**package** com.me.JobPortal.controller;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpSession;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.servlet.ModelAndView;

**import** com.me.JobPortal.pojo.JobList;

**import** com.me.JobPortal.pojo.User;

@Controller

@RequestMapping("/authorize/\*")

**public** **class** AuthorizationController {

@RequestMapping(value = "/authorize/error.htm", method = RequestMethod.***GET***)

**protected** ModelAndView addjobs(HttpServletRequest request) **throws** Exception {

ModelAndView mv = **new** ModelAndView();

mv.setViewName("errorpage");

**return** mv;

}

@RequestMapping(value = "/authorize/JobUpdateError.htm", method = RequestMethod.***GET***)

**protected** ModelAndView errorjobs(HttpServletRequest request) **throws** Exception {

ModelAndView mv = **new** ModelAndView();

mv.setViewName("error-jobupdate");

**return** mv;

}

}

**package** com.me.JobPortal.controller;

**import** java.util.List;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpSession;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.beans.factory.annotation.Qualifier;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.validation.BindingResult;

**import** org.springframework.web.bind.WebDataBinder;

**import** org.springframework.web.bind.annotation.InitBinder;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.servlet.ModelAndView;

**import** com.me.JobPortal.DAO.CompanyDAO;

**import** com.me.JobPortal.DAO.JobsDAO;

**import** com.me.JobPortal.DAO.UserDAO;

**import** com.me.JobPortal.exception.CompanyException;

**import** com.me.JobPortal.pojo.Company;

**import** com.me.JobPortal.pojo.JobList;

**import** com.me.JobPortal.pojo.User;

**import** com.me.JobPortal.validator.CompanyValidator;

@Controller

@RequestMapping("/company/\*")

**public** **class** CompanyController {

@Autowired

@Qualifier("companyDao")

CompanyDAO companyDao;

@Autowired

@Qualifier("companyValidator")

CompanyValidator validator;

@Autowired

@Qualifier("jobsDao")

JobsDAO jobsDao;

@Autowired

@Qualifier("userDao")

UserDAO userDao;

@InitBinder

**private** **void** initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}

@RequestMapping(value = "/company/register.htm", method = RequestMethod.***POST***)

**protected** ModelAndView registerNewUser(HttpServletRequest request, @ModelAttribute("company") Company company, BindingResult result) **throws** Exception {

validator.validate(company, result);

**if** (result.hasErrors()) {

**return** **new** ModelAndView("register-company", "company", company);

}

**try** {

System.***out***.print("registerNewCompany");

Company c = companyDao.register(company);

**return** **new** ModelAndView("company-add", "company", c);

} **catch** (CompanyException e) {

System.***out***.println("Exception: " + e.getMessage());

**return** **new** ModelAndView("error", "errorMessage", "error while registering company");

}

}

@RequestMapping(value = "/company/register.htm", method = RequestMethod.***GET***)

**protected** ModelAndView registerUser() **throws** Exception {

System.***out***.print("registerUser");

**return** **new** ModelAndView("register-company", "company", **new** Company());

}

@RequestMapping(value = "/company/delete.htm", method = RequestMethod.***POST***)

**protected** ModelAndView searchjobs(HttpServletRequest request) **throws** Exception {

ModelAndView mv = **new** ModelAndView();

HttpSession session = (HttpSession) request.getSession();

User u = (User) session.getAttribute("user");

**int** compId = Integer.*parseInt*(request.getParameter("compID"));

Company comp = companyDao.getById(compId);

// List<JobList> jobs = jobsDao.listCompany(comp);

companyDao.delete(comp);

mv.addObject("companies", companyDao.list());

mv.setViewName("admin-home");

**return** mv;

}

}

package com.me.JobPortal.controller;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.hibernate.HibernateException;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.http.HttpRequest;

import org.springframework.mail.SimpleMailMessage;

import org.springframework.mail.javamail.JavaMailSender;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.me.JobPortal.DAO.CompanyDAO;

import com.me.JobPortal.DAO.JobsDAO;

import com.me.JobPortal.DAO.UserDAO;

import com.me.JobPortal.exception.JobsException;

import com.me.JobPortal.pojo.Company;

import com.me.JobPortal.pojo.JobList;

import com.me.JobPortal.pojo.User;

import com.me.JobPortal.validator.JobsValidator;

@Controller

@RequestMapping("/jobs/\*")

public class JobsController {

@Autowired

@Qualifier("jobsDao")

JobsDAO jobsDao;

@Autowired

@Qualifier("userDao")

UserDAO userDao;

@Autowired

@Qualifier("companyDao")

CompanyDAO companyDao;

@Autowired

@Qualifier("jobsValidator")

JobsValidator validator;

@Autowired

private JavaMailSender mailSender;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}

@RequestMapping(value = "/jobs/register.htm", method = RequestMethod.GET)

protected ModelAndView addjobs(HttpServletRequest request) throws Exception {

ModelAndView mv = new ModelAndView();

HttpSession session = (HttpSession) request.getSession();

User u = (User) session.getAttribute("user");

mv.addObject("companyName", u.getCompany().getCompanyName());

mv.addObject("joblist", new JobList());

mv.setViewName("job-add");

return mv;

}

@RequestMapping(value = "/jobs/register.htm", method = RequestMethod.POST)

protected ModelAndView registerNewUser(HttpServletRequest request, @ModelAttribute("joblist") JobList joblist,

BindingResult result) throws Exception {

validator.validate(joblist, result);

if (result.hasErrors()) {

return new ModelAndView("job-add", "joblist", joblist);

}

try {

ModelAndView mv = new ModelAndView();

HttpSession session = (HttpSession) request.getSession();

User u = (User) session.getAttribute("user");

System.out.print("Add New Job");

joblist.setCompany(u.getCompany());

joblist.setPostedName(u.getFirstName());

joblist.setPostedID(u.getUsername());

JobList job = jobsDao.register(joblist);

mv.addObject("jobs", jobsDao.listCompany(u.getCompany()));

mv.addObject("joblist", job);

mv.setViewName("employer-home");

return mv;

} catch (JobsException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while adding job");

}

}

@RequestMapping(value = "/jobs/update.htm", method = RequestMethod.GET)

protected ModelAndView updatejobs(HttpServletRequest request) throws Exception {

ModelAndView mv = new ModelAndView();

HttpSession session = (HttpSession) request.getSession();

int id = Integer.parseInt(request.getParameter("jobId"));

JobList job = jobsDao.getJob(id);

System.out.println("THE PRODUCT ID IS :" + id);

User u = (User) session.getAttribute("user");

mv.addObject("companyName", u.getCompany().getCompanyName());

// mv.addObject("joblist", new JobList());

mv.addObject("jobList", job);

mv.setViewName("job-update");

return mv;

}

@RequestMapping(value = "/jobs/update.htm", method = RequestMethod.POST)

protected ModelAndView updatejob(HttpServletRequest request) throws Exception {

ModelAndView mv = new ModelAndView();

HttpSession session = (HttpSession) request.getSession();

int id = Integer.parseInt(request.getParameter("jobId"));

JobList job = jobsDao.getJob(id);

job.setJobName(request.getParameter("jobName"));

job.setJobDesc(request.getParameter("jobDesc"));

System.out.println("THE PRODUCT ID IS :" + id);

jobsDao.update(job);

User u = (User) session.getAttribute("user");

// mv.addObject("companyName", u.getCompany().getCompanyName());

// mv.addObject("joblist", new JobList());

mv.addObject("jobList", job);

mv.setViewName("update-success");

return mv;

}

@RequestMapping(value = "/jobs/modify.htm", method = RequestMethod.POST)

protected ModelAndView modifyJob(HttpServletRequest request, @ModelAttribute("joblist") JobList joblist,

BindingResult result) throws Exception {

// validator.validate(joblist, result);

// if (result.hasErrors()) {

// return new ModelAndView("job-add", "joblist", joblist);

// }

try {

ModelAndView mv = new ModelAndView();

String action = request.getParameter("action");

int job\_id = Integer.parseInt(request.getParameter("productID"));

System.out.println("the action is : " + action);

if (action.equals("Delete")) {

JobList job = jobsDao.getJob(job\_id);

jobsDao.deleteJob(job);

mv.setViewName("employer-home");

}

return mv;

// HttpSession session = (HttpSession) request.getSession();

// User u = (User) session.getAttribute("user");

// System.out.print("Add New Job");

// joblist.setCompany(u.getCompany());

//

// JobList job = jobsDao.register(joblist);

// mv.addObject("jobs", jobsDao.listCompany(u.getCompany()));

// mv.addObject("joblist", job);

// mv.setViewName("employer-home");

// return mv;

} catch (JobsException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while adding job");

}

}

@RequestMapping(value = "/jobs/add.htm", method = RequestMethod.POST)

protected ModelAndView addJob(HttpServletRequest request) throws Exception {

try {

boolean flag = false;

ModelAndView mv = new ModelAndView();

HttpSession session = (HttpSession) request.getSession();

User u = (User) session.getAttribute("user");

int job\_id = Integer.parseInt(request.getParameter("jobID"));

for (JobList job1 : u.getJoblists()) {

if (job1.getJobID() == job\_id) {

flag = true;

}

}

System.out.print("Add New Job");

if (!flag) {

JobList job = jobsDao.getJob(job\_id);

// u.getJoblists().add(job);

userDao.update(u, job);

System.out.println("Entered the job addition new para");

job.getUsers().add(u);

jobsDao.update(job);

String address = u.getEmail().getEmailAddress();

String subject = "Job Applied Notification for " +job.getJobName();

String message = "Dear "+u.getFirstName()+","+"\n"+"\nYou Applied for the position : "+job.getJobName()+"\n"+"\nJob Desc : " +job.getJobDesc()

+"\non : " +new Date() +"\n"+"\n"+"\nThank you for using our online Job Portal"+"\n"+"\nRegards"+"\nAdmin";

SimpleMailMessage email = new SimpleMailMessage();

email.setTo(address);

email.setSubject(subject);

email.setText(message);

mailSender.send(email);

mv.addObject("jobs", jobsDao.listCompany(u.getCompany()));

mv.addObject("joblist", job);

mv.setViewName("add-success");

flag = false;

return mv;

} else {

System.out.println("Entered the other para where its already added");

mv.addObject("jobs", null);

mv.setViewName("add-already");

return mv;

}

} catch (JobsException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while adding job");

}

}

@RequestMapping(value = "/jobs/search.htm", method = RequestMethod.GET)

protected ModelAndView searchjob(HttpServletRequest request) throws Exception {

ModelAndView mv = new ModelAndView();

HttpSession session = (HttpSession) request.getSession();

User u = (User) session.getAttribute("user");

// mv.addObject("companyName", u.getCompany().getCompanyName());

// mv.addObject("joblist", new JobList());

mv.setViewName("search-job");

return mv;

}

@RequestMapping(value = "/jobs/search.htm", method = RequestMethod.POST)

protected ModelAndView searchjobs(HttpServletRequest request) throws Exception {

ModelAndView mv = new ModelAndView();

HttpSession session = (HttpSession) request.getSession();

User u = (User) session.getAttribute("user");

String name = request.getParameter("inputtext");

String search = request.getParameter("searchkey");

if (search.equalsIgnoreCase("jobname")) {

List<JobList> jobs = jobsDao.listJobname(name);

mv.addObject("jobs", jobs);

if (jobs.isEmpty())

System.out.println("The returned list is empty");

for (JobList job : jobs) {

System.out.println("The matching jobs are : " + job.getJobName());

}

} else if (search.equalsIgnoreCase("company")) {

Company c = companyDao.get(name);

if (c == null) {

mv.setViewName("search-fail");

return mv;

}

List<JobList> jobs = jobsDao.listCompany(c);

if (jobs.isEmpty()) {

mv.setViewName("search-failjobs");

return mv;

}

mv.addObject("jobs", jobs);

}

mv.setViewName("search-success");

return mv;

}

@RequestMapping(value = "/jobs/applied.htm", method = RequestMethod.GET)

protected ModelAndView appliedjob(HttpServletRequest request) throws Exception {

ModelAndView mv = new ModelAndView();

// HttpSession session = (HttpSession) request.getSession();

// User u = (User) session.getAttribute("user");

// mv.addObject("companyName", u.getCompany().getCompanyName());

// mv.addObject("joblist", new JobList());

mv.setViewName("jobs-applied");

return mv;

}

@RequestMapping(value = "/jobs/viewalljobs.htm", method = RequestMethod.GET)

protected ModelAndView viewAllJob(HttpServletRequest request) throws Exception {

ModelAndView mv = new ModelAndView();

// HttpSession session = (HttpSession) request.getSession();

// User u = (User) session.getAttribute("user");

// mv.addObject("companyName", u.getCompany().getCompanyName());

// mv.addObject("joblist", new JobList());

mv.addObject("jobs", jobsDao.list());

mv.setViewName("viewalljobs");

return mv;

}

}

package com.me.JobPortal.controller;

import java.util.Set;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.me.JobPortal.DAO.CompanyDAO;

import com.me.JobPortal.DAO.JobsDAO;

import com.me.JobPortal.DAO.UserDAO;

import com.me.JobPortal.exception.UserException;

import com.me.JobPortal.pojo.Company;

import com.me.JobPortal.pojo.JobList;

import com.me.JobPortal.pojo.User;

import com.me.JobPortal.validator.UserValidator;

@Controller

@RequestMapping("/user/\*")

public class UserController {

@Autowired

@Qualifier("userDao")

UserDAO userDao;

@Autowired

@Qualifier("userValidator")

UserValidator validator;

@Autowired

@Qualifier("companyDao")

CompanyDAO companyDao;

@Autowired

@Qualifier("jobsDao")

JobsDAO jobsDao;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}

@RequestMapping(value = "/user/login.htm", method = RequestMethod.GET)

protected String goToUserHome(HttpServletRequest request) throws Exception {

return "home";

}

@RequestMapping(value = "/user/login.htm", method = RequestMethod.POST)

protected ModelAndView loginUser(HttpServletRequest request) throws Exception {

ModelAndView mv = new ModelAndView();

HttpSession session = (HttpSession) request.getSession();

try {

System.out.print("loginUser");

User u = userDao.get(request.getParameter("username"), request.getParameter("password"));

if(u == null){

System.out.println("UserName/Password does not exist");

session.setAttribute("errorMessage", "UserName/Password does not exist");

mv.setViewName("error");

return mv;

}

session.setAttribute("user", u);

String role = u.getRole().trim();

if (role.equals("employee")){

System.out.println("Entered here");

// Set<JobList> job = u.getJoblists();

// System.out.println("The email addrees of Employee : " +u.getFirstName()+" is "+u.getEmail().getEmailAddress());

mv.addObject("jobs", jobsDao.list());

mv.setViewName("employee-home");

return mv ;

}

else if (role.equals("employer")){

// System.out.println("Entered in the region");

// System.out.println("The company id of the user is : " +u.getCompany().getCompanyId());

// mv.addObject("company", u.getCompany().getCompanyName());

// System.out.println("The email addrees of Employer : " +u.getFirstName()+" is "+u.getEmail().getEmailAddress());

mv.addObject("jobs", jobsDao.listCompany(u.getCompany()));

mv.setViewName("employer-home");

return mv;

}

else if(role.equals("admin")){

// System.out.println("The email addrees of Admin : " +u.getFirstName()+" is "+u.getEmail().getEmailAddress());

mv.addObject("companies", companyDao.list());

mv.setViewName("admin-home");

return mv;

}

else

return null;

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

session.setAttribute("errorMessage", "error while login");

mv.setViewName("error");

return mv;

}

}

@RequestMapping(value = "/user/register.htm", method = RequestMethod.GET)

protected ModelAndView registerUser() throws Exception {

System.out.print("registerUser");

ModelAndView mv = new ModelAndView();

mv.addObject("companies", companyDao.list());

mv.addObject("user",new User());

mv.setViewName("register-user");

// return new ModelAndView("register-user", "user", new User());

return mv;

}

@RequestMapping(value = "/user/employee.htm", method = RequestMethod.GET)

protected ModelAndView employeeHome() throws Exception {

System.out.print("Employee Home");

ModelAndView mv = new ModelAndView();

mv.addObject("jobs", jobsDao.list());

mv.setViewName("employee-home");

// return new ModelAndView("register-user", "user", new User());

return mv;

}

@RequestMapping(value = "/user/employer.htm", method = RequestMethod.GET)

protected ModelAndView employerHome(HttpServletRequest request) throws Exception {

System.out.print("Employeer Home");

HttpSession session = (HttpSession) request.getSession();

User u = (User) session.getAttribute("user");

ModelAndView mv = new ModelAndView();

mv.addObject("jobs", jobsDao.listCompany(u.getCompany()));

mv.setViewName("employer-home");

return mv;

// return new ModelAndView("register-user", "user", new User());

}

@RequestMapping(value = "/user/admin.htm", method = RequestMethod.GET)

protected ModelAndView adminHome() throws Exception {

System.out.print("Admin Home");

ModelAndView mv = new ModelAndView();

mv.addObject("companies", companyDao.list());

mv.setViewName("admin-home");

// return new ModelAndView("register-user", "user", new User());

return mv;

}

@RequestMapping(value = "/user/register.htm", method = RequestMethod.POST)

protected ModelAndView registerNewUser(HttpServletRequest request, @ModelAttribute("user") User user, BindingResult result) throws Exception {

validator.validate(user, result);

if (result.hasErrors()) {

// return new ModelAndView("register-user", "user", user);

ModelAndView mv = new ModelAndView();

mv.addObject("companies", companyDao.list());

mv.addObject("user", user);

mv.setViewName("register-user");

return mv;

}

try {

System.out.print("registerNewUser");

String role = request.getParameter("myRole");

user.setRole(role);

String company = request.getParameter("myCompany");

System.out.println("The Company is :" +company);

Company c = companyDao.get(company);

System.out.println("The company ID : " +c.getCompanyId());

user.setCompany(c);

User u = userDao.register(user);

request.getSession().setAttribute("user", u);

return new ModelAndView("user-home", "user", u);

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

}

package com.me.JobPortal.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "companytable")

public class Company {

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

@Column(name="companyid")

private int companyId;

@Column(name="companyname")

private String companyName;

@Column(name="companyaddress")

private String companyAddress;

@Column(name="companydesc")

private String companyDescription;

public Company(){

}

public Company( String companyName, String companyAddress, String companyDescription){

this.companyName = companyName;

this.companyAddress = companyAddress;

this.companyDescription = companyDescription;

}

public int getCompanyId() {

return companyId;

}

public void setCompanyId(int companyId) {

this.companyId = companyId;

}

public String getCompanyName() {

return companyName;

}

public void setCompanyName(String companyName) {

this.companyName = companyName;

}

public String getCompanyAddress() {

return companyAddress;

}

public void setCompanyAddress(String companyAddress) {

this.companyAddress = companyAddress;

}

public String getCompanyDescription() {

return companyDescription;

}

public void setCompanyDescription(String companyDescription) {

this.companyDescription = companyDescription;

}

}

package com.me.JobPortal.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.PrimaryKeyJoinColumn;

import javax.persistence.Table;

import org.hibernate.annotations.Parameter;

import org.hibernate.annotations.GenericGenerator;

@Entity

@Table(name = "email\_table")

public class Email {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

// @GenericGenerator(name = "generator", strategy = "foreign", parameters = @Parameter(name = "property", value = "user"))

@Column(name = "emailID", unique = true)

private long id;

@Column(name = "email\_address")

private String emailAddress;

@OneToOne

@PrimaryKeyJoinColumn

private User user;

public Email() {

}

public Email(String emailAddress) {

this.emailAddress = emailAddress;

}

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getEmailAddress() {

return emailAddress;

}

public void setEmailAddress(String emailAddress) {

this.emailAddress = emailAddress;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

}

package com.me.JobPortal.pojo;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToOne;

import javax.persistence.OneToOne;

import javax.persistence.PrimaryKeyJoinColumn;

import javax.persistence.Table;

@Entity

@Table(name = "Employer\_Table")

@PrimaryKeyJoinColumn(name = "employeeID")

public class Employer extends User {

// @Id

// @GeneratedValue(strategy=GenerationType.AUTO)

// @Column(name = "employerID", unique=true, nullable = false)

// private int empID;

// @ManyToOne(cascade = CascadeType.ALL)

//// @Column(name = "company\_id")

private Company company;

public Employer(){

}

public Company getCompany() {

return company;

}

public void setCompany(Company company) {

this.company = company;

}

}

package com.me.JobPortal.pojo;

import java.util.Set;

import java.util.Date;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToMany;

import javax.persistence.ManyToOne;

import javax.persistence.PrimaryKeyJoinColumn;

import javax.persistence.Table;

import javax.persistence.Transient;

@Entity

@Table(name = "Jobs")

public class JobList {

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

@Column(name = "jobID", unique=true, nullable = false)

private int jobID;

@Column(name = "jobname")

private String jobName;

@Column(name = "jobDesc")

private String jobDesc;

@Column(name = "uname")

private String postedName;

@Column(name = "uId")

private String postedID;

@Column(name = "posteddate")

private Date postedDate;

@ManyToOne

@PrimaryKeyJoinColumn

private Company company;

@ManyToMany(mappedBy="joblists")

private Set<User> users;

public JobList(){

}

public JobList(String jobname, String jobdesc){

this.jobName = jobname;

this.jobDesc = jobdesc;

this.postedDate = new Date();

}

public int getJobID() {

return jobID;

}

public void setJobID(int jobID) {

this.jobID = jobID;

}

public Set<User> getUsers() {

return users;

}

public void setUsers(Set<User> users) {

this.users = users;

}

public String getJobName() {

return jobName;

}

public void setJobName(String jobName) {

this.jobName = jobName;

}

public String getJobDesc() {

return jobDesc;

}

public void setJobDesc(String jobDesc) {

this.jobDesc = jobDesc;

}

public Company getCompany() {

return company;

}

public void setCompany(Company company) {

this.company = company;

}

public String getPostedName() {

return postedName;

}

public void setPostedName(String postedName) {

this.postedName = postedName;

}

public String getPostedID() {

return postedID;

}

public void setPostedID(String postedID) {

this.postedID = postedID;

}

public Date getPostedDate() {

return postedDate;

}

public void setPostedDate(Date postedDate) {

this.postedDate = postedDate;

}

}

package com.me.JobPortal.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Inheritance;

import javax.persistence.InheritanceType;

import javax.persistence.Table;

@Entity

@Table(name="person\_table")

@Inheritance(strategy=InheritanceType.JOINED) //table per subclass

public class Person {

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

@Column(name = "personID", unique=true, nullable = false)

private long personID;

@Column(name = "firstName")

private String firstName;

@Column(name ="lastName")

private String lastName;

public Person(){

}

public long getPersonID() {

return personID;

}

public void setPersonID(long personID) {

this.personID = personID;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

}

package com.me.JobPortal.pojo;

import java.util.HashSet;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.Inheritance;

import javax.persistence.InheritanceType;

import javax.persistence.JoinTable;

import javax.persistence.ManyToMany;

import javax.persistence.ManyToOne;

import javax.persistence.OneToMany;

import javax.persistence.OneToOne;

import javax.persistence.PrimaryKeyJoinColumn;

import javax.persistence.Table;

import javax.persistence.JoinColumn;

import org.springframework.beans.factory.annotation.Autowired;

@Entity

@Table(name = "user\_table")

@PrimaryKeyJoinColumn(name = "personID")

//@Inheritance(strategy=InheritanceType.JOINED)

public class User extends Person {

@Column(name = "userName")

private String username;

@Column(name = "password")

private String password;

@OneToOne(cascade = CascadeType.ALL)

private Email email;

@Column(name = "role")

private String role;

@ManyToOne

private Company company;

@ManyToMany

@JoinTable (

name="user\_job\_table",

joinColumns = {@JoinColumn(name="personID", nullable = false, updatable = false)},

inverseJoinColumns = {@JoinColumn(name="jobID" )}

)

private Set<JobList> joblists = new HashSet<JobList>();

public User(String username, String password, String role) {

this.username = username;

this.password = password;

this.role = role;

}

public User() {

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public Email getEmail() {

return email;

}

public void setEmail(Email email) {

this.email = email;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public Company getCompany() {

return company;

}

public void setCompany(Company company) {

this.company = company;

}

public Set<JobList> getJoblists() {

return joblists;

}

public void setJoblists(Set<JobList> joblists) {

this.joblists = joblists;

}

}