

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	7
ВСТУП	8
ВИСНОВКИ	10
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА	11
ДОДАТКИ	12
БІБЛІОГРАФІЧНА ДОВІДКА	35

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка програмного алгоритмічного забезпечення багатофункціональної корпоративної системи для сумісної роботи, управління документами і проектами</i> <i>ПОЯСНЮВАЛЬНА ЗАПИСКА</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Розробив</i>	<i>Бойчук Я.В.</i>						6	35
<i>Перевірив</i>	<i>Випасняк Л.І.</i>					<i>ІНФТУНГ ПЗ-07-1</i>		
<i>Т.контр</i>								
<i>Н.контр</i>	<i>Вовк Р.Б.</i>							
<i>Затв.</i>	<i>Юрчишин В.М.</i>							

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

СКБД – Система керування базою даних

API – Application programming interface

JPA – Java Persistence API

SQL – Structured Query Language

GUI – Graphical User Interface

UML – Unified Modelling Language

MCC(MVCC) – Multiversion concurrency control

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

В нас час важливою економічною точкою опори для будь-якої комерційної організації є наявність певних факторів, які визнають чітку позицію компанії на ринку. До всіх цих чинників можна віднести багато варіантів, зокрема:

- капітал підприємства;
- матеріальна база;
- технічна база;
- кваліфікований персонал;
- місце підприємства на внутрішньому і зовнішньому ринку;
- наявність сучасних засобів виробництва та ведення бізнесу;
- та інші.

Тому кожний розділ ведення бізнесу повинний бути детально розглянутий та впроваджений у життя. Але якщо подивитися із точки зору програмного забезпечення, то на даному етапі розвитку цивілізації, якісне ПЗ відіграє напевно найбільш важливу роль. Адже не можливо зараз утримувати всі дані в паперовому вигляді, не можливо відсилати друковані листи, чи спілкуватися тільки по телефоні і взнавати новини компанії тільки при зустрічі. У наш стрімкий час розвитку, новини міняються із колосальною швидкістю, тому встигнути за всім просто не можливо без певного програмного продукту. Уявіть собі інформатор, який сповіщає будь-які для Вас новини чи корисну інформацію в зручний для Вас час, при цьому вміє фільтрувати і аналізувати дані із попередніх запитів. Також на даний момент важко уявити не можливість спільної роботи над документами, над електронними таблицями. Дані технології вже давно використовуються людьми і підприємствами, починаючи від найменших де працює двоє людей, до величезних із кількістю працівників більше ста тисяч. Але для цього всього використовуються дуже багато технологій, які важко налаштувати і потребують великих витрат на підтримку. Тому було розроблено багато сервісів і додатків, які полегшують роботу в мережі для підприємств.

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		8

Дане ПЗ використовується у всіх нішах нашого життя, починаючи від шкіл і лікарень, закінчуючи величезними корпораціями з будівництва космічних кораблів. Тому розробити універсальний продукт, який забезпечить всі вимоги, просто не можливо. Для кожної сфери існують свої нюанси.

Цікавою нішею для дослідження стало корпоративне програмне забезпечення для малого і середнього бізнесу. На даний момент існує багато програмних продуктів для комерційних цілей, проте вони здебільшого розраховані на великі корпорації і підприємства. Тому використання їх для менших фірм просто не доцільно, або дуже складно із фінансової сторони (витрати на підтримку передують вигоді). Як відомо, на ринку до цих пір зберігається тенденція на попит на корпоративне програмне забезпечення, яке б відповідало вимогам малого і середнього бізнесу, і в той же час було практично придатним для використання у великих корпоративних цілях

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Завдяки сучасним технологіям і корпоративним стандартам, розвиток розробки комерційних продуктів виріс дуже стрімко. Зокрема сюди і відноситься відносно молодий напрямок — це розробка корпоративних порталів. Було встановлено стандарти щодо розробки додатків і аплікацій – це допомогло добитися легкої інтеграції і взаємодії. Також проведено аналіз сучасного стану і потреб ринку в даній сфері, наведено всі вимоги до програмного продукту. Проведено аналіз щодо економічної вигоди та прораховано всі важливі аспекти щодо охорони праці.

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. <http://www.jcp.org/en/jsr/detail?id=286> - стандарт портлетів Java Portlet 2.0
2. <http://www.jcp.org/en/jsr/detail?id=168> - стандарт портлетів Java Portlet 1.0
3. <http://google.com> - пошук доступної в інтернеті інформації
4. <http://tomcat.apache.org/tomcat-5.5-doc/> – специфікація серлетів
5. <http://pz.nung.edu.ua/> - сайт кафедри ПЗАС
6. <http://www.intranetno.ru/> - бізнес рішення на базі SaaS, PaaS
7. http://en.wikipedia.org/wiki/Entity-attribute-value_model - EAV модель

					<i>ДП.ПЗ 04.00.00.000 ПЗ</i>	Аркуш
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

ДОДАТОК А

Продовження додатку А

Домен для мапінгу даних

```
package com.diploma.ccms.domain;
import java.util.Collection;
import java.util.Date;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.EntityManager;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.ManyToOne;
import javax.persistence.PersistenceContext;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.TypedQuery;
import javax.persistence.Version;
import javax.validation.constraints.NotNull;
import org.apache.commons.lang3.builder.ReflectionToStringBuilder
    ;
import org.apache.commons.lang3.builder.ToStringStyle;
import org.springframework.beans.factory.annotation.Configurable;
import org.springframework.format.annotation.DateTimeFormat;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.context.
    SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.transaction.annotation.Transactional;
import com.diploma.ccms.domain.service.DomainUserInterface;
```



```

@Entity
@Configurable
public class Worker implements UserDetails, DomainUserIntarface {
    private static final long serialVersionUID = 1148968754347580432L
        ;
    @NotNull
    @Column(unique = true)
    private String login;
    @NotNull
    private String pass;
    @NotNull
    private String name;
    @NotNull
    private String surname;
    private String phone;
    private String mobile;
    private String privateMail;
    private String street;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = dd/MM/yyyy)
    private Date birthday;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = dd/MM/yyyy)
    private Date dateHire;
    @ManyToOne
    private WorkerRole roleName;
    @NotNull
    @ManyToOne
    private Region regionName;
    @NotNull
    @ManyToOne
    private Team teamName;
    @NotNull
    @ManyToOne
    private WorkerJobType jobTypeName;
    @Lob
    private byte[] photo;

```

```
public static Worker getPrincipal(){  
return (Worker) SecurityContextHolder.getContext().  
    getAuthentication().getPrincipal();  
}  
public String getLogin() {  
return this.login;  
}  
public void setLogin(String login) {  
this.login = login;  
}  
public String getPass() {  
return this.pass;  
}  
public void setPass(String pass) {  
this.pass = pass;  
}  
public String getName() {  
return this.name;  
}  
public void setName(String name) {  
this.name = name;  
}  
public String getSurname() {  
return this.surname;  
}  
public void setSurname(String surname) {  
this.surname = surname;  
}  
public String getPhone() {  
return this.phone;  
}  
public void setPhone(String phone) {  
this.phone = phone;  
}  
public String getPrivateMail() {  
return this.privateMail;  
}
```

```
public void setPrivateMail(String privateMail) {  
    this.privateMail = privateMail;  
}  
  
public String getStreet() {  
    return this.street;  
}  
  
public void setStreet(String street) {  
    this.street = street;  
}  
  
public Date getBirthday() {  
    return this.birthday;  
}  
  
public void setBirthday(Date birthday) {  
    this.birthday = birthday;  
}  
  
public Date getDateHire() {  
    return this.dateHire;  
}  
  
  
public void setDateHire(Date dateHire) {  
    this.dateHire = dateHire;  
}  
  
public WorkerRole getRoleName() {  
    return this.roleName;  
}  
  
public void setRoleName(WorkerRole roleName) {  
    this.roleName = roleName;  
}  
  
public Region getRegionName() {  
    return this.regionName;  
}  
  
public void setRegionName(Region regionName) {  
    this.regionName = regionName;  
}  
  
public Team getTeamName() {  
    return this.teamName;  
}
```

```

public void setTeamName(Team teamName) {
this.teamName = teamName;
}

public WorkerJobType getJobTypeName() {
return this.jobTypeName;
}

public void setJobTypeName(WorkerJobType jobTypeName) {
this.jobTypeName = jobTypeName;
}

public byte[] getPhoto() {
return this.photo;
}

public void setPhoto(byte[] photo) {
this.photo = photo;
}

@PersistenceContext
transient EntityManager entityManager;

public static final EntityManager entityManager() {
EntityManager em = new Worker().entityManager;
if (em == null)
throw new IllegalStateException(Entity manager has not been
    injected (is the Spring Aspects JAR configured as an AJC/AJDT
    aspects library?));
return em;
}

public static long countWorkers() {
return entityManager().createQuery(SELECT COUNT(o) FROM Worker o,
    Long.class).getSingleResult();
}

public static List<Worker> findAllWorkers() {
return entityManager().createQuery(SELECT o FROM Worker o, Worker
    .class).getResultList();
}

public static Worker findWorker(Long id) {
if (id == null)
return null;
return entityManager().find(Worker.class , id);
}

```

```

}

public static List<Worker> findWorkerEntries(int firstResult, int
    maxResults) {
return entityManager().createQuery(SELECT o FROM Worker o, Worker
    .class).setFirstResult(firstResult).setMaxResults(maxResults).
    getResultList();
}

@Transactional
public void persist() {
if (this.entityManager == null)
this.entityManager = entityManager();
this.entityManager.persist(this);
}

@Transactional
public void remove() {
if (this.entityManager == null)
this.entityManager = entityManager();
if (this.entityManager.contains(this)) {
this.entityManager.remove(this);
} else {
Worker attached = Worker.findWorker(this.id);
this.entityManager.remove(attached);
}
}

@Transactional
public void flush() {
if (this.entityManager == null)
this.entityManager = entityManager();
this.entityManager.flush();
}

@Transactional
public void clear() {
if (this.entityManager == null)
this.entityManager = entityManager();
this.entityManager.clear();
}

@Transactional

```

```

public Worker merge() {
if (this.entityManager == null)
this.entityManager = entityManager();
Worker merged = this.entityManager.merge(this);
this.entityManager.flush();
return merged;
}

public String toString() {
return ReflectionToStringBuilder.toString(this, ToStringStyle.
    SHORT_PREFIX_STYLE);
}

public static TypedQuery<Worker> findWorkersByLoginEquals(String
    login) {
if (login == null || login.length() == 0)
throw new IllegalArgumentException(The login argument is required
    );
EntityManager em = Worker.entityManager();
TypedQuery<Worker> q = em.createQuery(SELECT o FROM Worker AS o
    WHERE o.login = :login, Worker.class);
q.setParameter(login, login);
return q;
}

@Id
@GeneratedValue(strategy = GenerationType.AUTO)
@Column(name = id)
private Long id;

@Version
@Column(name = version)
private Integer version;

public Long getId() {
return this.id;
}

public void setId(Long id) {
this.id = id;
}

public Integer getVersion() {
return this.version;
}

```

```
}  
  
public void setVersion(Integer version) {  
    this.version = version;  
}  
  
@Override  
public Collection<? extends GrantedAuthority> getAuthorities() {  
    Set<GrantedAuthority> roles = new HashSet<GrantedAuthority>();  
    roles.add(roleName);  
    return roles;  
}  
  
@Override  
public String getPassword() {  
    return pass;  
}  
  
@Override  
public String getUsername() {  
    return login;  
}  
  
@Override  
public boolean isAccountNonExpired() {  
    return true;  
}  
  
@Override  
public boolean isAccountNonLocked() {  
    return true;  
}  
  
@Override  
public boolean isCredentialsNonExpired() {  
    return true;  
}  
  
@Override  
public boolean isEnabled() {  
    return true;  
}  
  
public String getUiValue() {  
    return name +      + surname;  
}
```

```

public String getMobile() {
return mobile;
}

public void setMobile(String mobile) {
this.mobile = mobile;
}
}

```

Контроллер даних

```

package com.diploma.ccms.web.controller;

import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.validation.Valid;

import org.joda.time.format.DateTimeFormat;
import org.springframework.context.i18n.LocaleContextHolder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.util.UriUtils;
import org.springframework.web.util.WebUtils;

import com.diploma.ccms.domain.Region;
import com.diploma.ccms.domain.Team;
import com.diploma.ccms.domain.Worker;
import com.diploma.ccms.domain.WorkerJobType;
import com.diploma.ccms.domain.WorkerRole;

```



```

@RequestMapping("/workers")
@Controller
public class WorkerController {

    @RequestMapping(method = RequestMethod.POST, produces = "text/html")
    public String create(@Valid Worker worker, BindingResult bindingResult, Model uiModel, HttpServletRequest httpServletRequest) {
        if (bindingResult.hasErrors()) {
            populateEditForm(uiModel, worker);
            return "workers/create";
        }
        uiModel.asMap().clear();
        worker.persist();
        uiModel.addAttribute("menu", "WORKER");
        return "redirect:/workers/" + encodeUrlPathSegment(worker.getId()).toString(), httpServletRequest);
    }

    @RequestMapping(params = "form", produces = "text/html")
    public String createForm(Model uiModel) {
        populateEditForm(uiModel, new Worker());
        List<String[]> dependencies = new ArrayList<String[]>();
        if (Region.countRegions() == 0) {
            dependencies.add(new String[] { "region", "regions" });
        }
        if (Team.countTeams() == 0) {
            dependencies.add(new String[] { "team", "teams" });
        }
        if (WorkerJobType.countWorkerJobTypes() == 0) {
            dependencies.add(new String[] { "workerjobtype", "workerjobtypes" });
        }
        uiModel.addAttribute("dependencies", dependencies);
        uiModel.addAttribute("menu", "WORKER");
        return "workers/create";
    }

```

```
}
```

```
@RequestMapping(value =("/{id}", produces = "text/html")
public String show(@PathVariable("id") Long id, Model uiModel) {
    addDateTimeFormatPatterns(uiModel);
    uiModel.addAttribute("worker", Worker.findWorker(id));
    uiModel.addAttribute("itemId", id);
    uiModel.addAttribute("menu", "WORKER");
    return "workers/show";
}
```

```
@RequestMapping(produces = "text/html")
public String list(@RequestParam(value = "page", required = false
    ) Integer page, @RequestParam(value = "size", required = false
    ) Integer size, Model uiModel) {
    //          if (page != null || size != null) {
    //              int sizeNo = size == null ? 10 : size.intValue();
    //              final int firstResult = page == null ? 0 : (page.
    //                  intValue() - 1) * sizeNo;
    //              uiModel.addAttribute("workers", Worker.
    //                  findWorkerEntries(firstResult, sizeNo));
    //              float nrOfPages = (float) Worker.countWorkers() /
    //                  sizeNo;
    //              uiModel.addAttribute("maxPages", (int) ((nrOfPages
    //                  > (int) nrOfPages || nrOfPages == 0.0) ? nrOfPages + 1 :
    //                  nrOfPages));
    //          } else {
    uiModel.addAttribute("workers", Worker.findAllWorkers());
    //          }
    uiModel.addAttribute("menu", "WORKER");
    addDateTimeFormatPatterns(uiModel);
    return "workers/list";
}
```

```
@RequestMapping(method = RequestMethod.PUT, produces = "text/html
    ")
```

```

public String update(@Valid Worker worker, BindingResult
    bindingResult, Model uiModel, HttpServletRequest
    httpRequest) {
    if (bindingResult.hasErrors()) {
    populateEditForm(uiModel, worker);
    uiModel.addAttribute("menu", "WORKER");
    return "workers/update";
    }
    uiModel.asMap().clear();
    worker.merge();
    uiModel.addAttribute("menu", "WORKER");
    return "redirect:/workers/" + encodeUrlPathSegment(worker.getId()
        .toString(), httpRequest);
    }

```

```

@RequestMapping(value =("/{id}", params = "form", produces = "
    text/html")

```

```

public String updateForm(@PathVariable("id") Long id, Model
    uiModel) {
    populateEditForm(uiModel, Worker.findWorker(id));
    uiModel.addAttribute("menu", "WORKER");
    return "workers/update";
    }

```

```

@RequestMapping(value =("/{id}", method = RequestMethod.DELETE,
    produces = "text/html")

```

```

public String delete(@PathVariable("id") Long id, @RequestParam(
    value = "page", required = false) Integer page,
    @RequestParam(value = "size", required = false) Integer size,
    Model uiModel) {
    Worker worker = Worker.findWorker(id);
    worker.remove();
    uiModel.asMap().clear();
    uiModel.addAttribute("page", (page == null) ? "1" : page.toString
        ());
    uiModel.addAttribute("size", (size == null) ? "10" : size.
        toString());

```

```
uiModel.addAttribute("menu", "WORKER");  
return "redirect:/workers";  
}
```

```
void addDateTimeFormatPatterns(Model uiModel) {  
uiModel.addAttribute("worker_birthday_date_format",  
    DateTimeFormat.patternForStyle("M-", LocaleContextHolder.  
        getLocale()));  
uiModel.addAttribute("worker_datehire_date_format",  
    DateTimeFormat.patternForStyle("M-", LocaleContextHolder.  
        getLocale()));  
}
```

```
void populateEditForm(Model uiModel, Worker worker) {  
uiModel.addAttribute("worker", worker);  
addDateTimeFormatPatterns(uiModel);  
uiModel.addAttribute("regions", Region.findAllRegions());  
uiModel.addAttribute("teams", Team.findAllTeams());  
uiModel.addAttribute("workerjobtypes", WorkerJobType.  
    findAllWorkerJobTypes());  
uiModel.addAttribute("workerroles", WorkerRole.findAllWorkerRoles  
    ());  
uiModel.addAttribute("menu", "WORKER");  
}
```

```
String encodeUrlPathSegment(String pathSegment,  
    HttpServletRequest httpRequest) {  
String enc = httpRequest.getCharacterEncoding();  
if (enc == null) {  
enc = WebUtils.DEFAULT_CHARACTER_ENCODING;  
}  
try {  
pathSegment = UriUtils.encodePathSegment(pathSegment, enc);  
} catch (UnsupportedEncodingException uee) {  
}  
return pathSegment;  
}
```

```

@RequestMapping(params = { "find=ByLoginEquals", "form" }, method
    = RequestMethod.GET)
public String findWorkersByLoginEqualsForm(Model uiModel) {
uiModel.addAttribute("menu", "WORKER");
return "workers/findWorkersByLoginEquals";
}

@RequestMapping(params = "find=ByLoginEquals", method =
    RequestMethod.GET)
public String findWorkersByLoginEquals(@RequestParam("login")
    String login, Model uiModel) {
uiModel.addAttribute("workers", Worker.findWorkersByLoginEquals(
    login).getResultList());
uiModel.addAttribute("menu", "WORKER");
return "workers/list";
}

/**
 * Generate preview for e-mail
 *
 * @param petId
 * @param model
 * @return
 */
@RequestMapping(value = "/preview_email", method = RequestMethod.
    POST)
@ResponseBody
public String getPreview(@RequestParam("data") String data) {
return data;
}

}

```

Сервіс для збереження даних

```

package com.diploma.ccms.service;

```

```

import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.
    UserDetailsService;
import org.springframework.security.core.userdetails.
    UsernameNotFoundException;

import com.diploma.ccms.domain.Worker;

public class UserService implements UserDetailsService {

@Override
public UserDetails loadUserByUsername(String username) throws
    UsernameNotFoundException {
return Worker.findWorkersByLoginEquals(username).getSingleResult
    ();
}

}

```

Інтерфейс для збереження даних користувача

```

package com.diploma.ccms.domain.service;

public interface DomainUserIntarface {
    /**
     * used for getting value for showing on UI form
     */
    public String getUiValue();
}

```

Конфігурація виглядів для контролера

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<tiles-definitions>
<definition extends="default" name="workers/list">
<put-attribute name="body" value="/WEB-INF/views/workers/list.
    jsp" />

```

```

</definition>
<definition extends="default" name="workers/show">
<put-attribute name="body" value="/WEB-INF/views/workers/show.
    jsp" />
</definition>
<definition extends="default" name="workers/create">
<put-attribute name="body" value="/WEB-INF/views/workers/create.
    jsp" />
</definition>
<definition extends="default" name="workers/update">
<put-attribute name="body" value="/WEB-INF/views/workers/update.
    jsp" />
</definition>
<definition extends="default" name="workers/
    findWorkersByLoginEquals">
<put-attribute name="body"
value="/WEB-INF/views/workers/findWorkersByLoginEquals.jsp" />
</definition>
</tiles-definitions>

```

Вигляд створення для користувача

```

<?xml version=1.0 encoding=UTF-8 standalone=no?>
<div xmlns:c=http://java.sun.com/jsp/jstl/core xmlns:field=urn:
    jsptagdir:/WEB-INF/tags/form/fields xmlns:form=urn:jsptagdir:/
    WEB-INF/tags/form
xmlns:jsp=http://java.sun.com/JSP/Page xmlns:spring=http://www.
    springframework.org/tags version=2.0>
<jsp:directive.page contentType=text/html;charset=UTF-8 />
<jsp:output omit-xml-declaration=yes />

<section class=main-section grid_7>

<div class=main-content>
<header>
<h2>Employees</h2>
</header>
<section class=container_6 clearfix>

```

```
<field:input field=login id=
  c_com_diploma_ccms_domain_Worker_login required=true label=
  Login smallLabel=Enter login type=text/>
<field:input field=pass id=c_com_diploma_ccms_domain_Worker_pass
  required=true label=Password smallLabel=Enter password type=
  password/>
<field:input field=name id=c_com_diploma_ccms_domain_Worker_name
  required=true label=Name smallLabel=Enter name type=text/>
<field:input field=surname id=
  c_com_diploma_ccms_domain_Worker_surname required=true label=
  Surname smallLabel=Enter surname type=text/>
<field:input field=mobile id=
  c_com_diploma_ccms_domain_Worker_mobile required=false label=
  Mobile phone number smallLabel=Private phone number type=text
  />
<field:input field=privateMail id=
  c_com_diploma_ccms_domain_Worker_privateMail required=false
  label=Email smallLabel=A valid email address type=email/>
<field:select field=regionName id=
  c_com_diploma_ccms_domain_Worker_regionName items=${regions}
  required=true label=Region smallLabel=Please select/>
<field:input field=street id=
  c_com_diploma_ccms_domain_Worker_street required=false label=
  Address smallLabel=Enter full home address type=text/>
<field:input field=birthday id=
  c_com_diploma_ccms_domain_Worker_birthday required=false label=
  Birthdate smallLabel=mm/dd/yyyy type=date/>
<field:input field=dateHire id=
  c_com_diploma_ccms_domain_Worker_dateHire required=false label=
  Hire Date smallLabel=mm/dd/yyyy type=date/>
<field:select field=roleName id=
  c_com_diploma_ccms_domain_Worker_roleName items=${workerroles}
  required=true label=Employee role smallLabel=Please select
  role/>
<field:select field=teamName id=
  c_com_diploma_ccms_domain_Worker_teamName items=${teams}
  required=true label=Employee related team smallLabel=Please
```



```

        select working team/>
<field:select field=jobTypeName id=
    c_com_diploma_ccms_domain_Worker_jobTypeName items=${
    workerjobtypes} required=true label=Job type smallLabel=Like
    QA, Developer, etc../>
<field:input field=phone id=
    c_com_diploma_ccms_domain_Worker_phone required=false label=
    Phone number smallLabel=Working office phone number type=text
    />
<field:textarea field=photo id=
    c_com_diploma_ccms_domain_Worker_photo />
</form:create>
</section>
</div>
</section>
</div>

```

Вигляд для показу даних про

```

<?xml version=1.0 encoding=UTF-8 standalone=no?>
<div xmlns:field=urn:jsptagdir:/WEB-INF/tags/form/fields
xmlns:c=http://java.sun.com/jsp/jstl/core
xmlns:sec=http://www.springframework.org/security/tags
xmlns:jsp=http://java.sun.com/JSP/Page xmlns:page=urn:jsptagdir:/
    WEB-INF/tags/form version=2.0>
<jsp:directive.page contentType=text/html;charset=UTF-8 />
<jsp:output omit-xml-declaration=yes />
<!-- Main Section -->
<section class=main-section grid_7>
<div class=main-content grid_4 alpha>
<header class=clearfix>
<span class=avatar> <!-- -->
</span>
<hgroup>
<h2>${worker.name} ${worker.surname}</h2>
<h4>
<a href=#>${worker.jobTypeName.jobTypeName}</a>
</h4>

```

```

</hgroup>
<p class=tags>
<!-- <a href=#>Add To Contacts</a> -->
</p>
</header>
<section>
<script type=text/javascript>
    $(document).ready(function() {
        // Add markItUp! to your textarea in one line
        $('.markItUpTextarea').markItUp(mySettings, {
            root : 'markitup/skins/simple/'
        });
    });
</script>
<style>
.markItUp {
width: 390px;
}
</style>
<sec:authentication var=principal_login property=principal.login
/>
<c:choose>
<c:when test=${worker.login eq principal_login}>
<h3> Have a nice day ;) </h3>
</c:when>
<c:otherwise>
<h3>Send private message to ${worker.name} ${worker.surname}</h3>
<form method=post action=/ccms/messages?id=${worker.id} style=
width: 400px;>
<field:input id=c_com_diploma_ccms_domain_Message_title field=
title required=true label=Title smallLabel= type=text/>
<textarea id=c_com_diploma_ccms_domain_Message_text name=text
class=markItUpTextarea style=height: 80px; width: 100%;
required=required><!-- --></textarea>
<button class=fr button button-gray type=submit>Send message</
button>
</form>

```

```

</c:otherwise>
</c:choose>
<div class=clear>
<!-- -->
</div>
<!-- <h3>History</h3>
<ul class=listing list-view>
<li class=note><a href=editnote.html class=more></a> <span class=
    timestamp>Dec 28, 2010</span>
<p>Vestibulum ultrices vehicula leo ac tristique. Mauris id nisl
    nibh.</p>
<div class=entry-meta>Posted by Administrator</div></li>
<li class=note><a href=editnote.html class=more></a> <span class=
    timestamp>Dec 28, 2010</span>
<p>Vestibulum ultrices vehicula leo ac tristique. Mauris id nisl
    nibh.</p>
<div class=entry-meta>Posted by Administrator</div></li>
<li class=note><a href=editnote.html class=more></a> <span class=
    timestamp>Dec 28, 2010</span>
<p>Vestibulum ultrices vehicula leo ac tristique. Mauris id nisl
    nibh.</p>
<div class=entry-meta>Posted by Administrator</div></li>
</ul> -->
</section>
</div>
<div class=preview-pane grid_3 omega>
<div class=content>
<h3>${worker.name}s contact information</h3>
<ul class=profile-info>
<li class=email><c:if test=${empty worker.privateMail}>:: none
    ::</c:if> ${worker.privateMail}<span>email</span></li>
<li class=mobile><c:if test=${empty worker.mobile}>:: none ::</c:
    if>${worker.mobile}<span>mobile</span></li>
<li class=phone><c:if test=${empty worker.phone}>:: none ::</c:if
    >${worker.phone}<span>work</span></li>
</ul>
<!-- <h3>Tasks About John</h3>

```

```
None so far. <a href=#>Add a task now</a> -->
<h3>Additional info</h3>
<ul class=profile-info>
<li class=calendar-day><c:if test=${empty worker.birthday}>::
    none ::</c:if>${worker.birthday}<span>birthday</span></li>
<li class=calendar-day><c:if test=${empty worker.dateHire}>::
    none ::</c:if>${worker.dateHire}<span>hire date</span></li>
<li class=building><c:if test=${empty worker.regionName.
    regionName}>:: none ::</c:if>${worker.regionName.regionName}<
    span>living region</span></li>
<li class=house><c:if test=${empty worker.street}>:: none ::</c:
    if>${worker.street}<span>home address</span></li>
</ul>
</div>
<div class=preview>
<!-- -->
</div>
</div>
</section>
<!-- Main Section End -->
<!-- <page:show id=ps_com_diploma_ccms_domain_Worker object=${
    worker}
path=/workers z=iQamgfMz7tfHSdGgWB8YCdDd83Q=>
<field:display field=login
id=s_com_diploma_ccms_domain_Worker_login object=${worker}
z=4HjNYZAHZPDLTqADXwCdyd4GRAk= />
<field:display field=pass id=
    s_com_diploma_ccms_domain_Worker_pass
object=${worker} z=HYhzipM8Xx937WviE05t1DOTtIbU= />
<field:display field=name id=
    s_com_diploma_ccms_domain_Worker_name
object=${worker} z=egBKl4f8HsCgN5Ynj/jGjRUelAo= />
<field:display field=surname
id=s_com_diploma_ccms_domain_Worker_surname object=${worker}
z=ES6rDyTeRfbtJ4oZ9zGg6RpkdIO= />
<field:display field=phone
id=s_com_diploma_ccms_domain_Worker_phone object=${worker}
```

```
z=1VohSBGktVWKME/kgylZA99MluU= />
<field:display field=privateMail
id=s_com_diploma_ccms_domain_Worker_privateMail object=${worker}
z=JZug5Vfq4gKYkQsNndLyunI+pLo= />
<field:display field=street
id=s_com_diploma_ccms_domain_Worker_street object=${worker}
z=e/Rkpq+iZVtm5lYsnThOISuQXRA= />
<field:display date=true
dateTimePattern=${worker_birthday_date_format} field=birthday
id=s_com_diploma_ccms_domain_Worker_birthday object=${worker}
z=EmRENzOD8cefRn6vszpdLPB3+04= />
<field:display date=true
dateTimePattern=${worker_datehire_date_format} field=dateHire
id=s_com_diploma_ccms_domain_Worker_dateHire object=${worker}
z=oQ0LtnMFCGhHLh6m2nvl4LtMrJU= />
<field:display field=roleName
id=s_com_diploma_ccms_domain_Worker_roleName object=${worker}
z=wJnkS5H0d6iZ3AnuqTU07TI2wX8= />
<field:display field=regionName
id=s_com_diploma_ccms_domain_Worker_regionName object=${worker}
z=obgn4zC59pMlQ8GaXlDJYjUcoGs= />
<field:display field=teamName
id=s_com_diploma_ccms_domain_Worker_teamName object=${worker}
z=iRDO/HwFc7JQ+f+sLvRXQwEzBOw= />
<field:display field=jobTypeName
id=s_com_diploma_ccms_domain_Worker_jobTypeName object=${worker}
z=a43X3qenfQB/KwXlhCUeDmDGIYM= />
<field:display field=photo
id=s_com_diploma_ccms_domain_Worker_photo object=${worker}
z=lcI4umkwCdlmQFaIzB4gefboNPI= />
</page:show>
-->
</div>
```

БІБЛІОГРАФІЧНА ДОВІДКА

ТЕМА ДИПЛОМНОГО ПРОЕКТУ: «Розробка програмного та алгоритмічного забезпечення багатофункціональної корпоративної системи для спільної роботи, управління документами і проектами засобами Java EE»

Обсяг пояснювальної записки: 36 аркуша

Дата закінчення проекту 4 травня 2012 р.

Підпис студента-дипломника _____