**Odbrana projekta – Career Adviser System**

Prilikom logovanja na sistem, ukoliko korisnik vise od tri puta u roku od min ima neuspesno logovanje, dobija zabranu od trajanja od min. To je implementirano tako sto svaki put kada pokusa logovanje, u radnu memoriju se ubaci dogadjaj LoginEvent sa trajanjem od 1 min. Tek kada neki od tih dogadjaja istekne, odnosno kada broj padne ispod tri ima ponovo mogucnost logovanja. Fajl je **loginEvent.drl**

FUNKCIJE USERA

**Test licnosti**

Korisniku su ponudjene izjave koje treba da cekira, ukoliko se u njima pronalazi. Po zavrsetku, svi odgovori se submituju i sistem se bavi proracunom licnosti korisnika. Cilj je za svako polje, segment licnosti dodeliti jednu od dve vrednosti: MIND(Extraverted/Intraverted), IDENTITY(Assertive/Turbulent), NATURE(Thinking/Feeling), ENERGY(Realist/Visionary), TACTICS(Judging/Prospecting).

Prolazicemo kroz sve odgovore na pitanja, odnosno cekirane stavke korisnika – objekte TraitQuestionResult. Stavka ima za cilj dokazivanje jedne od moguce dve opcije za neki segment licnosti, odnosno mind, identity, nature, energy, tactics. Na primer, jedna od stavki koju cu ja cekirati glasi: “You enjoy vibrant social events with lots of people.” i to znaci da ce se inkrementirati polje Extraverted u objektu klase TraitsResult. Da nije bilo cekirano inkrementiralo bi se polje Introverted. Analogno vazi za ostale segmente licnosti ( necu objasnjavati posebno da ne gubim vreme, stavila sam ispis (“TARGET EXTRAVERTED, IS CHECKED == TRUE”) u navedeni primer kao dokaz)

Kada se prodje kroz sve odgovore, u objektu klase TraitsResult cemo imati akumulirane rezultate i uporedjivacemo moguce dve opcije za svaki segment licnosti. Na primer ako polje Extravered ima vecu vrednost nego polje Intraverted, zakljucicemo da je korisnik ekstrovertan i insertovacemo objekat klase Trait sa tom vrednoscu u radnu memoriju. Ispis koji cu ukazati kao dokaz, TARGET EXTRAVERTED, IS CHECKED == TRUE Isto vazi i za ostale segmente licnosti.

Na kraju cemo skupiti sve objekte klase Trait u listu, zarad olaksavanja dalje kalkulacije profesionalne orijentacije.

Objekat klase TraitResults je zapravo entitet I bice sacuvan u bazi podataka, kako bi se korisniku omogucilo ponovno prikazivanje rezultata testiranja.

**Test preferencija**

Svaka profesija u sistemu ima listu aktivnosti koje je opisuju.

Korisniku su ponudjene stavke koje se odnose na te aktivnosti, koje je potrebno da cekira ukoliko voli ili bi voleo da ih radi. Kako bi sva pravila u preferencesTest.drl fajlu bila okinuta, cekiracu stavke na sledeci nacin.

* Cekiracu sve stavke koje se odnose na Medicinu (5/5) kako bi se aktiviralo pravilo 0.

Aktivira se jer je to profesija kod koje su cekirane sve aktivnosti koje ona poseduje, plus je to najveci broj cekiranih svih aktivnosti.

* Cekiracu sve stavke koje se odnose na Programiranje (3/3) kako bi se aktiviralo pravilo 1.

Aktivira se jer je to profesija kod koje su cekirane sve aktivnosti koje ona poseduje, ali to nije najveci broj cekiranih svih aktivnosti neke profesije.

* Cekiracu 6/7 stavki koje se odnose na Pravo kako bi se aktiviralo pravilo 2.

Aktivira se jer je to profesija kod koje nisu cekirane sve aktivnosti koje ona poseduje, ali je cekirani broj aktivnosti veci od maksimalnog cekiranog broja aktivnosti (Medicina).

* Cekiracu 2/3 stavke koje se odnose na Psihologiju kako bi se aktiviralo pravilo 3.

Nisu cekirane sve aktivnosti koje profesija poseduje i cekirani broj aktivnosti je manji od minimalnog cekiranog broja aktivnosti (Programiranje).

* Cekiracu 4/6 stavki koje se odnose na Turizam kako bi se aktiviralo pravilo 4.

Nisu cekirane sve aktivnosti koje profesija poseduje i cekirani broj aktivnosti je veci od minimalnog cekiranog broja aktivnosti (Programiranje).

**Prikaz rezultata**

**Criteriums: Personality- fajl traitsTest.drl**

Prvo se mora pokrenuti personalityTest.drl da bi se objekti Trait ubacili u radnu memoriju.

Na osnovu rezultata testa licnosti (mind: extraverted, identity: assertive, nature: thinking, energy: realist, tactics: judging), kada odeberemo ovaj kriterijum aktivirace se sledeca pravila:

Najpre pravilo “Calculating numbe of matching traits”,koje pomocu Query-a racuna da li je neka profesija potencijalna, a na osnovu toga da li korisnik zadvoljava barem jednu zahtevanu osobinu. Zatim ce se aktivirati sledeca pravila:

* Pravilo 0 – prvo mesto jer je score 1.5

Za profesiju Pravo, jer su ispunjeni svi zahtevi za osobinama I to je profesija koja ima maksimalan broj zahtevajucih osobina koje su zadovoljene (3/3) - (mind: extraverted, nature: thinking, energy: realist).

* Pravilo 1 – score 1.0

Za profesiju Programiranje, jer su ispunjeni svi zahtevi za osobinama ali nije profesija koja ima maksimalan zahtevajuci broj osobina (2/2) – (energy: realist, nature: thinking).

Za profesiju Administracija, jer su ispunjeni svi zahtevi za osobinama ali nije profesija koja ima maksimalan zahtevajuci broj osobina (1/1) – (energy: realist).

Za profesiju Statistike, jer su ispunjeni svi zahtevi za osobinama ali nije profesija koja ima maksimalan zahtevajuci broj osobina (1/1) – (energy: realist).

* Pravilo 2 - score 1.25

Za profesiju Medicina, jer nisu zadovoljene sve osobine (zadvoljeno 4/5 - (mind: extraverted, identity: assertive, nature: thinking, energy: realist), nije zadovoljeno tactics: prospecting) ali je broj zadovoljenih osobina veci od maksimalnog broja zadovoljenih osobina – Pravo.

* Pravilo 3

Za profesija Psihologija, jer nisu zadovoljene sve osobine (zadovoljeno 1/5 – (identity: assertive), nije zadovoljeno 4/5- (mind: introverted, nature: feeling, energy: visionary, tactics: prospecting) I taj broj je manji od minimalnog broja svih zadovoljenih osobina – Programiranje, Administracija, Statistika.

Za profesija Graficki dizajn, jer nisu zadovoljene sve osobine (zadovoljeno 1/3 – (identity: assertive), nije zadovoljeno 2/3- (energy: visionary, tactics: prospecting) I taj broj je manji od minimalnog broja svih zadovoljenih osobina – Programiranje, Administracija, Statistika.

* Pravilo 4 – score 1.0

Za profesiju Turizam, jer nisu zadovoljene sve osobine (zadvoljeno 3/4 - (mind: extraverted, nature: thinking, energy: realist), nije zadovoljeno tactics: prospecting) ali je broj zadovoljenih osobina veci od minimalnog broja zadovoljenih osobina – Programiranje, Administracija, Statistika.

Rezultat na konzoli bice:

**Rule 0 : score 1.5** - \*\*All traits satisfied, but and it is the highest number of traits required.\*\* - **Profession Law**

**Rule 1 : score 1.0** - \*\*All traits satified, but it is not the highest number of traits required.\*\* - **Profession Statistics**

**Rule 1 : score 1.0** - \*\*All traits satified, but it is not the highest number of traits required.\*\* - **Profession Programming**

**Rule 1 : score 1.0** - \*\*All traits satified, but it is not the highest number of traits required.\*\* - **Profession Administration**

**Rule 2 : score 1.25** - \*\*Not all traits satisfied, but it is larger number than the highest number of maximum all satisfied.\*\* - **Profession Medicine**

**Rule 3 : score 0.2** - \*\*Not all traits satisfied, and it is smaller number than min number of all required.\*\* - **Profession Psychology**

**Rule 3 : score 0.3333333333333333** - \*\*Not all traits satisfied, and it is smaller number than min number of all required.\*\* - **Profession Graphycal design**

**Rule 4 : score 1.0** - \*\*Not all traits satisfied, but it is larger number than the minimum all required.\*\* - **Profession Travel and tourism**

Zatim se aktivira pravilo iz results.drl fajla, koje skuplja sve objekte PossibleProfession u listu, koje cemo obaciti ponovo u radnu memoriju I aktivirati pravilo iz topThree.drl fajla, koje ce vratiti top tri objekta PossbileProfession, odnosno objekte koji imaju najveci score. Morala sam ovako da odradim, jer se dinamicki salience koji se korisit u topThree.drl fajlu odredjuje pre svih pravila a on koristi vrednost score koji se naknadno izracuna I mora biti poznat znaci pre nego sto se aktivira ova grupa pravila.

**Criteriums: Preferences – fajl preferencesTest.drl**

Bice okinuta pravila naspram kojih sam cekirala aktivnosti kako bi sva bila aktivirana I na konzoli ce biti ispisano sledece:

**Rule 0 : score 1.5** - \*\*All activities selected, but and it is the highest number of activities.\*\* - **Profession Medicine**

**Rule 1 : score 1.0** - \*\*All activities selected, but it is not the highest number of activities.\*\* - Pr**ofession Programming**

**Rule 2** **: score 1.25** - \*\*Not all activities selected, but it is larger number than the highest number of maximum all selected.\*\* - **Profession Law**

**Rule 3 : score 0.6666666666666666** - \*\*Not all activities selected, and it is smaller number than min number of all selected.\*\* - **Profession Psychology**

**Rule 4 : score 1.0** - \*\*Not all activities selected, but it is larger number than the minimum of all selected.\*\* - **Profession Travel and tourism**

Takodje vidimo da je aktivirano pravilo “If there are not possible profeessions find and insert them.” koje ubacuje objekte PossibleProfession koji se koriste u prethodno navedenim pravilima. Ono radi po principu da ubacuje profesiju kao potencijalnu, ukoliko je korisnik cekirao barem jednu aktivnost koju ona poseduje.

Vracanje rezultata pomocu pravila prethodno opisanih u results.drl I topThree.drl fajlu.

**Criteriums: Payment – fajl paymentTest.drl**

Bice aktivirano pravilo “If there are not possible professions, find and insert them.”. Gde ce score potencijalne profesije biti izracunat kao: plata te profesije / najveca plata.

Vracanje rezultata pomocu pravila prethodno opisanih u results.drl I topThree.drl fajlu.

**Criteriums: Employment – fajl employmentTest.drl**

Pre izvrsavanja pravila iz employmentTest.drl fajla poziva izvravanje template-a iz fajla EmploymentScore.drt u kojem ce biti izracunat employmentScore svake profesije.

Nakon toga ce se izvrsiti pravila iz ovog fajla.

Vracanje rezultata pomocu pravila prethodno opisanih u results.drl I topThree.drl fajlu.

**Criteriums: Personality + Preferences+ Payment + Employment**

Kombinacija kriterijuma vrsi se tako sto se aktiviraju pravila iz redom ovih fajlova (broj kombinovanih kriterijuma je proizvoljan). U fajlu traitsTest.drl ce se ubaciti objekti PossibleProfession nad kojima ce se raucnat I akumulirati i score, prema pravilima za svaki kriterijum koji sam zasebno objasnila.

Vracanje rezultata pomocu pravila prethodno opisanih u results.drl I topThree.drl fajlu.

**Matching preferences and traits**

Korisnik kada klikne na prikaz detalja profesije ima mogunost prikaza osobina koje profesija zahteva, za naznakom koje poseduje a koje ne, kao i aktivnosti koje je opisuju, sa naznakom koje je selektovao a koje nije. To je opisano u matching.drl fajlu.

Posto postoje dve opcije postoje I dva pravila: *“Calculating number of matching preferences”* I *“Calculating number of matching traits.”* Sto se tice prvog pravila ubacuju se objekti **Profession** i objekti **Traits** korisnika i filtiraju se Traits objekti koji su u preseku. Po istom principu je implementirano I drugo pravilo, ubacuju se objekti **Profession** I objekti **PreferenceQuestionResult** i izdvajaju Preference objekti u preseku.

**Candidate professions**

Korisnik ima takodje mogucnost prikaza svih profesija sa kojima se poklapa makar po jednoj osobini ili makar po jednoj aktivnosti. To je implementirano u fajlu suitableProfessions.drl.

U fajlu se nalaze dva pravila pomocu kojih je ovo realizovano – *Find candidates based on traits* I *Find candidates based on preferences.* Sustina ovih pravila jeste da je profesija odbarana da se smatra kandidatom za korisnika, ukoliko zahteva barem jednu karakternu crtu koju korisnik poseduje ili aktivnost koju je cekirao da voli da obavlja. To se ispituje tako sto se ubace sve profesije u radnu memoriju, korisnikove karakterne crte - **Traits** objekti i svi korisnikovi odgovori na test preferencije – objekte **PreferenceQuestionResult** i zatim se pomocu dva query-a nalaze poklapanja.

Sada cu traziti ponovno izvrsavanje testa I selektovati samo jednu aktivnost kako bih dokazala da ce samo jedna profesija biti pronadjena kao potencijalna. Takodje cu pokazati da je samo ta aktivnost medju svim aktivnostima te profesije selektovana.

**Ponovno izvrsavanje testova**

Repeat preferences test-newPersonalityTest.drl

Svaki put kada korisnik zatrazi ponovno izvrsavanje testa, insertuje se dogadjaj NewPersonalityTestEvent, koji traje 2 min. Ukoliko se poslednjih 2 min naslo tri takva dogajaja, korisniku se zabranjuje novi test. Tek kada barem jedan istekne (broj zahteva u poslednje 2 min padne ispod tri), korisniku ce se dozvoliti obavljanje testa.

Sada cu to demonstrirati.

Repeat profession test-newProfessionTest.drl

Svaki put kada korisnik zatrazi ponovno izvrsavanje testa, insertuje se dogadjaj NewPersonalityTestEvent, koji traje 2 min. Ukoliko se poslednjih 2 min naslo tri takva dogajaja, korisniku se zabranjuje novi test. Tek kada barem jedan istekne (broj zahteva u poslednje 2 min padne ispod tri), korisniku ce se dozvoliti obavljanje testa.

Posto je implementacija identicna, necu trositi vreme na prikazivanje.

FUNKCIJE ADMIN

**Definisanje skale za racunanje employmentScore-a**

Adminu je omoguceno unosenje granica opsega prilikom racunanja employmentScore-a koje se obavlja u employmentScore.drt fajlu. Bice napravljeni novi EmploymentScoreTemplate objekti, ubaceni u radnu memoriju zajedno sa Profession objektima za koje ce se izraucnati novi employmentScore.

**Prikaz izvestaja**

Ovo je uradjeno pomocu **Complex Event Processing** –a u fajlu **report.drl.** Sustina je da bi svaki put kada bi korisnik odradio testove, bio ubacen TestDoneEvent objekat u radnu memoriju u kojoj je pomocu window pricipa belezen njihov broj u poslednjih 24h. Pomocu Scheduled anotacije se svaki dan u 8h ucitava dnevni report (testsDoneInDay.service)

**Dodavanje profesije**

**Edit profesije**

**Brisanje profesije**

**Dodavanje osobine**

**Dodavanje pitanja za test licnosti**

**Brisanje pitanja za test licnosti**

**Edit pitanja za test licnosti**

**Ponovna aktivacija obrisane profesije**

**Registracija admina**

**Registracija korisnika**