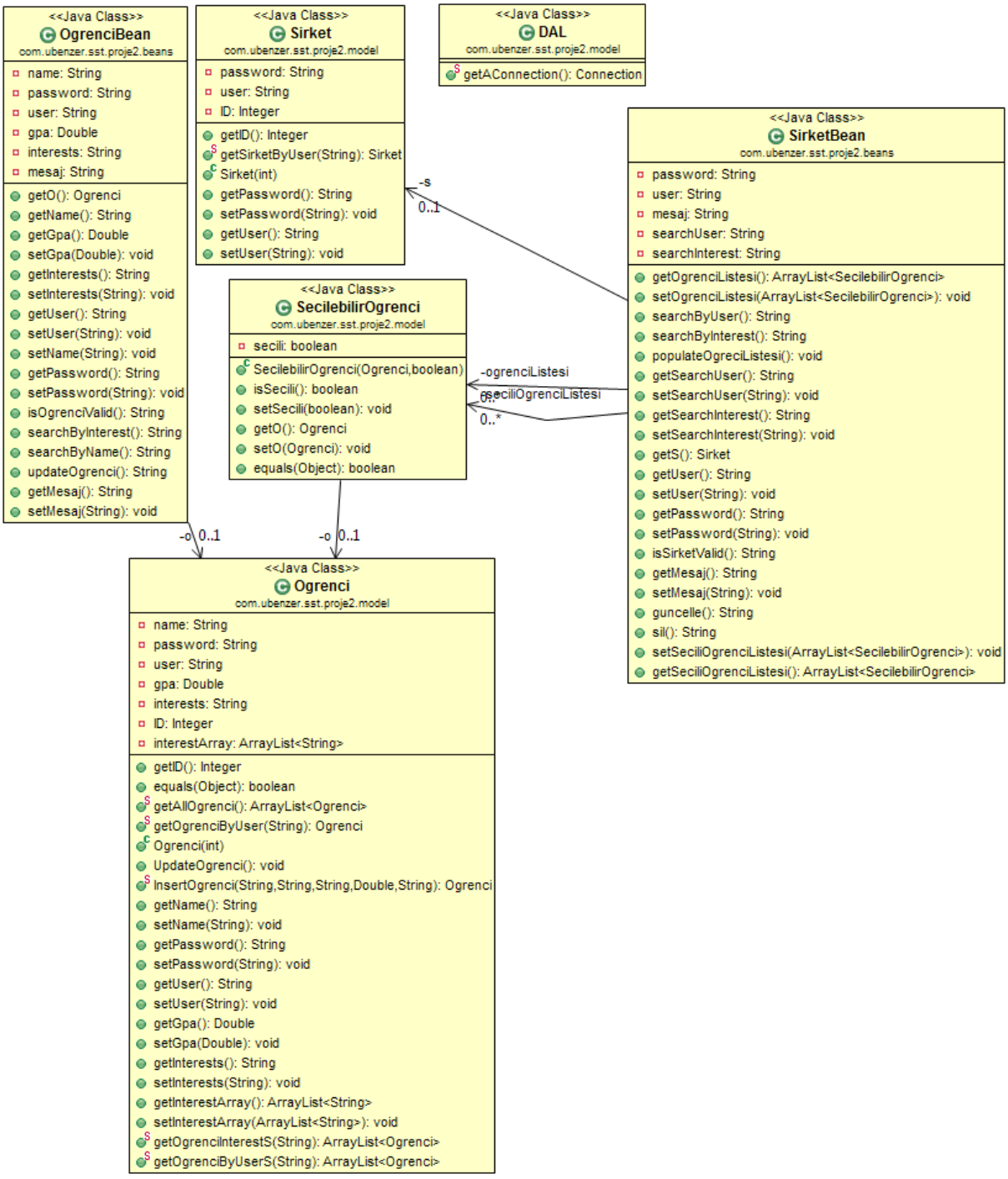
Bu dosya standart temanın header kısmını oluşturur. Sayfanın başlığının tasarımı bu sayfa ile belirlenir.

footer.xhtml

Bu dosya standart temanın footer kısmını oluşturur. Sayfanın alt kısmının tasarımı ve buradaki linkler bu sayfaya kodlananlar ile belirlenir.

Sınıf Diyagramı

* Sınıf diyagramı 14 Mayıs 2011 günü rapora aktarılmıştır. Çok ufak bir ihtimal de olsa sınıf diyagramı alındıktan sonra programın bulunan bir hata dolayısıyla güncellenmiş olmasından dolayı diyagramda ufak değişiklikler yapılmış olunabilir.



Veritabanı

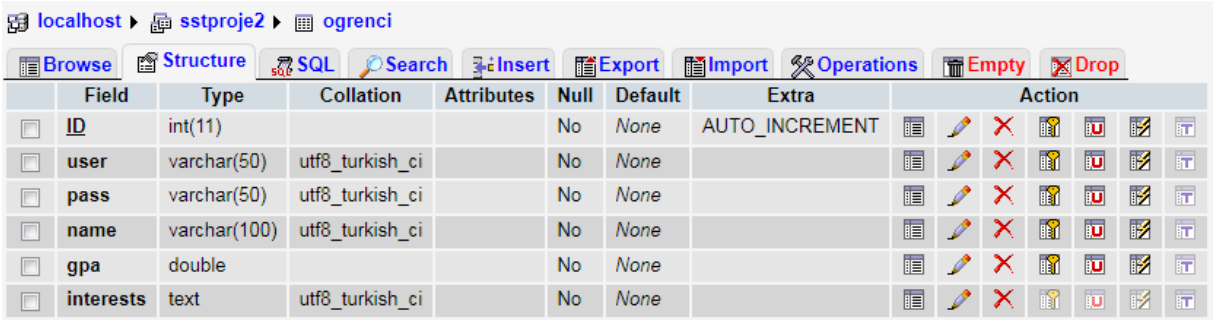
Proejede veritabanı motoru olarak MYSQL kullanılmıştır. Ancak DAL sınıfında yapılabilecek ufak değişiklikler ile tüm DBMS'ler kullanılabilir.

* Veritabanını (içindeki bilgiler haricinde) yaratabileceğiniz SQL cümlecikleri de tablolarla birlikte aşağıda verilecektir.

Veritabanı iki tablodan oluşmaktadır:

Ogrenci

Bu tabloda öğrencilerin bilgileri, kullanıcı adı ve şifreleri tutulmaktadır.



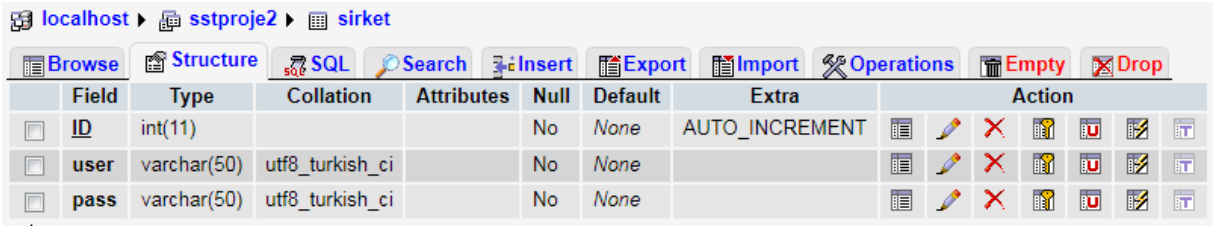
Field	Type	Collation	Attributes	Null	Default	Extra	Action
ID	int(11)			No	None	AUTO_INCREMENT	
user	varchar(50)	utf8_turkish_ci		No	None		
pass	varchar(50)	utf8_turkish_ci		No	None		
name	varchar(100)	utf8_turkish_ci		No	None		
gpa	double			No	None		
interests	text	utf8_turkish_ci		No	None		

Tabloyu yaratmak için gerekli SQL cümleciği

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
CREATE TABLE IF NOT EXISTS `ogrenci` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `user` varchar(50) COLLATE utf8_turkish_ci NOT NULL,
  `pass` varchar(50) COLLATE utf8_turkish_ci NOT NULL,
  `name` varchar(100) COLLATE utf8_turkish_ci NOT NULL,
  `gpa` double NOT NULL,
  `interests` text COLLATE utf8_turkish_ci NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_turkish_ci AUTO_INCREMENT=9 ;
```

Sirket

Bu tabloda şirket yetkililerinin kullanıcı adları ve şifreleri tutulmaktadır.



Field	Type	Collation	Attributes	Null	Default	Extra	Action
ID	int(11)			No	None	AUTO_INCREMENT	
user	varchar(50)	utf8_turkish_ci		No	None		
pass	varchar(50)	utf8_turkish_ci		No	None		

Tabloyu yaratmak için gerekli SQL cümleciği

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
CREATE TABLE IF NOT EXISTS `sirket` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `user` varchar(50) COLLATE utf8_turkish_ci NOT NULL,
  `pass` varchar(50) COLLATE utf8_turkish_ci NOT NULL,
  PRIMARY KEY (`ID`),
  UNIQUE KEY `user` (`user`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 COLLATE=utf8_turkish_ci AUTO_INCREMENT=2 ;
```

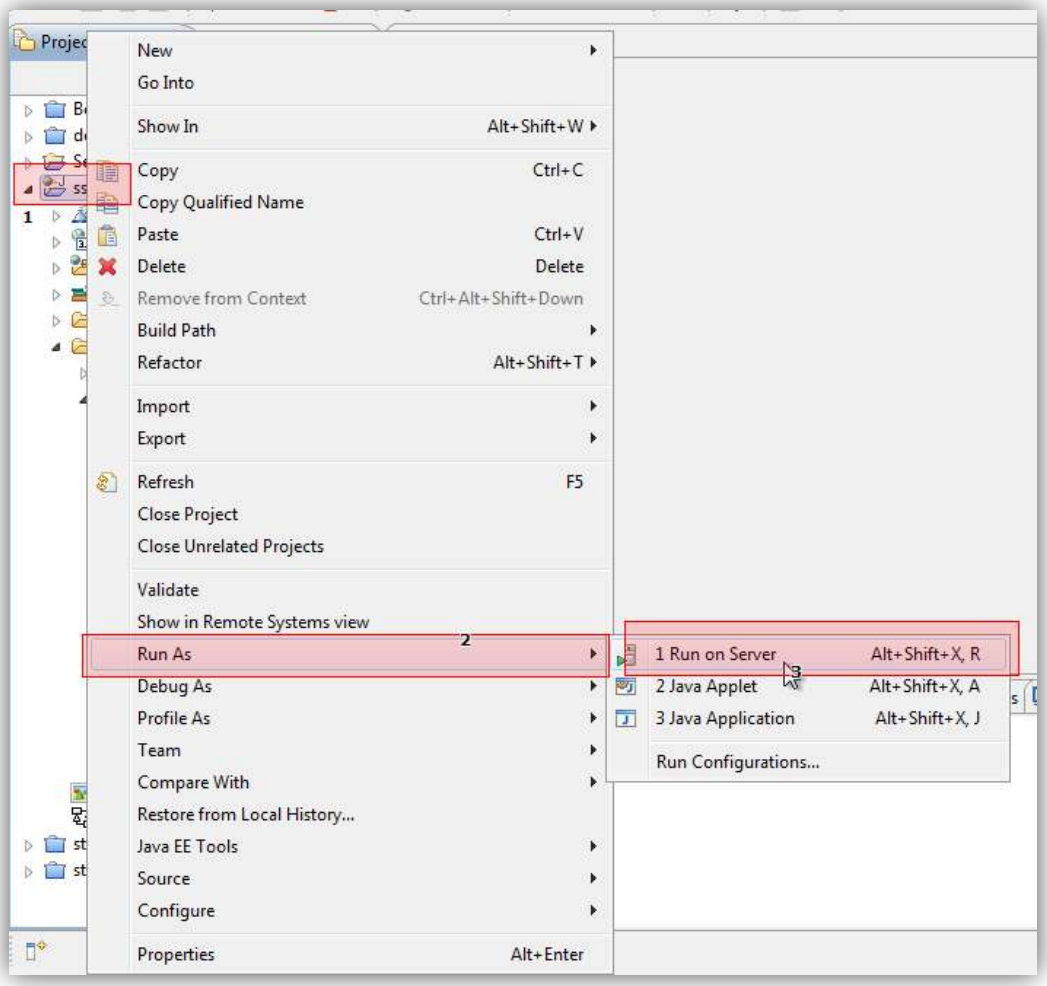
Kullanım Kılavuzu

LÜTFEN DİKKAT: Bu raporda yer alan kaynak kodları ve ekran görüntüleri projenin son halini tam yansıtmayabilir. Bu kodlar ve ekran görüntüleri otomatik güncellenmediğinden rapor yazıldıktan sona yakalanan bir hata sonrası kodların ve görüntülerin değiştirilmesi durumunda bu rapora yansımayacaktır.

Programı Başlatmak

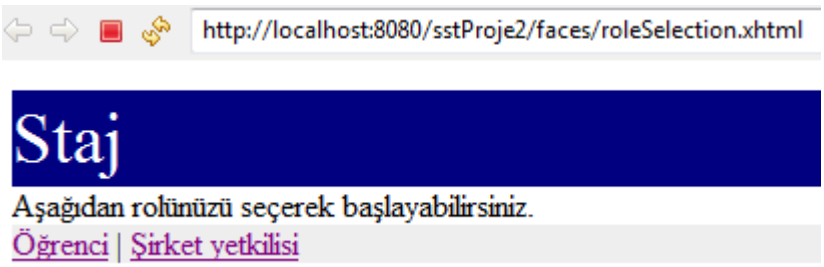
Her ne kadar ticari bir ortamda bu uygulama gerçekten yayına alınacak olursa bu çalıtırma yöntemi doğru olmasa da, proje geliştiricileri ve kontrolcüleri için doğrudan Eclipse üzerinden proje başlatılabilir. Bunun için yapılması gereken projeye sağ tıklamak ve Run As->On Server seçeneklerini seçmektir.

Bu noktada bilgisayarınızda kurulu ve düzgün yapılandırılmış, ilgili teknolojileri destekleyen bir sunucu olduğu varsayılmıştır.



Ana Menü

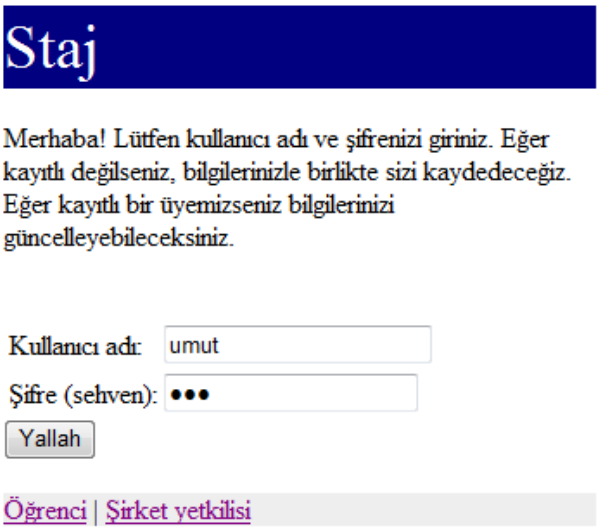
Sitenin ilk açılış sayfasında ve diğer tüm sayfaların alt kısımlarında Öğrenci ve Şirket işlemleri arasında dolaşabileceğiniz bağlantılar bulunmaktadır.



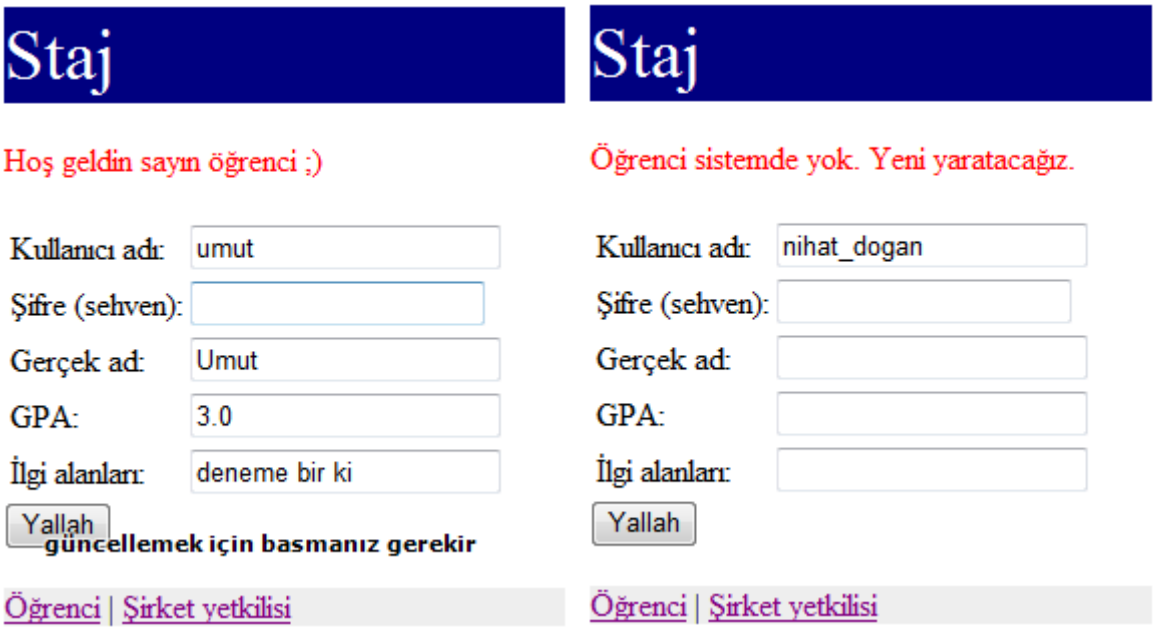
Öğrenci İşlemleri

Yeni öğrenci eklenebilecek veya zaten ekli bir öğrencinin bilgilerini düzenlemek amacı ile kullanıcı girişi yapılabilecek arayüz. Bu kısımda kullanıcı adı ve şifre girilmelidir. Üye değilseniz, sisteme eklenirsiniz, üyeyseniz oturum açmış olursunuz.

Hatalı şifre girerseniz projenin kontrolü açısından doğru şifreniz size "hatırlatılır".

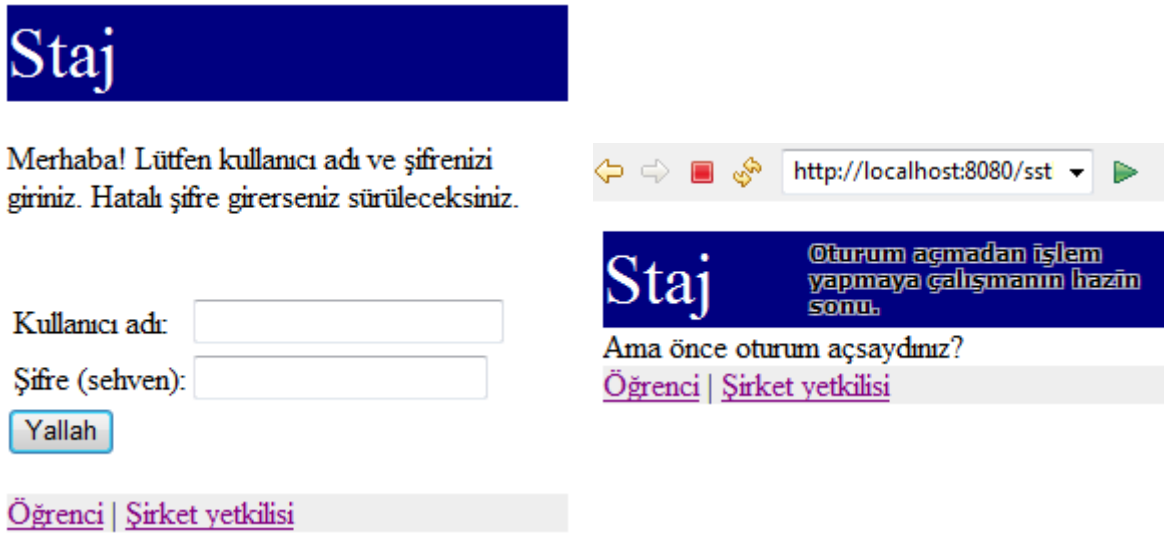


Oturum açınca şifre hariç güncel bilgileriniz gösterilir. Yenilerini yazıp formu göndererek değişiklik yapabilirsiniz. Daha ayıntılı bilgi için teknik kısımları okumanız önerilir.



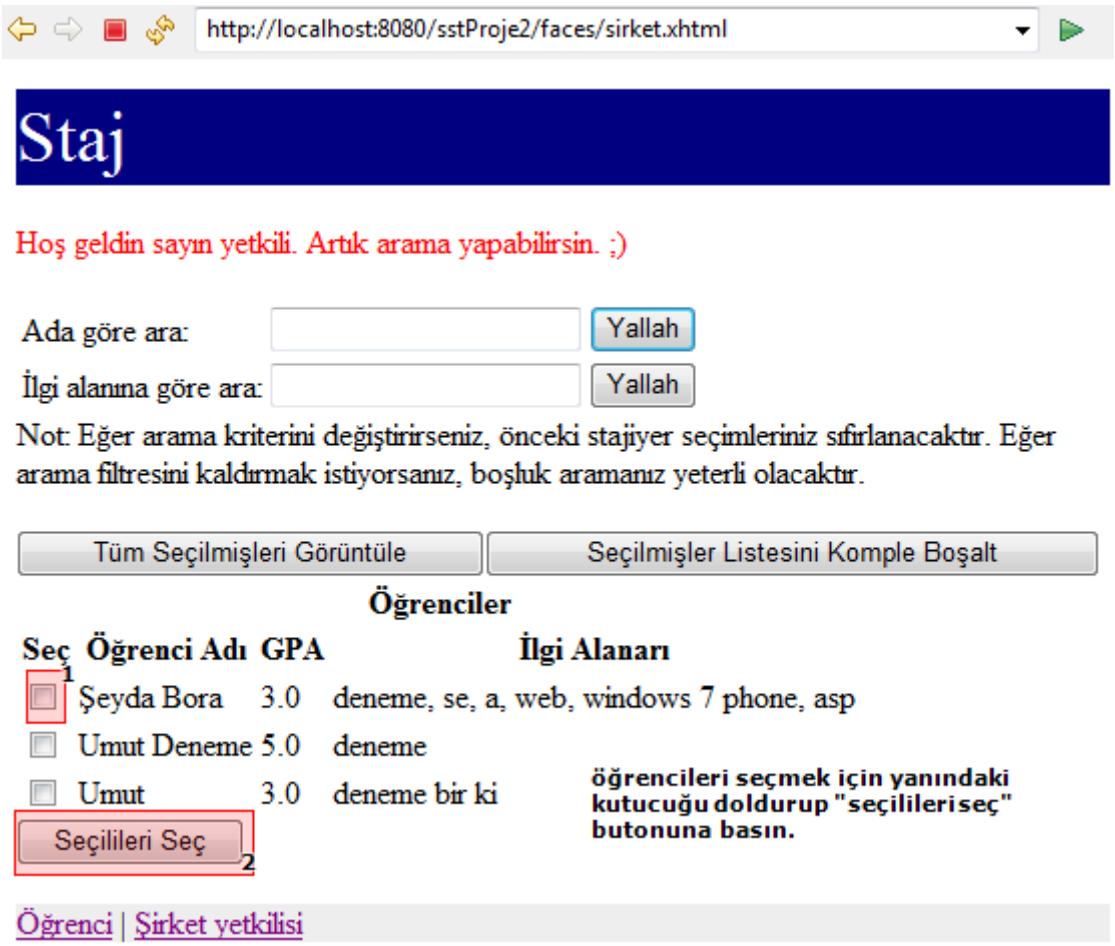
Şirket İşlemleri

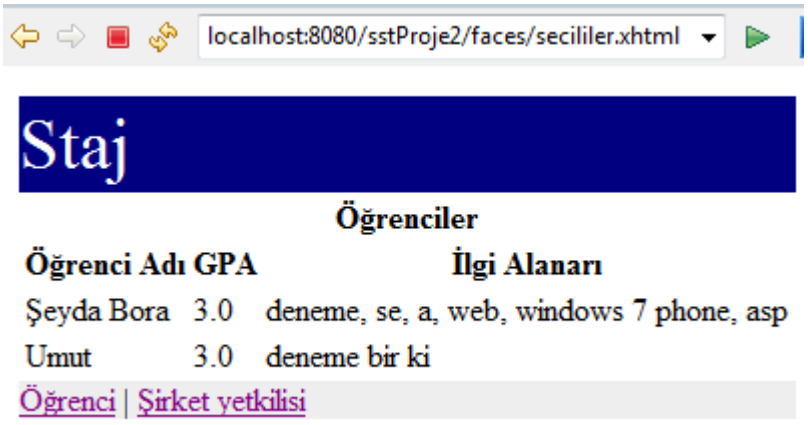
Stajyer öğrencilerin tamamının görüntülenebileceği ve ad/ilgi alanına göre arama yapılabileceği, seçim ve seçilenleri görüntüleme işlemlerinin yapılabileceği arayüzlerdir.



Oturum açtığınızda görüntülenen sayfada ilk etapta tüm kayıtlı öğrenciler listelenir. Daha sonra isterseniz üstteki iki kutucuk ile filtreleme yapabilirsiniz. Arama metin bazlı çalışır, bu kısım kasten basit tutulmuşur. İki alanda aynı anda arama yapamazsınız. Örneğin ad için filtreleme yapıyorken aynı anda ilgi alanına göre de yapamasınız.

Eğer tekrar tüm öğrencileri listeletmek istiyorsanız boşluk “” aratınız.


http://localhost:8080/sstProje2/faces/sirket.xhtml
Staj
Hoş geldin sayın yetkili. Artık arama yapabilirsin. ;)
Ada göre ara:
Yallah
İlgi alanına göre ara:
Yallah
Not: Eğer arama kriterini değiştirirseniz, önceki stajiyer seçimleriniz sıfırlanacaktır. Eğer arama filtresini kaldırmak istiyorsanız, boşluk aramanız yeterli olacaktır.
Tüm Seçilmişleri Görüntüle
Seçilmişler Listesini Komple Boşalt
Öğrenciler
Seç
Öğrenci Adı
GPA
İlgi Alanları
1
Şeyda Bora 3.0 deneme, se, a, web, windows 7 phone, asp
Umut Deneme 5.0 deneme
Umut 3.0 deneme bir ki
öğrencileri seçmek için yanındaki kutucuğu doldurup "seçilileri seç" butonuna basın.
Seçilileri Seç
2
Öğrenci | Şirket yetkilisi


localhost:8080/sstProje2/faces/secililer.xhtml
Staj
Öğrenciler
Öğrenci Adı
GPA
İlgi Alanları
Şeyda Bora 3.0 deneme, se, a, web, windows 7 phone, asp
Umut 3.0 deneme bir ki
Öğrenci | Şirket yetkilisi

Kaynak Kodlar

OgrenciBean.java

```
/**
 * Öğrenci ile ilgili arayüzlerin bilgilerinin yönetildiği
 * beandır.
 */
package com.ubenzer.sst.proje2.beans;

import com.ubenzer.sst.proje2.model.Ogrenci;

public class OgrenciBean {

	private Double gpa = new Double(0);
	private String interests = new String();
	private String mesaj = new String();
	private String name = new String();
	private Ogrenci o;
	private String password = new String();
	private String user = new String();

	/* Getter metotlar */
	public Double getGpa() {
		return gpa;
	}

	public String getInterests() {
		return interests;
	}

	public String getMesaj() {
		return mesaj;
	}

	public String getName() {
		return name;
	}

	public Ogrenci getO() {
		return o;
	}

	public String getPassword() {
		return password;
	}

	public String getUser() {
		return user;
	}

	/**
	 * Şu an beanda bulunan bilgileri kullanarak User ve Password bilgilerine
	 * uyan bir öğrencinin sistemde bulunup bulunmadığını denetler.
	 *
	 * Eğer öğrenci sistemde yoksa, yeni yatatılır. Eğer öğrenci sistemde ve
	 * şifresi doğruysa bilgileri düzenlenir.
	 *
	 * Bunlar "valid" döndürür. Dönen değerler faces.xml'de işlenir.
	 *
	 * Eğer şifre hatalı ise "invalid" döner.
	 *
	 * @return "valid" or "invalid"
	 */
	public String isOgrenciValid() {

		o = Ogrenci.getOgrenciByUser(getUser());

		if (o == null) {
			setMesaj("Öğrenci sistemde yok. Yeni yaratacağız.");
			setName("");
			setGpa(0.0);
			setInterests("");
			return "valid";
		}
		if (o.getPassword().equals(this.getPassword())) {
			setMesaj("Hoş geldin sayın öğrenci ;)");
			setName(o.getName());
			setGpa(o.getGpa());
```

```
                        setInterests(o.getInterests());
                        return "valid";
                } else {
                        setMesaj("Öğrenci mevcut ama parola hatalı. Bence bir de '"
                                        + o.getPassword()
                                        + "' girmeyi deneyin. [hiii, güvenlik açığı]");
                        return "invalid";
                }
        }

        /* Setter metotlar */
        public void setGpa(Double gpa) {
                this.gpa = gpa;
        }

        public void setInterests(String interests) {
                this.interests = interests;
        }

        public void setMesaj(String mesaj) {
                this.mesaj = mesaj;
        }

        public void setName(final String name) {
                this.name = name;
        }

        public void setPassword(final String password) {
                this.password = password;
        }

        public void setUser(String user) {
                this.user = user;
        }

        /**
         * Güncel öğrencinin bilgilerinde yapılan değişikliği veritabanına yazar.
         * Eğer öğrenci yeni yaratılıyorsa, önce bir Öğrenci nesnesi yaratılarak bu
         * veritabanına eklenir.
         *
         * Öğrenci güncelleniyorsa Bean aracılığı ile bilgileri güncellenmiş
         * öğrencinin bilgileri öğrenci nesnesinde güncellenir ve bu bilgiler
         * veritabanına aktarılır.
         *
         * Eğer DB ile alakalı bir sıkıntı olursa (bağlantı problemi, unique
         * username problemi "error" döner.
         *
         * @return "success" or "error"
         */
        public String updateOgrenci() {
                if (o == null) {
                        try {
                                o = Ogrenci.InsertOgrenci(getUser(), getPassword(), getName(),
                                                getGpa(), getInterests());
                                setMesaj("Ekleme tamam.");
                        } catch (Exception e) {
                                setMesaj("Sıkıntı oldu. Öğrenciyi güncelleyemedik.");
                                e.printStackTrace();
                                return "error";
                        }

                } else {

                        o.setGpa(getGpa());
                        o.setInterests(getInterests());
                        o.setName(getName());
                        o.setPassword(getPassword());
                        o.setUser(getUser());

                        try {
                                o.UpdateOgrenci();
                                setMesaj("Güncelleme tamam.");
                        } catch (Exception e) {
                                setMesaj("Sıkıntı oldu. Öğrenciyi ekleyemedik.");
                                e.printStackTrace();
                                return "error";
                        }
                }
                return "success";
        }
}
```

SirketBean.java

```
/**
 * Şirket ile ilgili arayüzlerin bilgilerinin yönetildiği
 * beandır.
 */
package com.ubenzer.sst.proje2.beans;

import java.util.ArrayList;

import com.ubenzer.sst.proje2.model.Ogrenci;
import com.ubenzer.sst.proje2.model.SecilebilirOgrenci;
import com.ubenzer.sst.proje2.model.Sirket;

public class SirketBean {

	private String mesaj = new String();
	private ArrayList<SecilebilirOgrenci> ogrenciListesi;
	private String password = new String();
	private Sirket s;
	private String searchInterest = new String();
	private String searchUser = new String();
	private ArrayList<SecilebilirOgrenci> seciliOgrenciListesi = new ArrayList<SecilebilirOgrenci>();
	private String user = new String();

	/* Getter metotlar */
	public String getMesaj() {
		return mesaj;
	}

	public ArrayList<SecilebilirOgrenci> getOgrenciListesi() {
		if (ogrenciListesi == null)
			populateOgreciListesi();
		return ogrenciListesi;
	}

	public String getPassword() {
		return password;
	}

	public Sirket getS() {
		return s;
	}

	public String getSearchInterest() {
		return searchInterest;
	}

	public String getSearchUser() {
		return searchUser;
	}

	public ArrayList<SecilebilirOgrenci> getSeciliOgrenciListesi() {
		return seciliOgrenciListesi;
	}

	public String getUser() {
		return user;
	}

	/**
	 * Seçili öğrenci listesini güncellemektedir. Seçili öğrenciler
	 * seciliOgrenciListesi ArrayList'inde tutulur.
	 *
	 * @return success
	 */
	public String guncelle() {
		/*
		 * Eğer sadecee zaten daha önce seçilmemişse seçeceğiz. Bunun kontrolü
		 * için nesnelerin eşitliğinden faydalanılırız.
		 */
		for (SecilebilirOgrenci o : ogrenciListesi) {
			if (o.isSecili() && !seciliOgrenciListesi.contains(o))
				seciliOgrenciListesi.add(o);
		}
		return "success";
	}

	/**
	 * Şu an beanda bulunan bilgileri kullanarak User ve Password bilgilerine
	 * uyan bir şirketin sistemde bulunup bulunmadığını denetler.
	 *
```

```
     * Eğer şirket sistemde ve şifresi doğruysa bilgileri s nesnesine atılır.
     * Oturum açılmış olur.
     *
     * Bu "valid" döndürür. Dönen değerler faces.xml'de işlenir.
     *
     * Eğer şifre hatalı ise veya şirket yoksa "invalid" döner.
     *
     * @return "valid" or "invalid"
     */
    public String isSirketValid() {

            s = Sirket.getSirketByUser(getUser());

            if (s == null) {
                    setMesaj("Şirket sistemde yok.");
                    return "invalid";
            }
            if (s.getPassword().equals(this.getPassword())) {
                    setMesaj("Hoş geldin sayın yetkili. Artık arama yapabilirsin. ;)");
                    return "valid";
            } else {
                    setMesaj("Hatalı şifre. Hekır mısınız? Şifre olarak deneyin: "
                                    + s.getPassword());
                    s = null;
                    return "invalid";
            }
    }

    /**
     * Tüm öğrencileri veritabanından alarak ekranda gösterilmeye hazırlar.
     */
    public void populateOgreciListesi() {
            ogrenciListesi = new ArrayList<SecilebilirOgrenci>();

            ArrayList<Ogrenci> ogrs = Ogrenci.getAllOgrenci();
            for (Ogrenci o : ogrs) {
                    ogrenciListesi.add(new SecilebilirOgrenci(o, false));
            }
    }

    /**
     * Sadece belirli özelliğe sahip öğrencileri veritabanından alarak ekranda
     * gösterilmeye hazırlar.
     *
     * @return success
     */
    public String searchByInterest() {
            ogrenciListesi = new ArrayList<SecilebilirOgrenci>();

            ArrayList<Ogrenci> ogrs = Ogrenci.getOgrenciInterestS(this
                            .getSearchInterest());
            for (Ogrenci o : ogrs) {
                    ogrenciListesi.add(new SecilebilirOgrenci(o, false));
            }
            return "success";
    }

    /**
     * Sadece belirli isime sahip öğrencileri veritabanından alarak ekranda
     * gösterilmeye hazırlar.
     *
     * @return success
     */
    public String searchByUser() {
            ogrenciListesi = new ArrayList<SecilebilirOgrenci>();

            ArrayList<Ogrenci> ogrs = Ogrenci.getOgrenciByUserS(this
                            .getSearchUser());
            for (Ogrenci o : ogrs) {
                    ogrenciListesi.add(new SecilebilirOgrenci(o, false));
            }
            return "success";
    }

    /* Setter metotlar */
    public void setMesaj(String mesaj) {
            this.mesaj = mesaj;
    }

    public void setOgrenciListesi(ArrayList<SecilebilirOgrenci> ogrenciListesi) {
            this.ogrenciListesi = ogrenciListesi;
    }
```

```
	public void setPassword(final String password) {
		this.password = password;
	}

	public void setSearchInterest(String searchInterest) {
		this.searchInterest = searchInterest;
	}

	public void setSearchUser(String searchUser) {
		this.searchUser = searchUser;
	}

	public void setSeciliOgrenciListesi(
			ArrayList<SecilebilirOgrenci> seciliOgrenciListesi) {
		this.seciliOgrenciListesi = seciliOgrenciListesi;
	}

	public void setUser(String user) {
		this.user = user;
	}

	/**
	 * Seçili öğreci listesini boşaltır.
	 *
	 * @return success
	 */
	public String sil() {
		seciliOgrenciListesi = new ArrayList<SecilebilirOgrenci>();
		return "success";
	}
}
```

DAL.java

```
/**
 * Data Access Layer: Projenin veritabanı bağlantısı
 * kurmak için gereken bağlantıyı aldığı utility sınıfıdır.
 */
package com.ubenzer.sst.proje2.model;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DAL {

	/**
	 * Veritananına bağlanmak için gerekli bir bağlantı döndürür.
	 *
	 * @return Bağlantı
	 */
	public static Connection getAConnection() throws SQLException,
			ClassNotFoundException {
		Class.forName("com.mysql.jdbc.Driver");
		String url = "jdbc:mysql://localhost:3306/sstproje2?useUnicode=true&characterEncoding=utf8";
		Connection con = DriverManager.getConnection(url, "root", "");

		return con;
	}
}
```

Ogrenci.java

```
/**
 * Öğrenci bilgilerinin tutulduğu bir domain nesnesi,
 * aynı zamanda öğrencilerle ilgili işlerin yapılabildiği
 * statik metotlar içeren bir sınıftır.
 */
package com.ubenzer.sst.proje2.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;

public class Ogrenci {

	/* Statik Metotlar */

	/**
	 * Veritabanındaki tüm öğrencileri bir ArrayList halinde döndürür.
```

```
 *
 * @return Tüm öğrenciler
 */
public static ArrayList<Ogrenci> getAllOgrenci() {
	Connection con;
	ArrayList<Ogrenci> don = new ArrayList<Ogrenci>();
	try {

		con = DAL.getAConnection();

		PreparedStatement pstmt = con
				.prepareStatement("SELECT * FROM ogrenci");
		ResultSet rs = pstmt.executeQuery();

		while (rs.next()) {
			don.add(new Ogrenci(rs.getInt("ID")));
		}

	} catch (Exception e) {
		e.printStackTrace();
	}

	return don;
}

/**
 * Kullanıcı adına göre öğrenci sorgulaması yapar ve eğer öğrenci varsa
 * döndürür.
 *
 * @param Kullanıcı
 *            adı
 * @return Varsa öğrenci, yoksa null
 */
public static Ogrenci getOgrenciByUser(String user) {
	Connection con;
	try {
		con = DAL.getAConnection();

		PreparedStatement pstmt = con
				.prepareStatement("SELECT * FROM ogrenci WHERE user = ?");
		pstmt.setString(1, user);
		ResultSet rs = pstmt.executeQuery();

		if (rs.next()) {
			return new Ogrenci(rs.getInt("ID"));
		}
	} catch (Exception e) {
		e.printStackTrace();
		return null;
	}
	return null;
}

/**
 * Öğrencileri isimlerine göre arar. Sonuçlar ArrayList içerisinde
 * döndürülür.
 *
 * @param Aranacak isim
 * @return Öğrenci listesi
 */
public static ArrayList<Ogrenci> getOgrenciByUserS(String searchUser) {

	if (searchUser.length() < 1)
		return getAllOgrenci();

	Connection con;
	ArrayList<Ogrenci> don = new ArrayList<Ogrenci>();
	try {

		con = DAL.getAConnection();

		PreparedStatement pstmt = con
				.prepareStatement("SELECT * FROM ogrenci WHERE name LIKE ?");
		pstmt.setString(1, "%" + searchUser + "%");
		ResultSet rs = pstmt.executeQuery();

		while (rs.next()) {
			don.add(new Ogrenci(rs.getInt("ID")));
		}

	} catch (Exception e) {
		e.printStackTrace();
```

```
		}

		return don;
	}

	/**
	 * Öğrencileri ilgi alanlarına göre arar. Sonuçlar ArrayList içerisinde
	 * döndürülür.
	 *
	 * @param Aranacak ilgi alanı
	 * @return Öğrenci listesi
	 */
	public static ArrayList<Ogrenci> getOgrenciInterestS(String searchInterest) {

		if (searchInterest.length() < 1)
			return getAllOgrenci();

		Connection con;
		ArrayList<Ogrenci> don = new ArrayList<Ogrenci>();
		try {

			con = DAL.getAConnection();

			PreparedStatement pstmt = con
					.prepareStatement("SELECT * FROM ogrenci WHERE interests LIKE ?");
			pstmt.setString(1, "%" + searchInterest + "%");
			ResultSet rs = pstmt.executeQuery();

			while (rs.next()) {
				don.add(new Ogrenci(rs.getInt("ID")));
			}

		} catch (Exception e) {
			e.printStackTrace();
		}

		return don;
	}

	/**
	 * Verilen bilgilerle bir öğrenci nesnesi yaratır ve bunu veritabanına
	 * ekler.
	 *
	 * @param Kullanıcı adı (unique)
	 * @param Şifre
	 * @param İsim
	 * @param Ortalama
	 * @param İlgi alanları
	 * @return Yeni eklenen öğrenci nesnesi
	 * @throws Veritabanı veya unique key ile ilgili hatalar
	 */
	public static Ogrenci InsertOgrenci(String user, String pass, String name,
			Double gpa, String interests) throws Exception {

		Connection con = DAL.getAConnection();
		PreparedStatement pstmt = con
				.prepareStatement(
						"INSERT INTO ogrenci (user, pass, name, gpa, interests) VALUES
(?,?,?,?,?)",
						Statement.RETURN_GENERATED_KEYS);
		pstmt.setString(1, user);
		pstmt.setString(2, pass);
		pstmt.setString(3, name);
		pstmt.setDouble(4, gpa);
		pstmt.setString(5, interests);
		pstmt.executeUpdate();
		ResultSet rs = pstmt.getGeneratedKeys();

		rs.next();
		int key = rs.getInt(1);

		pstmt.close();

		return new Ogrenci(key);

	}

	private Double gpa;
	private Integer ID;
	private String interests;
	private String name;
	private String password;
```

```
private String user;

/**
 * Constuctor. Bir öğrenciID'si ile nesne yaratılır.
 *
 * @param ID
 */
public Ogrenci(int ogrenciID) {
	try {
		Connection con = DAL.getAConnection();
		PreparedStatement pstmt = con
				.prepareStatement("SELECT * FROM ogrenci WHERE ID = ?");
		pstmt.setInt(1, ogrenciID);
		ResultSet rs = pstmt.executeQuery();

		if (rs.next()) {
			ID = ogrenciID;
			setUser(rs.getString("user"));
			setPassword(rs.getString("pass"));
			setName(rs.getString("name"));
			setGpa(rs.getDouble("gpa"));
			setInterests(rs.getString("interests"));
		}
	} catch (Exception e) {
		e.printStackTrace();
	}
}

/**
 * Nesne eşitliğinin ID'lere göre karşılaştırılması sağlanır.
 */
public boolean equals(Object o) {
	if (o instanceof Ogrenci && ((Ogrenci) o).getID() == this.getID()) {
		return true;
	}
	return false;
}

/* Getter metotlar */
public Double getGpa() {
	return gpa;
}

public Integer getID() {
	return ID;
}

public String getInterests() {
	return interests;
}

public String getName() {
	return name;
}

public String getPassword() {
	return password;
}

public String getUser() {
	return user;
}

/* Setter metotlar */
public void setGpa(Double gpa) {
	this.gpa = gpa;
}

public void setInterests(String interests) {
	this.interests = interests;
}

public void setName(String name) {
	this.name = name;
}

public void setPassword(String password) {
	this.password = password;
}

public void setUser(String user) {
	this.user = user;
```

```
	}

	/**
	 * Nesnede yapılan değişiklikleri veritabanına aktarır.
	 */
	public void UpdateOgrenci() throws Exception {

		Connection con = DAL.getAConnection();
		PreparedStatement pstmt = con
				.prepareStatement("UPDATE ogrenci SET user=?, pass=?, name=?, gpa=?, interests=?
WHERE ID=?");
		pstmt.setString(1, getUser());
		pstmt.setString(2, getPassword());
		pstmt.setString(3, getName());
		pstmt.setDouble(4, getGpa());
		pstmt.setString(5, getInterests());
		pstmt.setInt(6, getID());
		pstmt.executeUpdate();
		pstmt.close();
	}
}
```

SecilebilirOgrenci.java

```
/**
 * Öğrencilerin sunum katmanında seçilebilir olmasını
 * sağlamak amaçlı kullanılan bir ara sınıftır.
 */
package com.ubenzer.sst.proje2.model;

public class SecilebilirOgrenci {

	private Ogrenci o;
	private boolean secili = false;

	public SecilebilirOgrenci(Ogrenci o, boolean b) {
		this.o = o;
		this.secili = b;
	}

	/**
	 * Seçilebilir öğrencilerin eşit olması, içerisindeki öğrenci nesnelerinin
	 * eşit olması demektir.
	 */
	public boolean equals(Object o) {
		if (o instanceof SecilebilirOgrenci
				&& ((SecilebilirOgrenci) o).getO().equals(this.getO())) {
			return true;
		}
		return false;
	}

	/* Getter metotlar */
	public Ogrenci getO() {
		return o;
	}

	public boolean isSecili() {
		return secili;
	}

	/* Setter metotlar */
	public void setO(Ogrenci o) {
		this.o = o;
	}

	public void setSecili(boolean secili) {
		this.secili = secili;
	}
}
```

Sirket.java

```
/**
 * Şirket bilgilerinin tutulduğu bir domain nesnesi,
 * aynı zamanda şirketlerle ilgili işlerin yapılabildiği
 * statik metotlar içeren bir sınıftır.
 */
package com.ubenzer.sst.proje2.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
```

```
public class Sirket {

	/* Statik Metotlar */

	/**
	 * Kullanıcı adına göre şirket sorgulaması yapar ve eğer şirket varsa
	 * döndürür.
	 *
	 * @param Kullanıcı adı
	 * @return Varsa şirket, yoksa null
	 */
	public static Sirket getSirketByUser(String user) {
		Connection con;
		try {
			con = DAL.getAConnection();

			PreparedStatement pstmt = con
					.prepareStatement("SELECT * FROM sirket WHERE user = ?");
			pstmt.setString(1, user);
			ResultSet rs = pstmt.executeQuery();

			if (rs.next()) {
				return new Sirket(rs.getInt("ID"));
			}
		} catch (Exception e) {
			e.printStackTrace();
			return null;
		}
		return null;
	}

	private Integer ID;
	private String password;
	private String user;

	/**
	 * Constuctor. Bir sirketID ile nesne yaratılır.
	 *
	 * @param ID
	 */
	public Sirket(int sirketID) {
		try {
			Connection con = DAL.getAConnection();
			PreparedStatement pstmt = con
					.prepareStatement("SELECT * FROM sirket WHERE ID = ?");
			pstmt.setInt(1, sirketID);
			ResultSet rs = pstmt.executeQuery();

			if (rs.next()) {
				ID = sirketID;
				setUser(rs.getString("user"));
				setPassword(rs.getString("pass"));
			}
		} catch (Exception e) {
			e.printStackTrace();
		}
	}

	/* Getter metotlar */
	public Integer getID() {
		return ID;
	}

	public String getPassword() {
		return password;
	}

	public String getUser() {
		return user;
	}

	/* Setter metotlar */
	public void setPassword(String password) {
		this.password = password;
	}

	public void setUser(String user) {
		this.user = user;
	}
}
```

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
         pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<body>
         <%
response.sendRedirect("faces/roleSelection.xhtml");
%>
</body>
</html>
```

loginOgrenci.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
         xmlns:ui="http://java.sun.com/jsf/facelets"
         xmlns:h="http://java.sun.com/jsf/html"
         xmlns:f="http://java.sun.com/jsf/core"
         xmlns:c="http://java.sun.com/jsp/jstl/core">

<ui:composition template="/WEB-INF/templates/StandartTema.xhtml">
         <ui:define name="content">
                  <p>Merhaba! Lütfen kullanıcı adı ve şifrenizi giriniz. Eğer
                           kayıtlı değilseniz, bilgilerinizle birlikte sizi kaydedeceğiz. Eğer
                           kayıtlı bir üyemizseniz bilgilerinizi güncelleyebileceksiniz.</p>
                  <br />
                  <p style="color: red">
                           <h:outputLabel value="#{OgrenciBean.mesaj}"></h:outputLabel>
                  </p>
                  <h:form>
                           <h:panelGrid columns="2">
                                    <h:outputText value="Kullanıcı adı:"></h:outputText>
                                    <h:inputText value="#{OgrenciBean.user}"></h:inputText>
                                    <h:outputText value="Şifre (sehven):"></h:outputText>
                                    <h:inputSecret value="#{OgrenciBean.password}"></h:inputSecret>
                           </h:panelGrid>
                           <h:commandButton value="Yallah"
                                    action="#{OgrenciBean.isOgrenciValid}"></h:commandButton>
                  </h:form>
         </ui:define>
</ui:composition>
</html>
```

loginSirket.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
         xmlns:ui="http://java.sun.com/jsf/facelets"
         xmlns:h="http://java.sun.com/jsf/html"
         xmlns:f="http://java.sun.com/jsf/core"
         xmlns:p="http://java.sun.com/jsp/jstl/core">

<ui:composition template="/WEB-INF/templates/StandartTema.xhtml">
         <ui:define name="content">
                  <p:if test="${SirketBean.s != null}">İyi de oturum açmışsınız? <a
                                    href="sirket.xhtml">Sizi öğrenci aramaya yönlendirelim.</a>
                  </p:if>
                  <p:if test="${SirketBean.s == null}">

                           <p>Merhaba! Lütfen kullanıcı adı ve şifrenizi giriniz. Hatalı
                                    şifre girerseniz sürüleceksiniz.</p>
                           <br />
                           <p style="color: red">
                                    <h:outputLabel value="#{SirketBean.mesaj}"></h:outputLabel>
                           </p>
                           <h:form>
                                    <h:panelGrid columns="2">
                                             <h:outputText value="Kullanıcı adı:"></h:outputText>
                                             <h:inputText value="#{SirketBean.user}"></h:inputText>
                                             <h:outputText value="Şifre (sehven):"></h:outputText>
                                             <h:inputSecret value="#{SirketBean.password}"></h:inputSecret>
                                    </h:panelGrid>
                                    <h:commandButton value="Yallah" ac-
tion="#{SirketBean.isSirketValid}"></h:commandButton>
                           </h:form>
                  </p:if>
```

```
        </ui:define>
</ui:composition>
</html>
```

ogrenci.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:ui="http://java.sun.com/jsf/facelets"
        xmlns:h="http://java.sun.com/jsf/html"
        xmlns:f="http://java.sun.com/jsf/core">

<ui:composition template="/WEB-INF/templates/StandartTema.xhtml">

        <ui:define name="content">
                <p style="color: red">
                        <h:outputLabel value="#{OgrenciBean.mesaj}"></h:outputLabel>
                </p>

                <h:form>
                        <h:panelGrid columns="2">
                                <h:outputText value="Kullanıcı adı:"></h:outputText>
                                <h:inputText value="#{OgrenciBean.user}"></h:inputText>
                                <h:outputText value="Şifre (sehven):"></h:outputText>
                                <h:inputSecret value="#{OgrenciBean.password}"></h:inputSecret>
                                <h:outputText value="Gerçek ad:"></h:outputText>
                                <h:inputText value="#{OgrenciBean.name}"></h:inputText>
                                <h:outputText value="GPA:"></h:outputText>
                                <h:inputText value="#{OgrenciBean.gpa}"></h:inputText>
                                <h:outputText value="İlgi alanları:"></h:outputText>
                                <h:inputText value="#{OgrenciBean.interests}"></h:inputText>
                        </h:panelGrid>

                        <h:commandButton value="Yallah" action="#{OgrenciBean.updateOgrenci}"></h:commandButton>
                </h:form>
        </ui:define>
</ui:composition>
</html>
```

roleSelection.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:ui="http://java.sun.com/jsf/facelets"
        xmlns:h="http://java.sun.com/jsf/html"
        xmlns:f="http://java.sun.com/jsf/core">

<ui:composition template="/WEB-INF/templates/StandartTema.xhtml">
        <ui:define name="content">
                Aşağıdan rolünüzü seçerek başlayabilirsiniz.
        </ui:define>
</ui:composition>
</html>
```

secililer.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:ui="http://java.sun.com/jsf/facelets"
        xmlns:h="http://java.sun.com/jsf/html"
        xmlns:f="http://java.sun.com/jsf/core"
        xmlns:p="http://java.sun.com/jsp/jstl/core">

<ui:composition template="/WEB-INF/templates/StandartTema.xhtml">
        <ui:define name="content">
                <h:dataTable id="itemsTable"
                        value="#{SirketBean.seciliOgrenciListesi}" var="item">

                        <f:facet name="header">
                                <h:outputText value="Öğrenciler" />
                        </f:facet>
                        <h:column>
                                <f:facet name="header">
                                        <h:outputText value="Öğrenci Adı" />
                                </f:facet>
                                <h:outputText value="#{item.o.name}"></h:outputText>
                        </h:column>
                        <h:column>
```

```
                                        <f:facet name="header">
                                                <h:outputText value="GPA" />
                                        </f:facet>
                                        <h:outputText value="#{item.o.gpa}"></h:outputText>
                                </h:column>
                                <h:column>
                                        <f:facet name="header">
                                                <h:outputText value="İlgi Alanarı" />
                                        </f:facet>
                                        <h:outputText value="#{item.o.interests}"></h:outputText>
                                </h:column>
                        </h:dataTable>
                </ui:define>
</ui:composition>
</html>
```

sirket.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
          xmlns:ui="http://java.sun.com/jsf/facelets"
          xmlns:h="http://java.sun.com/jsf/html"
          xmlns:f="http://java.sun.com/jsf/core"
          xmlns:p="http://java.sun.com/jsp/jstl/core">

<ui:composition template="/WEB-INF/templates/StandartTema.xhtml">

          <ui:define name="content">
                    <p:if test="${SirketBean.s == null}">Ama önce oturum açsaydınız?</p:if>
                    <p:if test="${SirketBean.s != null}">
                              <p style="color: red">
                                        <h:outputLabel value="#{SirketBean.mesaj}"></h:outputLabel>
                              </p>

                              <h:form>
                                        <h:panelGrid columns="3">
                                                  <h:outputText value="Ada göre ara:"></h:outputText>
                                                  <h:inputText value="#{SirketBean.searchUser}"></h:inputText>
                                                  <h:commandButton value="Yallah" ac-
tion="#{SirketBean.searchByUser}"></h:commandButton>
                                                  <h:outputText value="İlgi alanına göre ara:"></h:outputText>

                                                  <h:inputText value="#{SirketBean.searchInterest}"></h:inputText>
                                                  <h:commandButton value="Yallah"
                                                            action="#{SirketBean.searchByInterest}" />

                                        </h:panelGrid>Not: Eğer arama kriterini değiştirirseniz, önceki stajiyer seçimle-
riniz sıfırlanacaktır. Eğer arama filtresini kaldırmak istiyorsanız, boşluk aramanız yeterli olacaktır.<br />
                                        <br />
                                        <h:commandButton value="Tüm Seçilmişleri Görüntüle" action="goster" />
                                        <h:commandButton title="selectItems"
                                                  value="Seçilmişler Listesini Komple Boşalt"
                                                  action="#{SirketBean.sil}" />
                                        <h:form>
                                                  <h:dataTable id="itemsTable" value="#{SirketBean.ogrenciListesi}"
                                                            var="item">

                                                            <f:facet name="header">
                                                                      <h:outputText value="Öğrenciler" />
                                                            </f:facet>
                                                            <h:column>
                                                                      <f:facet name="header">
                                                                                <h:outputText value="Seç" />
                                                                      </f:facet>
                                                                      <h:selectBooleanCheckbox value="#{item.secili}" />
                                                            </h:column>
                                                            <h:column>
                                                                      <f:facet name="header">
                                                                                <h:outputText value="Öğrenci Adı" />
                                                                      </f:facet>
                                                                      <h:outputText value="#{item.o.name}"></h:outputText>
                                                            </h:column>
                                                            <h:column>
                                                                      <f:facet name="header">
                                                                                <h:outputText value="GPA" />
                                                                      </f:facet>
                                                                      <h:outputText value="#{item.o.gpa}"></h:outputText>
                                                            </h:column>
                                                            <h:column>
                                                                      <f:facet name="header">
                                                                                <h:outputText value="İlgi Alanarı" />
```

```
                                                                </f:facet>
                                                                <h:outputText va-
lue="#{item.o.interests}"></h:outputText>
                                                        </h:column>
                                                </h:dataTable>

                                                <h:commandButton title="selectItems" value="Seçilileri Seç"
                                                        action="#{SirketBean.guncelle}" />
                                        </h:form>

                                </h:form>
                        </p:if>
                </ui:define>

</ui:composition>
</html>
```

faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<faces-config
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
    version="2.0">
        <managed-bean>
                <managed-bean-name>OgrenciBean</managed-bean-name>
                <managed-bean-class>com.ubenzer.sst.proje2.beans.OgrenciBean</managed-bean-class>
                <managed-bean-scope>session</managed-bean-scope>
        </managed-bean>
        <managed-bean>
                <managed-bean-name>SirketBean</managed-bean-name>
                <managed-bean-class>com.ubenzer.sst.proje2.beans.SirketBean</managed-bean-class>
                <managed-bean-scope>session</managed-bean-scope>
        </managed-bean>
        <navigation-rule>
                <display-name>Login Öğrenci</display-name>
                <from-view-id>/loginOgrenci.xhtml</from-view-id>
                <navigation-case>
                        <from-action>#{OgrenciBean.isOgrenciValid}</from-action>
                        <from-outcome>valid</from-outcome>
                        <to-view-id>/ogrenci.xhtml</to-view-id>
                        <redirect />
                </navigation-case>
                <navigation-case>
                        <from-action>#{OgrenciBean.isOgrenciValid}</from-action>
                        <from-outcome>invalid</from-outcome>
                        <to-view-id>/loginOgrenci.xhtml</to-view-id>
                        <redirect />
                </navigation-case>
        </navigation-rule>
        <navigation-rule>
                <display-name>Update / Insert Öğreci</display-name>
                <from-view-id>/ogrenci.xhtml</from-view-id>
                <navigation-case>
                        <from-action>#{OgrenciBean.updateOgrenci}</from-action>
                        <from-outcome>error</from-outcome>
                        <to-view-id>/ogrenci.xhtml</to-view-id>
                        <redirect />
                </navigation-case>
                <navigation-case>
                        <from-action>#{OgrenciBean.updateOgrenci}</from-action>
                        <from-outcome>success</from-outcome>
                        <to-view-id>/ogrenci.xhtml</to-view-id>
                        <redirect />
                </navigation-case>
        </navigation-rule>
        <navigation-rule>
                <display-name>Login Şirket</display-name>
                <from-view-id>/loginSirket.xhtml</from-view-id>
                <navigation-case>
                        <from-action>#{SirketBean.isSirketValid}</from-action>
                        <from-outcome>valid</from-outcome>
                        <to-view-id>/sirket.xhtml</to-view-id>
                        <redirect />
                </navigation-case>
                <navigation-case>
                        <from-action>#{SirketBean.isSirketValid}</from-action>
                        <from-outcome>invalid</from-outcome>
                        <to-view-id>/loginSirket.xhtml</to-view-id>
                        <redirect />
                </navigation-case>
```

```
	</navigation-rule>
	<navigation-rule>
		<display-name>Şirket İşlem Sayfası</display-name>
		<from-view-id>/sirket.xhtml</from-view-id>
		<navigation-case>
			<from-action>#{SirketBean.guncelle}</from-action>
			<from-outcome>success</from-outcome>
			<to-view-id>/sirket.xhtml</to-view-id>
			<redirect />
		</navigation-case>
	</navigation-rule>
	<navigation-rule>
		<display-name>Şirket İşlem Sayfası</display-name>
		<from-view-id>/sirket.xhtml</from-view-id>
		<navigation-case>
			<from-action>#{SirketBean.sil}</from-action>
			<from-outcome>success</from-outcome>
			<to-view-id>/sirket.xhtml</to-view-id>
			<redirect />
		</navigation-case>
	</navigation-rule>
	<navigation-rule>
		<display-name>Şirket İşlem Sayfası</display-name>
		<from-view-id>/sirket.xhtml</from-view-id>
		<navigation-case>
			<from-outcome>goster</from-outcome>
			<to-view-id>/secililer.xhtml</to-view-id>
			<redirect />
		</navigation-case>
	</navigation-rule>
</faces-config>
```