

Pametni vodič za prvo putovanje u Sloveniju

Autor: Ilija Bešlin SV71/2021

Motivacija

Planiranje putovanja, naročito u geografski i sadržajno raznoliku zemlju kao što je Slovenija, predstavlja kompleksan problem. Turisti koji dolaze prvi put često su preplavljeni izborom između Alpa, mora, gradova i pećina, i teško im je da kreiraju logistički optimizovan i personalizovan plan. Postojeći alati su uglavnom statične liste. Iz tog razloga sam odlučio da razvijem pametnog vodiča koji dinamički generiše plan na osnovu korisničkih preferenci i realnih ograničenja koja već postojeća rešenja ne uzimaju u obzir.

Opis problema

Aplikaciju mogu da koriste dve grupe korisnika: **Administrator** i **korisnik (putnik)**.

Administrator može da:

- Dodaje i ažurira podatke o lokacijama i atrakcijama (npr. Bled, Postojnska jama).
- Definiše ključne attribute lokacija: tip (jezero, pećina), cenu, radno vreme, potrebnu fizičku spremu, tagove (npr. "romantika", "avantura").
- Upravlja matricom povezanosti: unosi prosečno vreme putovanja između ključnih lokacija.

Korisnik (putnik) može da:

- Definiše svoje preference za putovanje: broj dana, budžet (nizak, srednji, visok), način transporta (automobil, javni prevoz), fizičku spremu i listu interesovanja (npr. "planinarenje", "istorija", "gastronomija"), kao i listu "must-have" lokacija koje obavezno želi da poseti.
- Pregleda listu "trending" lokacija, koje se dinamički identifikuju na osnovu popularnosti planova drugih korisnika.
- Dobije automatski generisan, detaljan i personalizovan plan putovanja po danima.
- Dobije logistička upozorenja (npr. "Plan za ovaj dan je previše ambiciozan" ili "Potrebna vam je topla odeća za posetu pećini").

Metodologija rada

Rešenje ovog problema biće implementirano pomoću sledećeg modela podataka (entiteta):

- **Location:** Centralni entitet sistema. Određen je nazivom, tipom, regionom, cenom ulaznice, procenjenim vremenom za obilazak, zahtevanom fizičkom spremom, dostupnošću javnim prevozom i listom **tagova** koji ga opisuju.
- **TravelPreferences:** Određen je brojem dana putovanja, budžetom, načinom transporta, nivoom fizičke spreme, listom interesovanja, mesecom putovanja (zbog sezonskih pravila) i listom ID-jeva "must-have" lokacija.
- **Route:** Predstavlja logističku vezu između dve lokacije. Sadrži lokacijaA, lokacijaB i vremeVoznjeMin.
- **Recommendation:** Prelazni objekat koji se kreira tokom rezonovanja. Sadrži referencu na Lokaciju, obrazloženje zašto je preporučena i dodeljene poene.
- **ItineraryItem:** Finalni izlazni objekat. Predstavlja jednu stavku u planu putovanja. Određena je danom, lokacijom i predloženim aktivnostima.
- **Alert:** Objekat koji sadrži tekstualno upozorenje za korisnika (npr. logističko, budžetsko, sezonsko).
- **RuleParameter:** Objekat kojim administrator može da podešava logiku sistema bez izmene koda (npr. maksimalno dozvoljeno vreme vožnje u danu, gornje granice za budžetske kategorije).
- **LocationVisitedEvent:** Događaj koji se kreira za svaku lokaciju u novogenerisanom planu. Ubacuje se u "stream" sesiju i služi kao ulaz za CEP pravila.
- **TrendingLocation:** Činjenica koja se kreira od strane CEP pravila kada broj poseta određenoj lokaciji pređe definisani prag u zadatom kliznom prozoru.

Opis pravila

Logika sistema je podeljena u sledeće grupe pravila:

1. Pravila osnovnog filtriranja i klasifikacije - Ova pravila vrše prvu selekciju lokacija na osnovu osnovnih preferenci korisnika.

- **Filtriranje po transportu:** Ako korisnik izabere "javni prevoz", iz daljeg razmatranja se izbacuju sve lokacije koje su označene kao teško dostupne tim vidom transporta (npr. Logarska dolina).
- **Filtriranje po fizičkoj spremi:** Iz razmatranja se izbacuju lokacije koje zahtevaju viši nivo fizičke spreme od one koju je korisnik naveo (npr. korisnik sa "niskom" spremom neće dobiti preporuku za uspon na Triglav).

- **Filtriranje po budžetu:** Lokacije čija cena ulaznice značajno premašuje budžet korisnika dobijaju negativne poene ili se izbacuju, osim ako ih je korisnik eksplicitno naveo na svojoj "must-have" listi.

2. Pravila za kreiranje personalizovanog itinerera (Višeslojno rezonovanje) Ovo je srce sistema koje koristi forward-chaining kroz više nivoa da bi se od grube liste došlo do finog plana.

✚ **Nivo 1: Generisanje početnih preporuka.** Pravilo pronalazi sve lokacije čiji se tagovi poklapaju sa interesima korisnika i za svaku od njih kreira Preporuka objekat sa početnim brojem poena.

- *Primer pravila:* Reward locations based on interests

✚ **Nivo 2: Bodovanje i nijansiranje preporuka.** Niz pravila analizira Preporuka objekte i dodaje ili oduzima poene na osnovu složenije logike izvedene iz analize (Bled vs. Bohinj, Postojna vs. Škocjan).

- *Primer pravila:* Ensure 'Must-Have' locations:
 - Pravilo sa najvišim prioritetom (salience 1000) koje pronalazi sve preporuke čiji se ID lokacije nalazi u korisnikovoj mustHaveLocationIds listi. Ovim lokacijama dodeljuje ogroman broj poena (npr. 1.000.000), čime se garantuje njihovo uključivanje u finalni plan bez obzira na ostale faktore.
- *Primeri pravila:*
 - Reward Bled for "romance" interest: Dodeljuje dodatne poene Bledu ako korisnik voli romantiku.
 - Reward Bohinj for "adventure" interest: Dodeljuje dodatne poene Bohinju ako korisnik voli avanturu.
 - Punish Skocjan for low fitness: Dodeljuje negativne poene Škocjanskim jamama ako korisnik ima nisku fizičku spremu.
 - Reward Postojna for families: Dodeljuje dodatne poene Postojnskoj jami ako korisnikov profil ukazuje na porodično putovanje.

✚ **Nivo 3: Sklapanje logičkog itinerera.** Nakon bodovanja, sistem uzima najbolje ocenjene preporuke i pokušava da ih složi u dane.

- *Primer pravila:* Collect best recommendations: Koristi accumulate da prikupi N najbolje ocenjenih preporučenih objekata.

- *Primer pravila:* Group nearby locations: Analizira najbolje preporuke i, koristeći Udaljenost činjenice, grupiše one koje su geografski blizu (npr. Bled i Vintgar) u jedan dan. Kreira ItineraryItem objekte.

✚ **Nivo 4: Finalna provera i generisanje upozorenja.** Pravila na ovom nivou analiziraju generisane ItinererStavka objekte za svaki dan.

- *Primer pravila:* Check daily schedule duration: Koristi accumulate da sabere vreme obilaska svih aktivnosti u danu i vreme vožnje između njih. Ako suma prelazi definisani limit (npr. 8 sati), kreira se upozorenje - "Plan za Dan X je previše ambiciozan".

3. Pravila zasnovana na templejtima Ova pravila omogućavaju administratoru da lako menja poslovnu logiku sistema.

- **classifyByBudget (template):** Administrator definiše gornje granice cena ulaznica za kategorije "nizak", "srednji" i "visok" budžet. Templejt generiše pravila koja klasifikuju svaku lokaciju u odgovarajuću budžetsku kategoriju. Ova pravila ignorišu budžetsko ograničenje za lokacije koje je korisnik označio kao "must-have".

4. Pravila za interaktivno razjašnjavanje (Backward Chaining)

Pored osnovne logike zasnovane na zaključivanju unapred (forward-chaining), sistem koristi zaključivanje unazad za dve odvojene, napredne funkcionalnosti:

A. Klasifikacija Tipa Putovanja

Nakon što se generiše plan putovanja, sistem koristi lanac zaključivanja unazad da bi izveo kvalitativni, deskriptivni zaključak o prirodi samog putovanja. Umesto da donese zaključak u jednom koraku, sistem postavlja finalni cilj – TripClassification – i radi unazad da bi prvo dokazao nekoliko privremenih, apstraktnih zaključaka:

- **Tematski Fokus (TripFocus):** Sistem prvo analizira sve lokacije u planu i korišćenjem accumulate funkcije određuje da li je plan dominantno **alpski, kraški, gradski** ili vezan za **istočno nasleđe**. Zaključak se donosi na osnovu procentualne zastupljenosti lokacija iz određenih regiona ili tipova.
- **Intenzitet (TripIntensity):** Na osnovu početnih preferenci korisnika i fizičke zahtevnosti lokacija u planu, sistem zaključuje da li je putovanje **opuštajuće, umereno** ili **aktivno**.
- **Tempo (TripPacing):** Na osnovu odnosa broja dana i ukupnog broja lokacija u planu, sistem zaključuje da li je tempo putovanja **opušten, balansiran** ili **ubrzan**.

Tek kada su ovi privremeni zaključci dokazani, pravila višeg nivoa ih kombinuju da bi formirala finalnu, nijansiranu klasifikaciju kao što je "Ekstremna alpska ekspedicija" ili "Opuštena gradska tura posvećena kulturi"

B. Rekurzivna Provera Dostižnosti Lokacija

Sistem implementira **rekurzivni query** pod nazivom **isReachable**. Ovaj query odgovara na pitanje: "Da li je lokacija B dostižna iz lokacije A unutar zadatog vremenskog budžeta, čak i preko više međustanica?".

- **Osnovni slučaj:** Dostižnost je dokazana ako postoji direktna Route između A i B koja je kraća od vremenskog budžeta.
- **Rekurzivni slučaj:** Dostižnost je dokazana ako postoji ruta od A do međustanice C, i ako se rekurzivnim pozivom dokaže da je B dostižno iz C sa preostalim, umanjenim vremenskim budžetom.

5. Pravila za obradu složenih događaja (CEP - Complex Event Processing)

Pored statičkog planiranja, sistem koristi i drugu, odvojenu Kie sesiju (*cepSession*) koja radi u "stream" modu za detekciju trendova.

- **Generisanje događaja:** Kada god korisnik generiše *TravelPlanResponse*, pravilo-prevodilac presreće ovaj objekat i kreira niz *LocationVisitedEvent* događaja (po jedan za svaku lokaciju u planu) koje ubacuje u radnu memoriju CEP sesije.
- **Primer pravila *Detect Trending Location*:**
 - Ovo pravilo koristi klizni prozor (***over window:length***) da analizira poslednjih N (npr. 1000) *LocationVisitedEvent* događaja.
 - Koristeći *accumulate* i *group by*, pravilo broji koliko se puta svaka lokacija pojavila u tom prozoru. Ako broj poseta za određenu lokaciju pređe definisani prag (npr. 50), pravilo kreira *TrendingLocation* činjenicu.
 - Ovu činjenicu frontend zatim preuzima i prikazuje korisnicima kao "popularnu" ili "trending" lokaciju, koju oni mogu dodati na svoju "must-have" listu.

Konkretan primer rezonovanja

Da bismo ilustrovali kako sistem funkcioniše, pratićemo jedan zahtev korisnika od početka do kraja.

1. **Korisnički unos (Input):** Korisnik unosi sledeće preference (TravelPreferences objekat):

- numberOfDays: 3
- budget: MEDIUM
- transport: CAR

- fitnessLevel: MEDIUM
- interests: ["nature", "history"]
- travelMonth: 7 (Jul)
- mustHaveLocationIds: ["bc58d7c3-3cb3-4799-8d88-b5e060ddbb2b"] - Bled

2. Inicijalizacija sesije:

- Prvo se pokreće privremena sesija za filtriranje po budžetu (filterLocationsByBudget). Pravilo iz templejta identifikuje da je "Lake Bled" (čija je cena > limita za MEDIUM budžet) kandidat za brisanje.
- Međutim, uslov nije zadovoljen (jer se ID Bleda *nalazi* u listi). Lokacija "Lake Bled" nije obrisana i prolazi u sledeću fazu.
- Zatim sistem kreira glavnu Drools sesiju (scoringSession). U nju ubacuje sve preostale Location i Route objekte, kao i TravelPreferences korisnika

3. Nivo 1: Generisanje početnih preporuka: Pravilo Reward locations based on interests se aktivira više puta.

- Za interes "nature", kreiraju se Recommendation objekti za Bled, Bohinj, Vintgar, Škocjanske jame itd. Svaki dobija početni skor od 10 poena.
- Za interes "history", kreiraju se Recommendation objekti za Blejski grad, Predjamski grad, Ljubljanski grad itd. I oni dobijaju početni skor od 10 poena.
- Radna memorija je sada popunjena listom potencijalnih preporuka.

4. Nivo 2: Bodovanje i filtriranje: Sada se aktiviraju specifičnija pravila.

- Prvo se aktivira pravilo Ensure 'Must-Have' locations. Ono pronalazi Recommendation objekat za "Lake Bled" i, pošto se ID poklapa sa mustHaveLocationIds listom, postavlja njegov skor na 1.000.000 poena.
- Pravilo koje filtrira po fizičkoj spremi se aktivira za lokaciju "Uspon na Triglav" (koja zahteva HIGH nivo) i uklanja tu preporuku, jer korisnik ima MEDIUM nivo.
- Aktivira se pravilo koje dodaje poene Blejskom gradu jer se poklapa sa interesom za istoriju i nalazi se na lokaciji Bled koja se poklapa sa interesom za prirodu, čime se njegov skor povećava.

5. Nivo 3: Sklapanje logičkog itinerera: Sistem sada radi sa visoko ocenjenim preporukama.

- Pravilo Assign BEST recommendation to Day 1 se aktivira. Zbog skora od 1.000.000, ono nepogrešivo bira "Lake Bled" kao prvu lokaciju za Dan 1.
- Zatim se aktivira pravilo Group nearby locations. Ono pronalazi da je "Vintgar Gorge" blizu Bleda i da se uklapa u preostalo vreme za Dan 1. Dodaje Vintgar u plan za Dan 1.
- Sistem nastavlja sa sledećom najbolje ocenjenom lokacijom (npr. Predjamski grad) za Dan 2, i proces se ponavlja.

6. **Nivo 4: Finalna provera i generisanje upozorenja:** Pravila na ovom nivou analiziraju kreirane ItineraryItem objekte.

- Pravilo Check daily schedule duration se izvršava za Dan 1. Sabira vreme: visit(Predjamski grad) (90 min) + visit(Škocjanske jame) (180 min) + travelTime (20 min) = 290 minuta. Pošto je $290 < 480$ (limit od 8 sati), pravilo ne generiše upozorenje.

7. **Finalni izlaz (Output):** Sistem prikuplja sve ItineraryItem i Alert objekte iz radne memorije. Korisniku se prikazuje generisani plan za 3 dana.