Koktel bar

Predlog projektnog zadatka, Sistemi bazirani na znanju

Projekat Koktel bar sastojao bi se iz dva glavna podsistema:

1. Sistem za preporuku, poručivanje koktela i praćenje inventara bara
2. Sistem za organizovanje koktel događaja i modifikaciju karte pića u realnom vremenu

# Motivacija

Za razilku od drugih pića kokteli se mogu sastojati od mnogo kompleksnih sastojaka, gde neki od njih mogu biti više ili manje poznati. Uglavnom karte piće nude samo naziv, ili u nekim slučajevima naziv I spisak sastojaka, tako da gosti koje nisu poznavaoci koktela često imaju problem da odluče koji bi im se dopao. Sa ovom aplikacijom je dovoljno znati da li preferiraju slatka ili kisela pića, da li žele jako ili slabo alkoholno piće i druge komponente ukusa i dobijaju personalizovanu kartu pića koja im odgovara. Još jedan motivator jeste to da može postojati mnoštvo kombinacija sastojaka koje bi bar mogao da nudi, ali kada bi postavili to na jednu kartu pića broj stranica bi bio prevelik.

Kada pričamo o organizovanju događaja ovo takođe može biti veoma kompleksan zadatak za nabavku pića. Da bismo izbacili potrebu za silnim proračunima ova aplikacija bi na osnovu broja ljudi i proizvoljnog dodatnog broja informacija odradila sve to umesto nas. Uz to ima mogućnost i modifikacije karte pića na osnovu preostalog inventara i predhodno poručenih koktela.

Na kraju praćenje inventara u toku večeri omogućava da se neće desiti da neko poruči koktel za koji nisu svi sastojci dostupni. Ovime osoblju olakšavamo posao, dok gostima optimizujemo ugođaj.

# Pregled problema

Kroz motivaciju opisali smo glavne probleme i načine na koje bih ih ova aplikacija rešila. U daljem tekstu izvršićemo rezime problema po tačkama.

1. Kokteli su kompleksna pića, pa većina ljudi nije upoznata sa svim sastojcima i sa ukusima kojima kombinacija sastojaka rezultuje.
2. Kalkulacije za nabavljanje sastojaka za uspešno organizovanje događaja su kompleksne.
3. Karta pića limitira na određen broj koktela, dok sastojci mogu različitim kombinacijama stvoriti mnogo više koktela.
4. Praćenje inventara iziskuje dosta vremena, automatizacija smanjuje obim posla.

## Srodna rešenja

Istraživanjem nismo došli do nijednog rešenja za preporuku koktela koje uzima u obzir preference ukusa gostiju. Uglavnom se preporučuju konačne liste najpopularnijih koktela. Što se tiče izračunavanja količine pića za događaje postoje online kalkulatori, međutim oni su fokusirani uglavnom na tri vrste pića: pivo, vino i žestoka alkoholna pića. Nismo pronašli softver koji preporučuje koktele i količinu sastojaka potrebnu za njih.

# Metodologija rada

Ovu sekciju ćemo podeliti na osnovu podsistema aplikacije.

## Sistem za preporuku, poručivanje koktela i praćenje inventara bara

U okviru ovog podsistema postoje dva toka rezonovanja.

Prvi tok rezonovanja predstavlja proces preporuke koktela. Ulaz u sistem predstavljaju svi ukusi koje je gost odabrao, recepti svih koktela koji su u sistemu, kao i trenutni inventar bara što je količina sastojaka koje bar trenutno poseduje. Nakon obrade podataka kao izlaz iz sistema dobijamo listu koktela koji odgovara kriterijumima.

Drugi tok rezonovanja nadovezuje se na prvi. Ulaz u sistem je koktel koji gost poručuje sa liste koktela I trenutni inventar bara. Na osnovu recepta tog koktela, količine sastojaka koje sadrži, ažurira se količina sastojaka u inventaru i kao izlaz dobijamo modifikovani inventar sastojaka.

## Sistem za organizovanje koktel događaja i modifikaciju karte pića u realnom vremenu

U okviru ovog podsistema takođe postoje dva toka rezonovanja.

Prvi tok rezonovanja vezan je za planiranje sastojaka za događaj. Ulaz u sistem predstavljaju podaci o gostima (broj gostiju-broj muškaraca; broj žena, tip događaja) i recepti svih koktela koje bar nudi. Kao izlaz iz sistema dobijamo količinu sastojaka koje treba nabaviti, kao i inicijalnu kartu pića podložnu izmenama.

Drugi tok rezonovanja predstavlja ažuriranje karte pića u relanom vremenu. Ulaz u sistem su recepti svih koktela, trenutni inventar bara, istorija poručenih koktela za to veče. Kao izlaz iz sistema dobijamo ažuriranu kartu pića.

## Baza znanja

Bazu znanja možemo podeliti na podatke koje čuvamo u bazi podataka trajno i podatke koji se nalaze privremeno u engine-u.

### Baza podataka

Podatke koji se nalaze u bazi predstavićemo šematski kroz relacije. Svaka klasa poseduje jedinstveni identifikacioni broj.

### 

### Podaci koji se nalaze privremeno u engine-u

* FlavorPreference(userId, flavor)
* AlcoholAmountPreference(referenceId, alcoholStrenght)
* GlassPreference(referenceId, glass)
* TasteProfile(preferenceList, userId)
* PreferableIngridentList(ingridents, referenceId)
* PreferableCocktailList(cocktails, referenceId, templateCocktailsAdded)
* IngridentAlteration(ingridientId, amount)
* LowIngridientAlarm(ingridientId)
* MenuAlteringAlarm(eventId)
* GuestAmount(eventId, amount)
* MaleGuestAmount(eventId, amount)
* FemaleGuestAmout(eventId, amount)
* EventType(eventId, eventType)
* EventGlasses(eventId, glassList)

# Pravila

Kako za većinu pravila koje ćemo implementirati ne postoje određene zakucane granice pravila ćemo objasniti uopšteno.

## Preporuka koktela

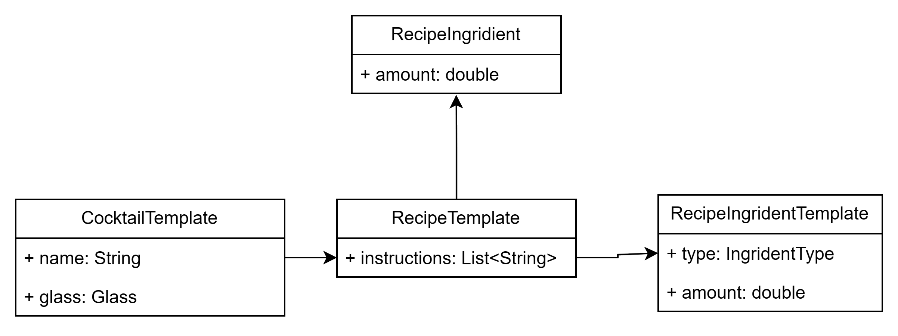
Korisnik unosi *FlavorPreference* i opciono *AlcoholAmountPreference*, *GlassPreference*.

1. Pokupe se svi objekti *FlavorPreference* (1 ili više) sa odgovarajućim *userId* i sa **accumulate** **funkcijom** se grupišu u *TasteProfile*. (*FlavorPreference* se sklanja iz engine-a)
2. Na osnovu *TasteProfil*-a se uzimaju svi sastojci koji ga zadovoljavaju i grupišu se sa **accumulate** **funkcijom** u *PreferableIngridentList*.
   1. Ukoliko je neki sastojak sačinjen od drugih sastojaka rekurzivno se ide do najjednostavnijeg i na taj način dobijamo koji je ukus u pitanju. (**bacward chaining**)
3. Na osnovu *PreferableIngridentList* i *Invetory* dobija se lista koktela koji su sačinjeni od tih sastojaka, gde sastojaka ima dovoljno u *Invetory* i grupišu se u *PreferableCocktailList* gde je fleg *templateCocktailsAdded* postavljen na *false*.
4. *PreferableIngridentList* koji ima fleg *templateCocktailsAdded* postavljen na *false* je ulaz u sledeće pravilo koje se dobija pomoću ***template***-a.\* Kada se dodaju šablonski kokteli flag se postavlja na true.
5. *PreferableCocktailList* i *GlassPreference*, je ulaz u sledeće pravilo. Kokteli se filtriraju tako da zadovolje *GlassPreference*. *GlassPreference* se zatim briše. Ukoliko ne postoji *GlassPreference* ovo pravilo se neće okinuti.
6. *PreferableCocktailList* i *AlcoholAmountPreference*, je ulaz u sledeće pravilo. Kokteli se filtriraju tako da zadovolje *AlcoholAmountPreference* tako što se izračunava procenat alkohola u koktelu. *AlcoholAmountPreference* se zatim briše. Ukoliko ne postoji *AlcoholAmountPreference* ovo pravilo se neće okinuti.
   1. AlcoholAmountPreference može biti:
      1. light(<15%)
      2. medim(<35%, > 15%)
      3. strong(>35%).

Konačan izlaz iz sistema je *PreferableCocktailList*.

\*Neki kokteli se prave na sličan način sa malim modifikacijama. Na primer margarita koktel uvek sadrži tekilu i sok od limete ali je moguće dodati joj i proizvoljan liker.

Struktura objekata koje koristimo za ***template*** izgleda ovako:



Template bi funkcionisao tako što:

1. Dodajemo *CocktailTemplate* objekte u template.
2. *CocktailTemplate* ima *RecipeTemplate* koji ima referencu na *RecipeIngridientTemplate*.
3. Za svaki *RecipeIngridientTemplate* pronađemo u inventaru sve sastojke koji imaju isti tip (*IngridientType*) i čiji ukus odgovara odabranom. Na osnovu sastojaka sa istim tipom kreiramo novi *RecipeIngrident* koji ima isti amount kao *RecipeIngridientTemplate* i referencu na sastojak. Za svaki od ovih sastojaka možemo napraviti novi koktel.
4. Za svaki sastojak kreiramo novi *Cocktail* objekat, postavimo čašu na čašu CocktailTemplate-a. Postavimo ime, na: ime novog sastojka + ime *CocktailTemplate*-a.
5. U koktel objekat postavimo novi *Recipe* objekat koji ima istu listu instrukcija kao *RecipeTemplate*.
6. Dodamo sve sastojke koji su tipa *RecipeIngridient*.
7. Dodamo novokreirani *RecipeIngrident* sastojak.
8. Na kraju preuzmemo sve kreirane koktele.

Ovaj proces rezonovanja ima **ForwardChaining**, **BackwardChainig**, **accumulate** **funkciju** i **template**.

## Praćenje inventara

Korisnik unosi u sistem *Order*.

1. Ulaz u pravilo je *Order* i iz njega se uzimaju *RecipeIngridient* i na osnovu svakog se pravi *IngridentAlteration*.
2. Ulaz u pravilo je *IngridentAlteration* i *Inventory* kojem je *updateTime* bilo pre više od 15min. Promeni se količina sastojaka u Inventaru i ukoliko se neki sastojak ispraznio kreira se *LowIngridientAlarm*.

U ovom procesu reznovanja imamo primenu **CEP**-a.

Primer praćenja inventara

1. Dodaje se order koji ima Cocktail(„Rum and coke“) u sistem.
2. Dodaje se IngridientAlteration(„Rum“, 30) i IngridientAlteration(„Coca Cola“, 90) u sistem.
3. Menja se IngridientInvetory za „Rum“ kome je do sada amount bio 530, spada na 500.
4. Menja se IngridientInvetory za „Coca cola“ kome je do sada amount bio 90, spada na 0.
5. Dodaje se LowIngridientAlarm za „Coca cola“.

## Kreiranje inicijalne karte pića i liste sastojaka

Korisnik u sistem unosi *MaleGuestAmount*, *FemaleGuestAmout*, *EventType* i *Event*.

1. Ulaz u ovaj skup pravila je *EventType* i realizovaćemo preko **template**-a. **Template** ćemo kreirati pomoću *EventGlasses* tako da:
   1. Kada je *EventType* „Birthday“ *EventGlasses* će imati 4 shot čaše, 4 long čaše i 2 martini čaše.
   2. Kada je *EventType* „Buissiness“ *EventGlasses* 4 long čaše, 2 martini i 4 čaše za viski.
   3. Kada je *EventType* „Wedding“ *EventGlasses* će imati 3 long čaše, 3 martini i 2 čaše za viski čaše i dve shot čaše.
   4. Kada je *EventType* „Bachelorette/Bachelor Party“ *EventGlasses* će imati 6 shot čaše, 2 long čaše i 2 martini čaše.

Za svaku od „čaša“ se kreira *GlassPreference*.

1. Ulaz u ovo pravilo je *MaleGuestAmount* i *GuestAmount*. Pretpostavlja se da će 50% muškaraca biti “*medium* *drinkers*” i 50% biti “*strong* *drinkers*”. Zatim se izračunava procenat muškaraca na događaju tako što se podeli *GuestAmount* sa *MaleGuestAmount*. Zatim se od ukupne količine pića na meniju (10) uzme broj pića koji odgovara muškarcima. Za polovinu pića se dodaje *AlcoholAmountPreference* *medim*, a za drugu polovinu *AlcoholAmountPreference* *strong*.
2. Ulaz u ovo pravilo je *FemaleGuestAmount* i *GuestAmount*. Pretpostavlja se da će 50% žena biti “*medium* *drinkers*” i 50% biti “*light* *drinkers*”. Zatim se izračunava procenat žena na događaju tako što se podeli *GuestAmount* sa *FemaleGuestAmount*. Zatim se od ukupne količine pića na meniju (10) uzme broj pića koji odgovara ženama. Za polovinu pića se dodaje *AlcoholAmountPreference* *medim*, a za drugu polovinu *AlcoholAmountPreference* *light*.
3. Ulaz u sledeće pravilo je *GlassPreference* i *AlcoholAmountPreference*. Na osnovu ovih činjenica bira se koktel koji ih zadovoljava i dodaje se u engine.
4. Svi kokteli koji se nalaze u enginu se spajaju **accumulate** **funkcijom** i dodaju u *Event* *menu*.
5. Ulaz u sledeće pravilo je *Event* je dužina menija veća od 0. Na osnovu koktela pravi se spisak sastojaka koji trebaju da se nabave i dodaju se u *Event*.

U ovom procesu reznovanja imamo primenu **ForwardChainiga**, **Template**-a i **accumulate** **funkciju**.

## Izmena karte pića u realnom vremenu

Korisnik u sistem unosi *Order*

1. Ulaz u pravilo je *Order* i iz njega se uzimaju *RecipeIngridient* i na osnovu svakog se pravi *IngridentAlteration*.
2. Ulaz u pravilo je *IngridentAlteration* i *Event*. Promeni se količina sastojaka u inventaru tog *Event*-a.
3. Ulaz u pravilo su *Order*-i koji su se desili u poslednjih 15min. Izračunamo procenat „*strong*“, „*light*“ i „*medium*“ pića i pomnožimo ga sa brojem pića koje treba da dobijemo u karti pića (10) i insertujemo toliko *AlcoholAmountPreference* za svaku jačinu respektivno.
4. Ulaz u pravilo su *Order*-i koji su se desili u poslednjih 15min. Izračunamo procenat pića u određenoj čaši i pomnožimo ga sa brojem pića koje treba da dobijemo u karti pića (10) i insertujemo toliko *GlassPreference* za svaku čašu respektivno.
5. Ulaz u pravilo je *Event* čiji je updateTime bio pre više od 15min. Na osnovu inventara izvuče se *PreferableIngridentList*.
6. Ulaz u pravilo je *GlassPreference*, *AlcoholAmountPreference* i *PreferableIngridentList*. Na osnovu ovoga dobijemo koktel koji zadovoljava ove uslove.
7. Koktele koje smo dobili grupišemo pomoću **accumulate** **funkcije** u *PreferableCocktailList*.
8. Ulaz u ovo pravilo je *Event* čiji je *updateTime* bio pre više od 15min i *PreferableCocktailList*. Koktele iz liste postavimo na kartu pića u eventu. Ažuriramo *updateTime*.

U ovom procesu reznovanja imamo primenu **ForwardChainiga**, **CEP**-a i **accumulate** **funkciju**.

Pravila koja su predstavljena, kao i strukture podataka su podložna manjim promenama radi uspešnije implementacije ili tačnijeg izračuvanja.

Stojanov Dunja, SW-30/2019

Milovanović Miloš, SW-17/2019