

Informatizarea procesului de recrutare

Echipa: **ULBS12**

Membrii:

1. Burghilea Elimelec Doriner, grupa 232
2. Dragoi Alexandra Maria, grupa 231
3. Matei Alexandru, grupa 231
4. Modranga Cristina, grupa 232
5. Raulea Mihaela - Nicoleta, grupa 231

A. Introducere

a) User stories

User story	Implementat
DIRECTOR GENERAL	
Ca director general, vreau sa pot sa resping orice candidat. Nu ca nu mi-ar placea de el sau asa ceva.	X
Ca director general, eu vreau sa stiu tot ce se intampla in firma asta, sa imi vina mail cu toate update-urile. Ar fi super daca as avea si buton de respingere sau acceptare direct in mail.	
Ca director general, vreau sa pot sa accept orice candidat. Nu ma gandesc la rude sau prieteni. Nici prin gand nu mi-a trecut.	X
Ca director general, vreau sa accept sau sa resping un anunt pentru o pozitie	
DIRECTOR DE DEPARTAMENT	
Ca director de departament, vreau sa pot vedea toti candidatii care au aplicat la o pozitie pentru departamentul meu.	X
Ca director de departament, vreau sa accept sau sa resping un candidat care a aplicat la o pozitie pentru departamentul meu.	X
Ca director de departament, vreau sa adaug o pozitie pentru departamentul meu.	X
Ca director de departament, vreau sa inchid sau sa redeschid o pozitie a departamentul meu.	X
Ca director de departament, vreau sa modific numarul de locuri disponibile pentru o pozitie din departamentul meu.	X
Ca director de departament, vreau sa editez o pozitie a departamentul meu.	X
Ca director de departament, vreau sa fiu notificat daca cineva aplica la o pozitie in departamentul meu si daca are calificarea necesara.	X

DIRECTOR HR	
Ca director HR, vreau sa pot accesa oricand cv-ul cuiva.	X
Ca director HR, vreau sa adaug comentariu la cv-ul candidatului pentru a-mi ajuta la intrebarile pe care le voi pune la interviu.	X
ADMINISTRATOR	
Ca si administrator de sistem vreau sa pot sa adaug, sterg sau sa editez utilizatorii site-ului.	X
RECRUITER	
Ca recruiter, vreau sa vad un cv oricand.	X
Ca si recruiter vreau sa stiu, primind mail despre toate schimbarile ce apar in legatura cu pozitiile de care ma ocup.	
Ca si recruiter vreau sa pot scrie comentarii de bine sau de rau in privinta unui candidat.	X
VIZITATOR	
Ca si vizitator al acestui site, vreau sa pot sa vizualizez toate pozitiile deschise si sa aplic la cate pozitii vreau eu.	X

b) Operatii CRUD

Operatii CRUD	Director general	Director de department	Director HR	Recruiter	Admin	Viewer
Adaugare user	X				X	
Editare user	X				X	
Stergere user	X				X	
Adaugare pozitie	X	X	X	X		
Editare pozitie	X	X	X	X		
Deschidere pozitie		X				
Inchidere pozitie	X					
Adaugare candidat	X			X		
Editare candidat	X			X		
Stergere candidat	X			X		
Adaugare comentarii pentru pozitie	X	X	X	X		
Adaugare comentarii pentru candidat	X	X	X	X		

c) Implementari

Membru 1:

Operatii CRUD	Director general	Director de department	Director HR	Recruiter	Admin	Viewer
Adaugare user	X				X	
Editare user	X				X	
Stergere user	X				X	
Adaugare pozitie	X	X	X	X		
Editare pozitie	X	X	X	X		
Deschidere pozitie		X				
Inchidere pozitie	X					
Adaugare candidat	X			X		
Editare candidat	X			X		
Stergere candidat	X			X		
Adaugare comentarii pentru pozitie	X	X	X	X		
Adaugare comentarii pentru candidat	X	X	X	X		

Membru 3:

Operatii CRUD	Director general	Director de department	Director HR	Recruiter	Admin	Viewer
Adaugare user	X				X	
Editare user	X				X	
Stergere user	X				X	
Adaugare pozitie	X	X	X	X		
Editare pozitie	X	X	X	X		
Deschidere pozitie		X				
Inchidere pozitie	X					
Adaugare candidat	X			X		
Editare candidat	X			X		
Stergere candidat	X			X		
Adaugare comentarii pentru pozitie	X	X	X	X		
Adaugare comentarii pentru candidat	X	X	X	X		

Membru 5:

Operatii CRUD	Director general	Director de department	Director HR	Recruiter	Admin	Viewer
Adaugare user	X				X	
Editare user	X				X	
Stergere user	X				X	
Adaugare pozitie	X	X	X	X		
Editare pozitie	X	X	X	X		
Deschidere pozitie		X				
Inchidere pozitie	X					
Adaugare candidat	X			X		
Editare candidat	X			X		
Stergere candidat	X			X		
Adaugare comentarii pentru pozitie	X	X	X	X		
Adaugare comentarii pentru candidat	X	X	X	X		

B. Use cases

Author:	Membru 1 & Membrul 3
Use case:	Administratorul vizualizeaza lista cu utilizatori
Preconditions:	Administratorul este logat
Postconditions:	Administratorul vede o lista cu utilizatori
Trigger:	Dupa logare
Main scenario:	<ol style="list-style-type: none"> 1. Administratorul se logheza 2. Administratorul este dus pe pagina cu lista utilizatorilor 3. Administratorul vede toti utilizatorii cu emailul si rolul lor
Alternative flows:	-
Author:	Membru 1 & Membrul 3
Use case:	Administratorul sterge un utilizator
Preconditions:	Administratorul este pe pagina cu lista utilizatorilor
Postconditions:	Un utilizator este sters din sistem
Trigger:	Administratorul apasa pe butonul „Sterge Utilizator”
Main scenario:	<ol style="list-style-type: none"> 1. Administratorul apasa pe butonul „Sterge Utilizator” 2. Utilizatorul dispare din lista
Alternative flows:	-
Author:	Membru 1 & Membrul 3

Use case:	Administratorul adauga un utilizator
Preconditions:	Administratorul este pe pagina cu lista utilizatorilor
Postconditions:	Un utilizator nou este adaugat in sistem
Trigger:	Administratorul apasa pe butonul „Adauga Utilizator”
Main scenario:	<ol style="list-style-type: none"> 1. Administratorul apasa pe butonul „Adauga Utilizator” 2. Administratorul completeaza datele necesare 3. Administratorul apasa pe butonul „Adauga” 4. Administratorul este redirectionat catre pagina cu lista utilizatorilor 5. Noul utilizator apare in lista
Alternative flows:	-
Author:	Membru 1 & Membrul 3
Use case:	Administratorul modifica un utilizator
Preconditions:	Administratorul este pe pagina cu lista utilizatorilor
Postconditions:	Un utilizator din sistem este modificat
Trigger:	Administratorul apasa pe butonul „Modifica Utilizator”
Main scenario:	<ol style="list-style-type: none"> 1. Administratorul apasa pe butonul „Modifica Utilizator” 2. Administratorul modifica datele necesare 3. Administratorul este redirectionat catre pagina cu lista utilizatorilor 4. Utilizatorul este modificat
Alternative flows:	-
Author:	Membru 1
Use case:	Administratorul se logheaza
Preconditions:	Administratorul este pe site
Postconditions:	Administratorul este dus catre panoul de administrare
Trigger:	Administratorul este pe prima pagina a site-ului
Main scenario:	<ol style="list-style-type: none"> 1. Administratorul introduce emailul 2. Administratorul introduce parola 3. Administratorul apasa pe „Logare”
Alternative flows:	-
Author:	Membru 1
Use case:	Utilizatorul se inchide sesiunea
Preconditions:	Utilizatorul este logat pe site
Postconditions:	Utilizatorul nu mai este logat pe site
Trigger:	Utilizatorul apasa pe butonul „Logout”
Main scenario:	<ol style="list-style-type: none"> 1. Utilizatorul apasa pe butonul „Logout” 2. Utilizatorul este redirectionat catre pagina de logare
Alternative flows:	-
Author:	Membru 1
Use case:	Directorul de departament (candidatul, recruiterul) adauga un

	comentariu la o pozitie
Preconditions:	Directorul de departament se afla in lista cu comentarii la o pozitie
Postconditions:	Un nou comentariu este adaugat la pozitie
Trigger:	Directorul de departament are ceva de comentat
Main scenario:	<ol style="list-style-type: none"> 1. Directorul de departament da click in casuta pentru introdus comentariul 2. Directorul de departament tasteaza comentariul 3. Directorul de departament apasa pe butonul „Adauga” 4. Un nou comentariu este adaugat
Alternative flows:	-
Author:	Membru 1
Use case:	Directorul de departament (candidatul, recruiterul) vede comentariile la o pozitie
Preconditions:	Directorul de departament se afla in lista cu pozitii
Postconditions:	Directorul de departament citeste comentariile
Trigger:	Directorul de departament vrea sa citeasca comentariile
Main scenario:	<ol style="list-style-type: none"> 1. Directorul de departament apasa „Vezi comentariile” in lista cu pozitii 2. Directorul vede lista cu comentarii
Alternative flows:	-
Author:	Membru 2
Use case:	Directorul de departament adauga o pozitie
Preconditions:	Directorul a apasat butonul “Add position”
Postconditions:	Directorul este trimis la pagina pentru adaugare pozitie noua
Trigger:	Dupa apasarea butonului “Add position”
Main scenario:	<ol style="list-style-type: none"> 1. Directorul apasa butonul “Add position” 2. Directorul este redirectat la pagina destinata adaugarii unei pozitii 3. Directorul introduce numele pozitiei 4. Directorul introduce cerintele pozitiei 5. Directorul introduce responsabilitatile pozitiei 6. Directorul seteaza pozitia ca inchisa sau deschisa 7. Directorul introduce numarul de locuri disponibile pentru o pozitie 8. Directorul apasa butonul “Save”
Alternative flows:	-
Author:	Membru 2
Use case:	Directorul de departament vizualizeaza candidatii
Preconditions:	Directorul de departament apasa pe o pozitie
Postconditions:	Directorul vizualizeaza candidatii
Trigger:	Dupa apasare pe numele pozitiei
Main scenario:	<ol style="list-style-type: none"> 1. Directorul de departament intra in aplicatie.

	<ol style="list-style-type: none"> 2. Directorul de departament se logheaza. 3. Directorul alege o pozitie din departamentul lui. 4. Directorul priveste o lista cu toti candidatii la acea pozitie.
Alternative flows:	-
Author:	Membru 2
Use case:	Directorul de departament vizualizeaza cv-ul candidatului
Preconditions:	Directorul de departament apasa pe candidatul ales
Postconditions:	Directorul de departament vizualizeaza cv-ul
Trigger:	Dupa apasare pe numele candidatului
Main scenario:	<ol style="list-style-type: none"> 1. Directorul de departament intra in aplicatie. 2. Directorul de departament se logheaza. 3. Directorul alege un candidat. 4. Directorul priveste cv-ul candidatului
Alternative flows:	-
Author:	Membru 4
Use case:	Directorul de departament vizualizeaza o pozitie
Preconditions:	Directorul de departament apasa pe pozitia aleasa
Postconditions:	Directorul de departament vizualizeaza detaliile pozitiei
Trigger:	Dupa apasare pe pozitie
Main scenario:	<ol style="list-style-type: none"> 1. Directorul de departament intra in aplicatie 2. Directorul de departament se logheaza 3. Directorul de departament alege o pozitie 4. Directorul de departament poate apasa pe Editare Pozitie 5. Directorul de departament poate vizualiza toate CV-urile ce au aplicat pt acea pozitie 6. Directorul de departament poate vedea commenturi-le legate de pozitie
Alternative flows:	
Author:	Membru 4
Use case:	Directorul de departament face review pe un CV
Preconditions:	Directorul de departament apasa pe CV ales
Postconditions:	Directorul de departament vizualizeaza detaliile CV-ului si decide daca e acceptat sau respins
Trigger:	Dupa apasare pe CV
Main scenario:	<ol style="list-style-type: none"> 1. Directorul de departament intra in aplicatie 2. Directorul de departament se logheaza 3. Directorul de departament alege un CV 4. Directorul de departament poate apasa pe Accept, pentru a accepta CV-ul 5. Directorul de departament poate apasa pe Reject, pentru a respinge CV-ul 6. Directorul de departament poate vedea commenturi-le legate de CV

Alternative flows:	
Author:	Membru 5 & Membru 3
Use case:	Directorul de departament se logheaza
Preconditions:	Directorul este identificat si autentificat
Postconditions:	Exista pozitii adaugate
Trigger:	Dupa introducerea credentialelor
Main scenario:	<ol style="list-style-type: none"> 1. Directorul isi introduce email-ul si username-ul (ca si parola) 2. Directorul este logat 3. Directorul este trimis la pagina cu pozitii
Alternative flows:	<ol style="list-style-type: none"> 2.1. Directorul nu a introdus emailul si parola corecte <ol style="list-style-type: none"> 2.1.1 Sistemul arata o eroare 2.1.2 Directorul este nevoit sa se intoarca la pagina de login si sa-si reintroduce datele
Author:	Membru 5
Use case:	Directorul de departament vizualizeaza pozitiiile
Preconditions:	Directorul este autentificat
Postconditions:	Exista pozitii
Trigger:	-
Main scenario:	<ol style="list-style-type: none"> 1. Directorul poate vizualiza un tabel cu pozitiiile disponibile 2. Directorul poate edita o pozitie prin apasarea pe numele pozitiei 3. Directorul poate vedea comentariile unei pozitii existente prin apasarea pe "View comments" 4. Directorul poate adauga o pozitie noua prin apasarea butonului "Add position"
Alternative flows:	<ol style="list-style-type: none"> 1.1.Nu exista pozitii <ol style="list-style-type: none"> 1.1.2. Directorul nu poate efectua nicio operatiune asupra pozitiiilor intrucat nu exista
Author:	Membru 5
Use case:	Directorul de departament editeaza o pozitie
Preconditions:	Directorul a apasat pe numele pozitiei
Postconditions:	Directorul este trimis la pagina pentru editarea pozitiei
Trigger:	Dupa apasarea pe numele unei pozitii
Main scenario:	<ol style="list-style-type: none"> 1. Directorul apasa pe numele unei pozitii 2. Directorul este redirectat la pagina destinata editarii unei pozitii 3. Directorul introduce noul nume al pozitiei 4. Directorul introduce cerintele pozitiei 5. Directorul introduce noile responsabilitati pozitiei 6. Directorul seteaza pozitia ca inchisa sau deschisa 7. Directorul introduce numarul de locuri disponibile pentru pozitie 8. Directorul apasa butonul "Save"

Alternative flows:	-
Author:	Membru 5
Use case:	Directorul de departament inchide/deschide o pozitie
Preconditions:	Directorul a apasat pe numele pozitiei
Postconditions:	Directorul este trimis la pagina pentru editare pozitie
Trigger:	Dupa apasarea pe numele unei pozitii
Main scenario:	<ol style="list-style-type: none"> 1. Directorul apasa pe numele unei pozitii 2. Directorul este redirectat la pagina destinata editarii unei pozitii 3. Directorul seteaza pozitia ca inchisa sau deschisa 4. Directorul apasa butonul "Save"
Alternative flows:	-
Author:	Membru 5
Use case:	Directorul de departament schimba numarul de locuri disponibile
Preconditions:	Directorul a apasat pe numele pozitiei
Postconditions:	Directorul este trimis la pagina pentru editare pozitie
Trigger:	Dupa apasarea pe numele unei pozitii
Main scenario:	<ol style="list-style-type: none"> 1. Directorul apasa pe numele unei pozitii 2. Directorul este redirectat la pagina destinata editarii unei pozitii 3. Directorul introduce numarul de locuri disponibile pentru acea pozitie 4. Directorul apasa butonul "Save"
Alternative flows:	-
Author:	Membru 5
Use case:	Vizitatorul vizualizeaza pozitiiile disponibile
Preconditions:	Vizitatorul apasa pe "Go directly to available positions"
Postconditions:	Exista pozitii
Trigger:	Dupa apasarea pe "Go directly to available positions"
Main scenario:	<ol style="list-style-type: none"> 1. Vizitatorul poate vizualiza un tabel cu pozitiiile disponibile 2. Vizitatorul poate vizualiza cerintele unei pozitii prin apasarea pe numele pozitiei 3. Vizitatorul poate vedea comentariile unei pozitii existente prin apasarea pe "View comments"
Alternative flows:	<ol style="list-style-type: none"> 1.1.Nu exista pozitii 1.1.2. Vizitatorul nu poate efectua nicio operatiune asupra pozitiiilor intrucat nu exista
Author:	Membru 5
Use case:	Vizitatorul vizualizeaza cerintele unei pozitii
Preconditions:	Vizitatorul a apasat pe numele unei pozitii din tabelul tuturor pozitiiilor

Postconditions:	Exista pozitii disponibile
Trigger:	Dupa apasarea pe numele unei pozitii
Main scenario:	<ol style="list-style-type: none"> 1. Vizitatorul poate vede cerintele unei pozitii 2. Vizitatorul are posibilitatea de a aplica pentru o pozitie
Alternative flows:	<ol style="list-style-type: none"> 1.1.Nu exista pozitii 1.1.2. Vizitatorul nu poate efectua nicio operatiune asupra pozitiilor intrucat nu exista
Author:	Membru 5
Use case:	Vizitatorul aplica pentru o pozitie
Preconditions:	Vizitatorul a vazut cerintele pozitiei pentru care aplica
Postconditions:	Exista pozitia pentru care aplica
Trigger:	Dupa apasarea pe butonul “Apply for this job”
Main scenario:	<ol style="list-style-type: none"> 1. Vizitatorul apasa pe butonul “Apply for this job” 2. Vizitatorul este redirectat la pagina destinata introducerii datelor personale si a CV-ului 3. Vizitatorul isi introduce datele personale 4. Vizitatorul isi introduce CV-ul 5. Vizitatorul apasa pe butonul “Apply” 6. Vizitatorul este redirectat la pagina unde sunt vizibile toate pozitiile disponibile
Alternative flows:	
Author:	Membru 5
Use case:	Directorii de departament sunt notificati pe mail dupa ce un vizitator si-a adaugat CV-ul
Preconditions:	Un vizitator a aplicat pentru o pozitie disponibila
Postconditions:	-
Trigger:	Un vizitator a aplicat pentru o pozitie disponibila
Main scenario:	<ol style="list-style-type: none"> 1. Un vizitator aplica pentru o pozitie disponibila urmand pasii corespunzatori 2. Vizitatorul respectiv apasa pe butonul “Apply” 3. Toti directorii de departament sunt notificati ca un candidat a aplicat pentru o pozitie
Alternative flows:	-

C. Modelul de date

i. *Clasele Entity*

User:

Reprezinta un utilizator. Contine datele lui, email, parola, telefon. Tipul utilizatorului este salvat in coloana "permisiuni", care este si folosita automat de glassfish pentru restrictionarea accesului pe anumite pagini.

Position:

Informatii despre o pozitie libera din firma.

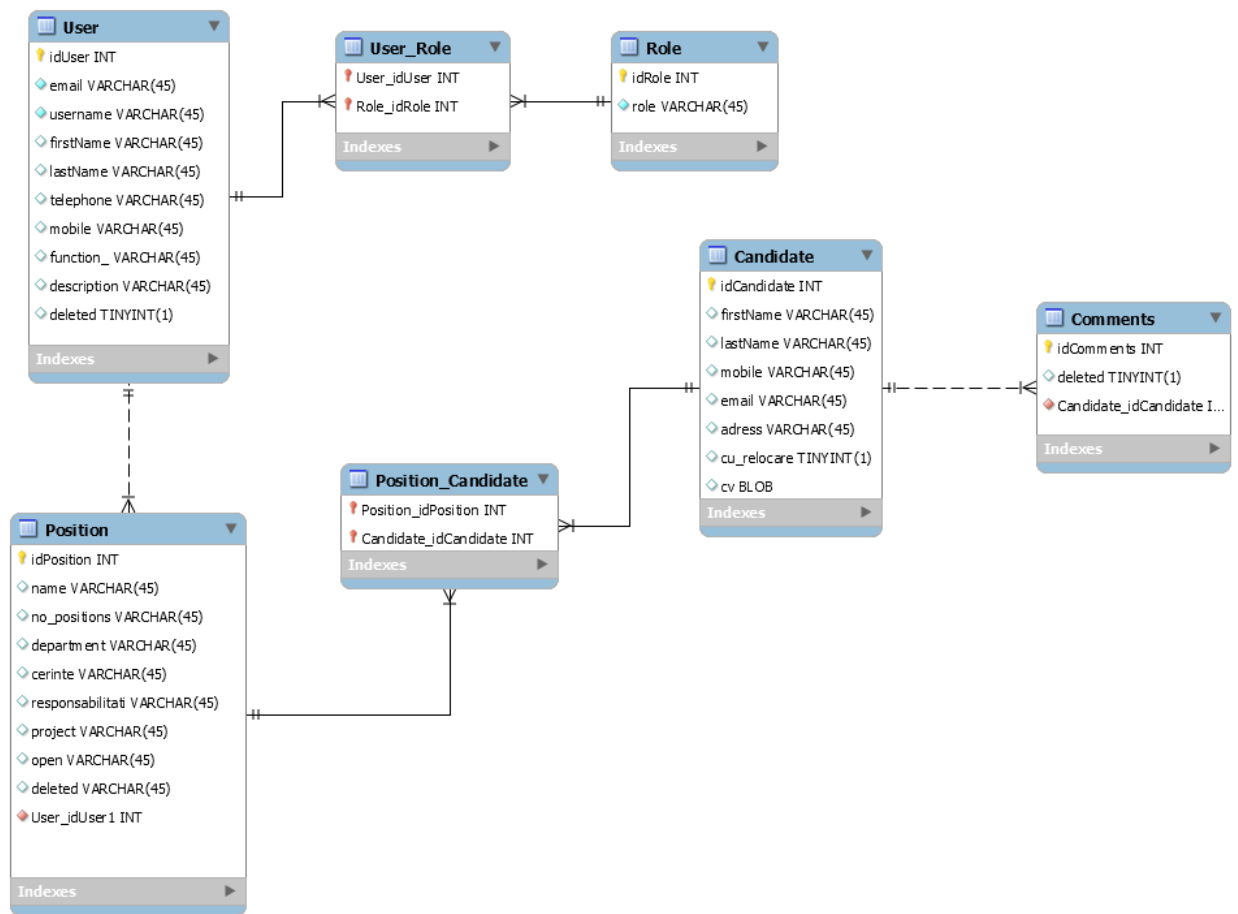
Position Comment:

Comentarii adaugate de candidat, cat si de director sau de recruiter in legatura cu pozitia.

Candidate:

Un candidat care a aplicat la o pozitie. Aici sunt salvate datele de contact si CV-ul.

Entitatile sunt mapate la tabelele din baza de date. Schema bazei de date este urmatoarea:



ii. DAO

In cadrul interfetei DAO, exista functii pentru operatiile CRUD cu baza de date. Implementariile (PositionDao, CandidateDao, UsersDao) se folosesc de EntityManager pentru a executa aceste operatii.

Un exemplu de implementare poate fi vazut aici:

```

@Stateless
public class PositionDao extends Dao<Position> {

    @PersistenceContext(unitName = "HR-ejbPU")
    private EntityManager entityManager;

    @Override
    public void add(Position position) {
        entityManager.persist(position);
    }

    @Override
    public void delete(Integer id) {
        entityManager.remove(getItemById(id));
    }

    @Override
    public void update(Position position) {
        entityManager.merge(position);
    }

    @Override
    public Position getItemById(Integer id) {
        return entityManager.find(Position.class, id);
    }

    @Override
    public List<Position> getAll() {
        return entityManager.createNamedQuery("Position.findAll").getResultList();
    }

    public List<PositionComment> getAllComments(Integer positionId) {
        return entityManager.createNamedQuery("PositionComment.findByPositionId").setParameter("positionId", positionId).getResultList();
    }

    public void addComment(PositionComment comment) {
        entityManager.persist(comment);
    }
}

```

iii. Servlets

Un servlet face maparea între un URL și o pagină jsp. În cadrul proiectului există servlet pentru pagina de login (UsersServlet) și de logout (LogoutServlet), pentru vizualizarea și adăugarea/editarea unei poziții (PositionListServlet, EditPositionServlet), pentru vizualizarea și adăugarea de comentarii și pentru redirectare în cazul unor erori de login sau de acces restricționat (ErrorServlet).

Un exemplu:

```

@WebServlet("/position-comments")
public class PositionCommentsListServlet extends HttpServlet {

    @EJB
    private PositionDao positionDao;

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        Position position = getCurrentPosition(request);
        String newComment = request.getParameter("newcomment");
        PositionComment positionComment = new PositionComment(newComment, position.getIdposition());
        positionDao.addComment(positionComment);

        response.sendRedirect(request.getRequestURI() + "?position=" + position.getIdposition());
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        Position position = getCurrentPosition(request);
        List<PositionComment> comments = getCurrentComments(request);
        request.setAttribute("position", position);
        request.setAttribute("comments", comments);
        getServletContext().getRequestDispatcher("/WEB-INF/pages/position-comments.jsp").forward(request, response);
    }

    private Position getCurrentPosition(HttpServletRequest request) {
        String positionId = request.getParameter("position");
        int id = Integer.parseInt(positionId);
        return positionDao.getItemById(id);
    }

    private List<PositionComment> getCurrentComments(HttpServletRequest request) {
        String positionId = request.getParameter("position");
        int id = Integer.parseInt(positionId);
        return positionDao.getAllComments(id);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

iv. *Pagini*

In cadrul proiectului, pentru view-uri am folosit jsp si jstl. Am folosit un template, pe baza caruia generam paginiile.

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix ="c" %>
<form action="add-edit-position" method="POST" id="add-edit-position">

    <label for="name">Name:</label>
    <input type="text" name="name" value="${position.name}"/>

    <label for="position_description">Responsibilities:</label>
    <textarea form ="add-edit-position" name="position_description" rows="20" cols="35">${position.responsabilitati}</textarea>


    <label for="no_positions">Number of positions:</label>
    <input type="text" name="no_positions" value="${position.noPositions}"/>

    <c:set var="opened" scope="session" value="${position.opened}"/>

    <label for="available">Availability:</label>
    <input type="radio" name="available" value="opened" <c:if test="${opened == 1}"> checked="true" </c:if>/>Open
    <input type="radio" name="available" value="closed" <c:if test="${opened == 0}"> checked="true" </c:if>/>Close

    <input type="hidden" name="id" value="${id}"/>
    <input type="submit" value="Save"/>
</form>

```



Name:

Responsibilities:

Number of positions:

Availability: ☒ Open ☐ Close