

Hiting App

-POO-

Interfata grafica contine:

- ***pagina de user:*** in subpagina home apar detaliile user-ului, in cea de search putem cauta un user dupa nume si in cea de notificari se updateaza notificarile automat.
- ***pagina de manager:*** in subpagina requests putem accepta / refuza cererile de angajare, in subpagina search putem cauta selecta departamentul si employee-ul si ne apar detaliile acestuia.
- ***pagina de admin:*** in subpagina home putem afisa informatiile pentru fiecare employee din fiecare companie, sa afisam job-urile, si avem un camp unde ne apare bugetul total de salarii(3 Jcombobox-uri), in subpagina search user avem un JComboBox de unde putem selecta toti userii si ne apar informatii despre acestia.
- ***pagina de employee:*** are niste mesaje printate si un buton de logout.

////Conectare///// - panou de login:

- ***pentru user*** trebuie numele user-ului, cu tot cu spatiu si parola user.

Exemplu: username: Daniel Edmund(pot fi si litere mici si mari), ***password:user***.

Exemplu: ***username:Tamara Hacı***(pot fi si litere mici si mari), ***password:user***.

Trebuie apasat butonul de login.

- **pentru manager** trebuie numele manager-ului, cu tot cu spatiu si parola manager.

Exemplu: username:Jeff Bezos(pot fi si litere mici si mari), **password:manager**.

Trebuie apasat butonul de login.

- **pentru employee** trebuie numele employee-ului cu tot cu spatiu si parola employee.

Exemplu: username:Harmony Lorinda (pot fi si litere mici si mari),
password:employee

Trebuie apasat butonul de login.

- **pentru admin: username:admin si password:admin**

Trebuie apasat butonul de Admin.

Resure:

Design pattern-uri: geeks, tutorialspoint, youtube, etc.

Interfata grafica: Youtube.

Parsare json cu Gson: <https://futurestud.io/tutorials/gson-getting-started-with-java-json-serialization-deserialization>.

Altele.

Functii mai importante:

Functia public int getDegreeInFriendship(Consumer consumer):

Determinarea gradului de prietenie cu un alt utilizator – se realizează o parcurgere în lățime în rețeaua socială a utilizatorului;

Am folosit o coada si 2 hashmap-uri: unul pentru a marca consumerii deja vizitati si celalalt pentru a calcula nivelul unui consumer in functie de nivelul parintelui dupa formula $nivel_consumer = nivel_parinte + 1$.

public int evaluate(Job job, User user):

Am creat un request, pe care l-am adaugat in lista de request-uri a managerului.

M-am folosit de numele companiei si de faptul ca aplicatia e singleton pentru a gasi managerul.

public void process(Job job):

Construim un vector de request-uri ce contine doar request-urile pentru jobul job(dat ca parametru) care va fi sortat crescator dupa scor. Concomitent vom sterge request-urile salvate din vectorul de request-uri ale managerului.

Verificam daca userul mai exista in lista de useri ai aplicatiei.

Vom crea employee-ul si ii vom completa campurile salary si company.

Cautam compania dupa nume in lista de companii si adaugam employee-ul in departamentul corespunzator job-ului.

Daca toate pozitiile au fost ocupate trimitem o notificare observatorilor.

public Recruiter getRecruiter(User user):

Cautam un recruiter potrivit pentru user-ul nostru, pe cel mai indepartat de acesta si care are rating-ul cel mai mare(dintre cei aflati la o distanta egala de user).

Pur si simplu luam pe rand fiecare recruiter din companie si il cautam pe cel mai indepartat de user. In cazul in care comparam 2 recruiteri si au level-ul fata de user egal, ii vom compara dupa rating.

private ArrayList<Integer> get_difference_day_month_year(String date1, String date2):

Datele le-am considerat string-uri. Functia returneaza un ArrayList cu diferenta din cele 2 date(reprezentate ca string-uri): pe pozitia 0 o sa avem zilele, pe pozitia 1 lunile si pe ultima pozitie anii.

private ArrayList<Integer> get_date(String date):

Functia get_date primeste un string si intoare un vector care are: pe pozitia 0 zilele, pe pozitia 1 lunile si pe ultima pozitie anii.

Public Double getTotalScore():

Am iterat prin lista de experiente si m-am folosit de functiile prezentate mai sus pentru a calcula numarul de ani de experienta. In cazul in care end_date-ul experientei era null, am facut diferenta dintre start_date si data curenta.

La final am transformati lunile in ani si apoi am aproximat prin adaos.