

CODING JOB ASSISTANT – PROJECT PROPOSAL

Nataša Ivanović SW47/2017, Mario Kujundžić SW59/2017

MOTIVACIJA

Traženje posla i priprema za intervju je najčešće iscrpljujuć proces. Neretko se dešava da studenti nakon završenog fakulteta nemaju predstavu za koju programersku poziciju da konkurišu, jer su tokom studija stekli širok spektar znanja. Takođe, ljudi koji su samouki često ne znaju kako da se usavršavaju za željenu poziciju, odn. ne umeju da u moru informacija sa interneta pronađu one koje bi im bile od najveće koristi.

PREGLED PROBLEMA

Iako postoje projekti koji se bave problemom predlaganja poslovne pozicije, nismo uspjeli da pronađemo zvanične projekte koji su u javnoj upotrebi – uglavnom su dostupni sajtovi koji ne vrše personalizovano predlaganje pozicije, već omogućuju filtriranje poslovnih oglasa na osnovu nekoliko parametara. Mi bismo ovaj koncept unapredili tako što bi sistem korisniku predlagao samo one pozicije za koje ispunjava kriterijume, kao i korake koje bi mogao da preduzme da bi imao što bolje šanse da dobije posao.

Coding job assistant aplikacija je namenjena da tražiocima programerskog posla pomogne tako što bi na osnovu podataka iz njihovog CV-ja, interesovanja i očekivanja od radnog mesta generisala predloge odgovarajućih programerskih pozicija. Na osnovu unetih podataka generisale bi se sugestije za unapređenje znanja koje je značajno za poziciju na koju tražilac posla želi da konkuriše. Takođe bi se na osnovu baze podataka oglasa i interesovanja korisnika predlagali adekvatni oglasi za posao.

PREGLED FUNKCIONALNOSTI

- **Administrator:**
 - Prijava na sistem
 - Manipulacija poslovnim pozicijama
 - Manipulacija kompanijama
 - Manipulacija oglasima
 - Manipulacija materijalima za učenje
 - Teme za projekte
 - Skup intervju pitanja
 - Materijali za dopunu znanja
 - Materijali za unapređenje stranih jezika
 - Manipulacija sistemskim entitetima:
 - Programski jezici
 - Tehnologije
 - Jezici
 - Soft skills
 - Pregled kandidata prijavljenih za oglas
 - Pregled kandidata koji prate oglas
 - Pregled kandidata koji prate poziciju
 - Pregled kandidata koji su dobili posao
 - Pregled i odobravanje review-ova
- **Job Seeker:**
 - Registracija na sistem
 - Prijava na sistem
 - Popunjavanje CV-ja
 - Zahtevanje predloga poslovne pozicije
 - Zahtevanje predloga oglasa na osnovu poslovne pozicije
 - Zahtevanje materijala za pripremu kandidata za konkretan oglas
 - Praćenje poslovne pozicije
 - Praćenje oglasa
 - Prijava na oglas
 - Postavljanje review-a
 - Pregled rang liste prijavljenih kandidata za konkretan oglas
 - Pregled rang liste kandidata koji prate poziciju
 - Pregled kompanija
 - Pregled oglasa
 - Pregled svih materijala koje je dobio u cilju unapređenja znanja (*classroom*)

METODOLOGIJA RADA

ULAZI U SISTEM

Deo sistema za predlaganje poslovne pozicije

- Elementi CV-ja (deo *JobSeeker*-a):
 - **EducationLevel:**
 - High School, Bachelors, Masters, Doctors, Self Taught
 - **List<ProgrammingLanguage>:**
 - Lista parova (*programski jezik, stepen poznavanja programskog jezika*)
 - Programski jezici – Java, JavaScript, C#, C++, Python...
 - Stepen poznavanja programskog jezika – *BASIC, GOOD, VERY GOOD, EXCELLENT, EXPERT*
 - Administrator dinamički proširuje listu
 - **List<Technology>:**
 - Lista parova (*tehnologija, stepen poznavanja tehnologije*)
 - Tehnologije – Spring, SpringBoot, Laravel, Angular, Scikit, Numpy...
 - Stepen poznavanja tehnologije – *BASIC, GOOD, VERY GOOD, EXCELLENT, EXPERT*
 - Administrator dinamički proširuje listu frameworka i biblioteka
 - **List<Knowledge>:**
 - Lista parova (*teorijski koncept, stepen poznavanja teorijskog koncepta*)
 - Teorijski koncepti – Baze podataka, REST, neuronske mreže, strukture podataka, sorting algoritmi, računarske mreže...
 - Stepen poznavanja teorijskog koncepta – *BASIC, GOOD, VERY GOOD, EXCELLENT, EXPERT*
 - Administrator dinamički proširuje listu teorijskih koncepata
 - **List<WorkingExperience>:**
 - Lista trojki (*pozicija, nivo senioriteta, dužina radnog odnosa*)
 - Pozicije – **Web Developer (Backend, Frontend, Full-Stack), Mobile developer, Game developer, Test developer, Data scientist, Project manager, Delivery manager, System administrator, DevOps**
 - Nivo senioriteta – *JUNIOR, MEDIOR, SENIOR*
 - Dužina radnog odnosa – broj meseci na radnom mestu
 - Administrator dinamički proširuje listu pozicija
 - **List<Language>:**
 - Lista parova (*strani jezik, stepen poznavanja jezika*)
 - Strani jezici – Engleski, Nemački, Francuski...
 - Stepen poznavanja jezika – *A1, A2, B1, B2, C1, C2*
 - Administrator dinamički proširuje listu stranih jezika
 - **List<SoftSkill>:**
 - Lista parova (*soft skill, stepen razvijenosti*)
 - Soft skills – **Communication, Self-motivation, Leadership, Responsibility, Teamwork, Problem solving, Decisiveness, Working under pressure, Time management, Flexibility, Negotiation**
 - Stepen razvijenosti – *BASIC, GOOD, VERY GOOD, EXCELLENT, EXPERT*

Deo sistema za predlaganje konkretnog oglasa

- Elementi CV-ja
- *JobPosition* (na osnovu predloga sistema ili samostalno odabrana iz liste)
 - *title* (glavni naziv pozicije), *subtitle* (detaljniji opis koji uključuje glavna znanja koja se koriste na poziciji)
 - *progLanguageImportance* – List<*ProgrammingLanguageImportance*>
 - Lista objekata koji sadrže polja *programmingLanguage* (referenca na objekat iz baze znanja), *importanceLevel* (procenat važnosti znanja za poziciju), *optional* (da li je znanje neophodno ili poželjno) i *proficiency* (stepen poznavanja programskog jezika)
 - Stepen poznavanja programskog jezika – *BASIC*, *GOOD*, *VERY GOOD*, *EXCELLENT*, *EXPERT*
 - Administrator dodaje kriterijume važnosti prilikom dodavanja i ažuriranja pozicije
 - *technologyImportance* – List<*TechnologyImportance*>
 - Lista objekata koji sadrže polja *technology* (referenca na objekat iz baze znanja), *importanceLevel* (procenat važnosti znanja za poziciju), *optional* (da li je znanje neophodno ili poželjno) i *proficiency* (stepen poznavanja tehnologije)
 - Stepen poznavanja tehnologije – *BASIC*, *GOOD*, *VERY GOOD*, *EXCELLENT*, *EXPERT*
 - Administrator dodaje kriterijume važnosti prilikom dodavanja i ažuriranja pozicije
 - *knowledgeImportance* – List<*KnowledgeImportance*>
 - Lista objekata koji sadrže polja *knowledge* (referenca na objekat iz baze znanja), *importanceLevel* (procenat važnosti znanja za poziciju), *optional* (da li je znanje neophodno ili poželjno) i *proficiency* (stepen poznavanja znanja)
 - Stepen poznavanja znanja – *BASIC*, *GOOD*, *VERY GOOD*, *EXCELLENT*, *EXPERT*
 - Administrator dodaje kriterijume važnosti prilikom dodavanja i ažuriranja pozicije
- *salaryExpectationsMinimum* (donja granica očekivane plate u evrima)
- *salaryExpectationsMaximum* (gornja granica očekivane plate u evrima)
- *remoteWork* (da li korisnik želi da radi od kuće)

Deo sistema za pisanje ocene kompanije

- *Review*:
 - *jobOffer* (referenca na objekat u bazi znanja)
 - *company* (referenca na objekat u bazi znanja)
 - *rating* (1-5)
 - *details* (opšti utisak o kompaniji)
 - *interviewComplexity* (1-5)
 - *jobOffered* (da li je korisniku ponuđen posao)
 - *hired* (da li je korisnik zaposlen)
 - *recommends* (da li korisnik preporučuje kompaniju)
 - *salary*
 - *ratedUseful* (broj korisnika koji ocenu smatraju korisnom)
 - *ratedNotUseful* (broj korisnika koji ocenu ne smatraju korisnom)

Deo sistema za predlaganje materijala za učenje

- Elementi CV-ja
- *JobOfferStatistic*:
 - *jobSeeker* (referenca na korisnika), *jobOffer* (referenca na oglas)
 - *progLangDifference*, *technologyDifference*, *knowledgeDifference*, *softSkillsDifference*, *languageDifference*
 - liste popunjene *ProficiencyDifference* objektima – *cvElement* (ime elementa), *jobSeekerKnowledge* (znanje korisnika), *requiredKnowledge* (potrebno znanje za poziciju) i *knowledgeType* (tip znanja)
 - tip znanja – *PROGRAMMING_LANGUAGE*, *TECHNOLOGY*, *KNOWLEDGE*, *SOFT_SKILL*, *LANGUAGE*

IZLAZI IZ SISTEMA

Deo sistema za predlaganje poslovne pozicije

- List<JobPositionRating>:
 - *rating* (procenat poklapanja korisnikovih kvalifikacija sa zahtevima pozicije)
 - *seniority* (nivo senioriteta – JUNIOR, MEDIOR, SENIOR)
 - *position* (referenca na objekat u bazi znanja)
- Ranking na listi kandidata koji prate poziciju

Deo sistema za predlaganje konkretnog oglasa

- List<JobOfferRating>:
 - Offer – objekat tipa JobOffer:
 - *company* – Administrator dinamički proširuje listu kompanija
 - *progLangImportance, knowledgeImportance, technologyImportance* – slično kao prethodno opisane liste *ProgrammingLanguageImportance, KnowledgeImportance i TechnologyImportance* objekata
 - *softSkillImportance* – List<SoftSkillImportance>
 - Lista objekata koji sadrže polja *language* (referenca na objekat iz baze znanja), *importanceLevel* (procenat važnosti znanja za poziciju), *optional* (da li je znanje neophodno ili poželjno) i *proficiency* (stepen poznavanja znanja)
 - Stepen poznavanja jezika – BASIC, GOOD, VERY GOOD, EXCELLENT, EXPERT
 - Administrator dodaje kriterijume važnosti prilikom dodavanja i ažuriranja oglasa
 - *languageImportance* – List<LanguageImportance>
 - Lista objekata koji sadrže polja *language* (referenca na objekat iz baze znanja), *importanceLevel* (procenat važnosti znanja za poziciju), *optional* (da li je znanje neophodno ili poželjno) i *proficiency* (stepen poznavanja znanja)
 - Stepen poznavanja jezika – A1, A2, B1, B2, C1, C2
 - Administrator dodaje kriterijume važnosti prilikom dodavanja i ažuriranja oglasa
 - *position* (referenca na objekat u bazi znanja)
 - *offerRank* (koliko je oglas aktuelan – NONE, BRONZE, SILVER, GOLD)
 - *ranking* (trenutna pozicija od prijavljenih kandidata)
 - *category* – JobOfferCategory (HOT_COMPANY, BEST_MATCH, LOW_COMPETITION)
 - korisnik dobija po jednu listu sugestija za sve tri kategorije (detaljnije objašnjene kasnije)
- *JobOfferStatistic* – statistika poklapanja korisnikovih kvalifikacija sa zahtevima oglasa

Deo sistema za predlaganje materijala za učenje

- Teme za projekte
- Skup intervju pitanja
- Materijali za dopunu znanja
- Materijali za unapređenje stranih jezika

BAZA ZNANJA

Projekat bi koristio relacionu bazu podataka koja bi sadržala podatke o korisnicima, podatke o pozicijama, kao i podatke koji bi mogli biti od značaja za pripremu pred intervju. Korisnik bi unosio podatke za svoj radni profil (kao što je prethodno navedeno), dok bi administrator vršio ažuriranje ostalih podataka u sistemu (poslovne pozicije i intervju materijali). Administrator bi takođe unosio informacije o oglasima za posao.

PRAVILA ODLUČIVANJA

Pravila za predlaganje poslovne pozicije

1. Filtriranje poslovnih pozicija

- Ulaz: *JobPositionSuggestion* (na početku sadrži praznu listu pozicija koja se puni okidanjem prioriternih pravila), *JobSeeker* (značajne liste *TechnologyProficiency*, *KnowledgeProficiency*)
- Izlaz: *JobPositionRating* (popunjena polja – *jobTitle*, *jobSeeker*)
- Na osnovu poznavanja tehnologija i teorijskih koncepata, okinuće se sva pravila za poslovne pozicije gde važi da korisnik ispunjava minimalne uslove (*BASIC* poznavanje krucijalnih znanja koje bi admin definisao) – pravila će se okidati dok lista pozicija ne sadrži barem 10 elemenata

2. Predlaganje poslovne pozicije uz konkretan programski jezik*

- Ulaz: *JobPositionRating* (značajna polja – *jobTitle*, *jobSeeker* – značajne liste *ProgrammingProficiency*, *TechnologyProficiency*)
- Izlaz: *JobPositionRating* (popunjeno polja *jobPosition*, *jobSubtitle*, *rating*)
- Na osnovu postojećih objekata *JobPositionRating*, za svaku poziciju (isfiltriranu iz prethodne grupe pravila) bi se okinulo pravilo koje ima najveći prioritet – računa se na osnovu poklapanja korisnikovog poznavanja programskih jezika i tehnologija, sa zahtevima poslovnih pozicija

3. Postavljanje senioriteta poslovne pozicije

- Ulaz: *JobPositionRating* (značajna polja – *jobPosition*, *jobSeeker* – značajne liste *WorkingExperience*, *TechnologyProficiency*, *LanguageProficiency*)
- Izlaz: *JobPositionRating* (popunjeno polje *seniority*)

Finalan izlaz iz ovog dela sistema je lista *JobPositionRating* objekata koji sadrže sledeća polja: *rating* (procenat poklapanja sa kvalifikacijama korisnika), *jobTitle* (pozicija), *jobSubtitle* (konkretni programski jezici i tehnologije za tu poziciju), *seniority*, *jobSeeker* (referenca na korisnika), *jobPosition* (referenca na objekat iz baze znanja).

* Pravila su međusobno isključiva (iste pozicije zahtevaju različite programske jezike i tehnologije) – pravila će se rangirati na osnovu stepena poznavanja jezika i tehnologija koje pozicija zahteva.

Primer izvršavanja pravila

* Prikazani su samo oni objekti/delovi objekata koji su relevantni za predstavljanje ovih pravila.

JobSeeker:

```
technologies: [(Spring, GOOD), (Django, BASIC), (Vue, BASIC)],
programmingLanguages: [(Java, VERY_GOOD), (JavaScript, GOOD), (Python, GOOD)]
knowledge: [(Algorithms, GOOD), (REST, VERY_GOOD), (Databases, GOOD)],
workingExperience: [(BackendDeveloper/SpringBoot, MEDIOR, 6), (FrontendDeveloper, JUNIOR, 3)]
```

JobPosition objekti u radnoj memoriji:

1. BackendDeveloper, SpringBoot, [(Spring, BASIC)], [(Java, BASIC)], [(REST, BASIC), (Databases, BASIC)]
2. BackendDeveloper, Django, [(Django, BASIC)], [(Python, BASIC)], [(REST, BASIC), (Databases, BASIC)]
3. BackendDeveloper, NodeJs, [(NodeJS, BASIC)], [(JavaScript, BASIC)], [(REST, BASIC), (Databases, BASIC)]
4. FrontendDeveloper, Vue, [(Vue, BASIC)], [(JavaScript, BASIC)], [(REST, BASIC)]
5. FullStackDeveloper, SpringBoot/React, [(Spring, BASIC), (React, BASIC)], [(Java, BASIC), (JavaScript, BASIC)], [(REST, BASIC), (Databases, BASIC)]
6. DataScientist, Python, [(Numpy, BASIC)], [(Python, BASIC)], [(NeuralNetwork, BASIC)]

Nakon izvršavanja pravila **(1)** filtriraju se sledeće pozicije: 1, 2, 4

Nakon izvršavanja pravila **(2)** filtriraju se sledeće pozicije: 1, 2, 4

Nakon izvršavanja pravila **(3)** postavlja se senioritet pozicije na osnovu JobSeeker-ovih working experience i predlaže se lista pozicija: (1, MEDIOR) -> (4, JUNIOR) -> (2, JUNIOR)

Pravila za predlaganje oglasa

1. Filtriranje oglasa koji odgovaraju kvalifikacijama korisnika

- Ulaz: *List<JobPositionRating>*, *JobOfferSuggestion* – sadrži trenutno praznu listu *JobOfferRating-a* koja će se popuniti na osnovu pravila sa najvećim prioritetom
- Izlaz: *List<JobOfferRating>* (popunjeno polje *jobOffer*)
- Na osnovu rezultata prethodne faze za svaki *jobPositionRating* objekat posmatraju se svi oglasi koji imaju referencu na isti *jobPosition* – lista *jobOfferRating-a* se popunjava oglasima koji odgovaraju prethodno predloženim pozicijama

2. Rangiranje oglasa na osnovu kvalifikacija korisnika

- Ulaz: *List<JobOfferRating>*
- Izlaz: *List<JobOfferRating>* – popunjeno polje *rating*
- Proverava se da li kvalifikacije korisnika (tehnologije, programski jezici, strani jezici, soft skills) odgovaraju zahtevima oglasa – za svaki ispunjen zahtev oglas dobija pozitivne bodove srazmerne stepenu znanja, a ukoliko korisnik ne poseduje neku od zahtevanih kompetencija oglas dobija negativne bodove srazmerne očekivanom stepenu znanja – odnosi se na obavezne kvalifikacije u oglasu.

3. Klasifikacija oglasa u kategorije*

- Oglasi koji potencijalno odgovaraju korisniku, klasifikuju se u tri grupe:
 - *BEST_MATCH*: Oglasi koji najviše odgovaraju kvalifikacijama korisnika – prosečno znanje korisnika je iznad osnovnih zahteva oglasa – odnosi se na opcione kvalifikacije u oglasu
 - *HOT_COMPANY*: Oglasi koji pripadaju popularnim kompanijama – broj pratilaca je veći od broja koji je admin predefinisao
 - *LOW_COMPETITION*: Oglasi koji nemaju mnogo pratilaca – implicira se da ima manje zainteresovanih, pa su šanse za dobijanje posla veće
- Ulaz: *List<JobOfferRating>*
- Izlaz: *List<JobOfferRating>* – popunjeno polje *category* (sa mogućim vrednostima *BEST_MATCH*, *HOT_COMPANY*, *LOW_COMPETITION*); ažurirano polje *rating* na osnovu *category*:
 - *BEST_MATCH* – dodaje se broj bodova koji je srazmeran razlici stepena znanja u odnosu na zahteve oglasa
 - *HOT_COMPANY* – dodaje se broj bodova na osnovu bedža kompanije
 - *LOW_COMPETITION* – dodaje se broj bodova na osnovu *offerRank* (sa mogućim vrednostima *NONE*, *BRONZE*, *SILVER*, *GOLD*) – što je oglas manje popularan, dodaje se više bodova

Finalan izlaz iz ovog dela sistema je lista *JobOfferRating* objekata koji sadrže sledeća polja: *rating* (procenat poklapanja sa kvalifikacijama korisnika u kombinaciji sa *category* bodovima), *category*, *jobSeeker* (referenca na korisnika), *jobOffer* (referenca na objekat iz baze znanja).

* Pravila su međusobno isključiva (isti oglasi mogu spadati pod iste kategorije) – pravila će se rangirati na sledeći način: Prvo će se proveravati skup pravila za *LOW_COMPETITION* (pretpostavlja se da će biti manji broj nepopularnih oglasa u odnosu na broj popularnih oglasa), zatim za *HOT_COMPANY* (broj hot kompanija neće biti jako velik) i onda za *BEST_MATCH* (ovo pravilo će rangirati preostale oglase).

Primer izvršavanja pravila

* Prikazani su samo oni objekti/delovi objekata koji su relevantni za predstavljanje ovih pravila.

Izlaz iz prethodnog primera izvršavanja pravila je ulaz u ovo.

JobOffer objekti u radnoj memoriji:

1. BackendDeveloper/SpringBoot, Company1, [(Spring, GOOD)], [(Java, VERY_GOOD)], [(REST, VERY_GOOD), (Databases, BASIC)], offerRank: SILVER
2. BackendDeveloper/Django, Company2, [(Django, GOOD)], [(Python, VERY_GOOD)], [(REST, VERY_GOOD), (Databases, BASIC)], offerRank: NONE

Nakon izvršavanja pravila **(1)** filtriraju se sledeći oglasi: 1 i 2 (trenutno baza znanja sadrži samo ono što mu odgovara)
Nakon izvršavanja pravila **(2)** rangiraju se oglasi: (1, 50 bodova), (2, 20 bodova – izgubio bodove jer mu fale kvalifikacije za Django)
Nakon izvršavanja pravila **(3)** kategorizuju se oglasi: (1, BEST_MATCH), (2, LOW_COMPETITION)

Pravila za predlaganje materijala za pripremu pred intervju

Generisanje statistike za korisnika

- Ulaz: *JobOfferStatistic* – sadrži referencu na *JobSeeker-a*, *JobOffer*, kao i inicijalno prazne liste *softSkillsDifference*, *technologyDifference*, *programmingLanguageDifference*, *languageDifference*
- Izlaz: *JobOfferStatistic* – popunjene liste *ProficiencyDifference* objektima (*cvElement*, *jobSeekerKnowledge*, *requiredKnowledge*, *knowledgeType*)

Pravilo će se okinuti kada korisnik klikne na oglas za koji je zainteresovan. Tada će se vršiti poređenje svih njegovih kvalifikacija (elementi CV-ja imaju *proficiency*) sa svakim zahtevom oglasa. Na ovaj način korisnik će dobiti jasan pregled u koliko meri stepen njegovog znanja odstupa od onoga što poslodavac očekuje.

Primer izvršavanja pravila

* Prikazani su samo oni objekti/delovi objekata koji su relevantni za predstavljanje ovih pravila.

JobOfferStatistic – sadrži reference na *JobSeeker-a* i oglas 2.

Nakon izvršavanja pravila popunjavaju se liste u *JobOfferStatistic* na sledeći način:

technologyDifference: [(Django, BASIC, GOOD, technology)]

progLangDifference: [(Python, GOOD, VERY_GOOD, programmingLanguage)]

knowledge: [(REST, VERY_GOOD, VERY_GOOD, knowledge), (Databases, GOOD, BASIC, knowledge)]

Generisanje saveta na osnovu korisnikovog odabira

- Ulaz: *ProficiencyDifference* – sadrži *cvElement*, *jobSeekerKnowledge*, *requiredKnowledge*, *knowledgeType*
- Izlaz: *List<InterviewSuggestion>* – objekti sadrže resurse za učenje (linkove)

Pravilo će se okinuti kada korisnik izabere koji konkretno *element CV-ja* želi da unapredi. Tada će se vršiti selekcija onih materijala gde je stepen znanja \geq stepenu znanja korisnika, a \leq zahtevu poslodavca.

Generisanje saveta na osnovu zahteva oglasa

- Ulaz: *JobOfferStatistic* – sadrži referencu na *JobSeeker-a*, *JobOffer*, kao i pune liste *softSkillsDifference*, *technologyDifference*, *programmingLanguageDifference*, *languageDifference*
- Izlaz: *LearningSchedule* – sadrži TODO spisak sa svim elementima relevantnim za oglas, ali tako da su sortirani na osnovu stepena važnosti tog elementa za poslodavca i stepena neznanja (tj. razlike između trenutnog i zahtevanog znanja)

Pravilo će se okinuti kada korisnik odabere opciju za generisanja plana učenja za konkretan oglas. Nakon generisanja, korisnik će biti preusmeren u *classroom* gde će imati pregled plana. Tu će moći da vodi evidenciju o svom učenju – kako savlada neku oblast, moći će da obeleži to na svom planu, nakon čega će mu se automatski ažurirati CV i pozicija na svim oglasima koje prati.

Pravila za određivanje statusa kompanije na osnovu Review-ova

Korisnik može da ostavi review za bilo koju poziciju u bilo kojoj kompaniji. Administrator mora da odobri svaki review pre objavljivanja. Review-ovi su značajni jer određuju status kompanije – *NONE*, *BRONZE*, *SILVER*, *GOLD*.

Pravila za dobijanje navedenih bedževa:

- **NONE:**
 - Kompanija inicijalno ima *NONE* status
- **BRONZE** – U poslednjih 6 meseci postoji bar 5 ovakvih review-ova
 - ponuđen posao
 - korisnik preporučuje firmu
 - $ratedUseful - ratedNotUseful > 0$
- **SILVER** – zahteva *BRONZE* status i da u poslednjih 6 meseci postoji bar 15 ovakvih review-ova:
 - ponuđen posao
 - korisnik preporučuje firmu
 - $ratedUseful - ratedNotUseful > 15$
 - u bar 5 review-ova korisnik je prihvatio ponudu
 - prosečna ocena kompanije je minimalno 3.5/5
- **GOLD** – zahteva *SILVER* status i da u poslednjih 6 meseci postoji bar 30 ovakvih review-ova:
 - ponuđen posao
 - korisnik preporučuje firmu
 - $ratedUseful - ratedNotUseful > 30$
 - u bar 15 review-ova korisnik je prihvatio ponudu
 - prosečna ocena kompanije je minimalno 4/5

Pravila za određivanje statusa oglasa

Kada oglasi ispune određene kriterijume, dobijaju status koji označava koliko su aktuelni – *NONE*, *BRONZE*, *SILVER*, *GOLD*. Što je status oglasa viši, to će imati veći prioritet prilikom predlaganja oglasa. Pravila za dobijanje navedenih bedževa:

- **NONE:**
 - Oglas inicijalno ima *NONE* status
- **BRONZE:**
 - minimalno 5 korisnika prati oglas
 - kompanija ima minimalno *BRONZE* status
 - oglas je star najviše 7 dana
- **SILVER** – zahteva *BRONZE* status oglasa i sledeće kriterijume:
 - minimalno 15 korisnika prati oglas
 - kompanija ima minimalno *SILVER* status
 - oglas je star najviše 14 dana
- **GOLD** – zahteva *SILVER* status oglasa i sledeće kriterijume:
 - minimalno 30 korisnika prati oglas
 - kompanija ima *GOLD* status
 - oglas je star najviše 21 dan

Za oglase starije od 30 dana, status se automatski setuje na *NONE*. Administrator može da obnovi aktuelnost oglasa bez da mora opet da kreira oglas, ukoliko je on i dalje aktuelan nakon isteka ovog roka.

Pravila za CEP (Complex Event Processing)

Pravilo za praćenje aktivnosti korisnika u učionici - TODO

Korisnik bi u učionici evidentirao svoj napredak – kada obradi neki materijal, treba da označi da je to završeno i na taj način se kreira događaj *StudiedTodayEvent* (polja *jobSeeker*, *date*, *daysActive*, *consecutive*, *finished*). Za svakog korisnika može da se napravi najviše jedan ovakav događaja u toku dana. Sistem bi svakog dana u ponoć obrađivao događaje od predhodnih 24h na sledeći način:

- Ukoliko postoji događaj koji je kreiran za istog korisnika u proteklih 24-48h, vrednost flega *consecutive* se postavlja na *true*
- Ukoliko je *consecutive* *true*, vrednost brojača *daysActive* se povećava za 1, u suprotnom se resetuje na 0.
- Ukoliko je brojač *daysActive* jednak 5, korisnik dobija bedž *ACTIVE* koji će da unapredi ranking korisnika na globalnim listama (jer kompanije vole ljude koji su spremni da uče)

Sistem bi vršio ažuriranje korisničkih bedževa tako što bi se proveravalo da li postoji *StudiedTodayEvent* – ukoliko ne postoji, korisnik gubi *ACTIVE* status.

Pravilo za praćenje malicioznih review-ova

Korisnik u bilo kom trenutku može da napiše review za bilo koju kompaniju. Pre objavljivanja review-a administrator proverava njegov sadržaj i donosi odluku da li je review relevantan ili ne. Svaki put kada administrator odbije review kreira se događaj *ReviewDeclinedEvent* (polja *jobSeeker*, *review*, *date*, *reason*, *completed*). Sistem bi proveravao broj ovih događaja na nivou korisnika i ukoliko bi se desilo da korisnik 3 puta bude odbijen, njemu mi bilo zabranjeno pisanje review-ova na 30 dana.

Pravilo za bezbednost sistema

Ukoliko korisnik prilikom logovanja pogreši lozinku, generiše se događaj *UnauthorizedLoginEvent* (polja *username*, *date*, *completed*) – tada se korisniku šalje mejl sa generisanim kodom koji mora da iskoristi prilikom sledećeg logovanja ili mu se daje opcija da ide na reset lozinke . Kod je neophodan sve dok se korisnik ne uloguje uspešno ili promeni lozinku.

Ideje za proširenje sistema:

1. Definisanje kriterijuma i pogodnosti aktuelnih pozicija
2. Sugestija pozicija/oglasa za koje korisniku fali samo jedna kvalifikacija
3. Proširenje CEP pravila za slanje mejlova ukoliko se prekine progress učenja
4. Proširenje učionice testovima za određenu oblast koje bi automatski unapredio (ili unazadio) korisnikov stepen znanja i njegovu poziciju na listama oglasa koje prate