

Tartu Ülikool
Arvutiteaduse instituut
Informaatika õppekava

Verner Läll

Päevapakkumiste soovitusüsteem juturoboti kujul

Bakalaureusetöö (9 EAP)

Juhendajad: Laura Ruusmann, Mark Fišel

Tartu 2020

Päevapakkumiste soovitusüsteem juturoboti kujul

Lühikokkuvõte:

Mitmed veebileheküljed võimaldavad inimestel iga päev leida sadu lõunasöögi päevapakkumisi. Nende läbitöötamine ja võrdlemine käsitsi on aeganõudev ja tülikas. Käesolevas töös arendati välja juturobot, mis kasutajale talle sobivaid päevapakkumisi soovitab. Soovitusi teeb kasutajale tema eelnevate valikutega kohanev soovitusüsteem. Päevapakkumised saadetakse kasutajatele Slacki töökeskkonda igal tööpäeval automaatselt. Slack on töö autori töökohas peamine suhtlusvahend ning seetõttu on juturobot integreeritud just selle suhtlusvahendiga. Juturobot saab päevapakkumiste andmed päevapakkumised.ee API-st, mis avaldati autorile juturoboti loomise eesmärgil.

Võtmesõnad:

Slack, juturobot, soovitusüsteem, automatiseerimine

CERCS: P176 - Tehisintellekt

Chatbot as a Lunch Offers Suggestion System

Abstract:

Many web pages help people to find hundreds of lunch offers every day. To filter and compare them to find one suitable is time consuming and frustrating. In this thesis a chatbot was developed that sends lunch offers to users and lets them choose from the suggestions. A recommendation system was also developed for that chatbot that adapts its recommendations according to the users' past choices. Since the author's workplace uses Slack as their primary communication software, the chatbot was integrated with Slack. Lunch offers data is fetched from the päevapakkumised.ee private API, which they made available for this thesis' purposes.

Keywords:

Slack, chatbot, recommendation system, automation

CERCS: P176 - Artificial intelligence

Sisukord

Mõisted ja terminid	6
1. Sissejuhatus	7
Töö eesmärk	7
2. Slack	9
3. Soovitussüsteemid	10
4. Lahenduse kirjeldus	11
4.1 Juturoboti kasutamine	12
4.2 Päevapakkumiste API	14
4.3 Juturoboti server (pp-bot)	15
4.3.1 Slack integratsioon	15
4.3.2 Päevapakkumiste hankimise rutiin	16
4.3.3 Soovituste saatmine	17
4.4 Soovitussüsteemi teenus (pp-classifier)	17
4.4.1 Tunnuste leidmine (/classify)	18
Tunnuste töötlemine	18
Käsitsi märgendamine	19
Kirjeldusest tuletatud tunnused	19
4.4.2 Päevapakkumiste soovitamise (/recommend)	20
4.5 Andmebaas	24
5. Kasutatud tehnoloogiad	26
5.1 TypeScript	26
5.2 JavaScript	26
5.3 Node.js	26
5.4 Bull	27
5.5 Redis	27
5.6 MySQL	27
5.7 Python	28
5.8 Conda	28
5.9 EstNLTK	28
5.10 Flask	28
5.11 Dokku	29
5.12 Docker	29

6. Edasiarendamise võimalused	30
6.1 Tunnuste leidmine	30
6.2 Soovituste tegemine	30
6.3 Erinevate suhtlusvahendite toetamine	31
Kokkuvõte	32
Viidatud kirjandus	33
Lisad	36

Mõisted ja terminid

Juturobot (ingl *chat bot*) - arvutiprogramm, mis simuleerib inimvestlust läbi tehisintelligentsi.

Integratsioon (ingl *integration*) - tervikliku süsteemi järk-järguline koostamine komponentidest.

Server (ingl *server*) - funktsionaalüksus, mis annab andmesidevõrgu kaudu tööjaamadele või muudele funktsionaalüksustele ühiskasutuslikke teenuseid.

JSON (ingl *JavaScript Object Notation*) - andmevahetusformaad.

API (ingl *Application Programming Interface*) - rakendusliides.

Kaldkriipskäsklus (ingl *slash command*) - sõnum, mis algab kaldkriipsuga ja mille saatmisel täidetakse mingi käsk.

1. Sissejuhatus

Lõuna söömine on oluline osa tööpäeva rutiinist ning paljud linnas töötavad inimesed eelistavad toidu kodust kaasa pakkimisele minna restorani või pubisse lõunapakkumist sööma. Kuid sageli leitakse end teadmatuses, mida või kus tänast lõunat süüa. Probleemi lahenduseks võib internetist leida kaks enimkasutatavat päevapakkumiste agregatorit, “päevapakkumised.ee” ja “päevapraad.ee”, kuhu restoranid ja kohvikud oma pakutava päevapakkumise ülesse laadivad. Kahjuks pole kummalgi leheküljel soovitusüsteemi, mis aitaks sadade pakkumiste hulgast välja valida lehekülastajale sobiv variant.

Kasutades igapäevaselt tööl suhtlusvahendit Slack, mis võimaldab firmasiseselt meilidest kiiremini informatsiooni jagada, tekkis mõte ehitada soovitusüsteem, mis Slacki vahendusel töö autorile ja töökaaslastele päevapakkumisi saadaks. Töö autor võttis ühendust päevapakkumised.ee lehekülje omanikuga, kes oli nõus juturoboti arendamise jaoks rakendusliidese avaldama.

Töö eesmärk

Lõputööga eesmärgiks on luua juturobot, mis aitab tõsta töökohas produktiivsust ja vähendada söömiskoha valimise ajakulu. Valminud juturobot peaks töötama igas linnas, mida päevapakkumised.ee lehekülg pakub. Carlene Lebeuf jt. väidavad, et selleks, et vältida ühe tüütuse teisega asendamist, peaks robotiga suhtlemine olema optimeeritud. Kasutaja peaks tegema võimalikult vähe klõpse ja kirjutama võimalikult vähe teksti, et juturobotiga suhelda [1].

Töö tulemusena valmivat juturobotit saab kasutada Slacki töökeskkonnas. Juturoboti eesmärk on registreeritud kasutajatele igal tööpäeval kella kaheistkümne paiku saata sellele kasutajale sobivaid päevapakkumisi. Sobivate pakkumiste valimiseks valmib töö tulemusena ka soovitusüsteem. Pakkumised valib soovitusüsteem välja kasutaja eelnevate valikute põhjal, ning soovitused peaksid paranema iga päevaga, kui kasutaja jälle uue valiku teeb.

Töö teises peatükis tehakse lugeja tuttavaks Slackiga. Kolmandas peatükis kirjeldatakse lühidalt soovitusüsteeme ning nende kasutusalasid. Neljandas peatükis kirjeldatakse põhjalikult

juturoboti ja soovitusüsteemi tööpõhimõtteid. Viiendas peatükis tuuakse esile lahenduses kasutatud tehnoloogiad. Kuuendas peatükis esitatakse mitmeid erinevaid edasiarendamise võimalusi.

2. Slack

Slack on vlkvestluse phine suhtlusvahend efektiivseks infovahetuseks ja kaugto korraldamiseks, mille eesmrk on krvaldada vajadus kasutada firmasiseseks suhtlemiseks e-kirjasid. Slackil vimaldab teha mitme inimesega korraga knesid, videoknesid, jagada videoknes ekraanipilti, jagada faile ning otsida ja vaadata kogu vestluste ajalugu. Erinevalt e-kirjadest, kus sissetulevad kirjad saabuavad kik hte postkasti, toimuvad Slackis vestlused organiseeritult avalikes kanalites (ingl *channel*), peidetud kanalites (ingl *private channel*) ja otsestes snumites (ingl *direct message*).

Avalikes kanalites saavad osaleda kik tokeskkonnas (ingl *workspace*) olevad kasutajad. Peidetud kanaleid nevad ainult kanalisse lisatud kasutajad. Otsestes snumites vestlevad kaks kasutajat ainult omavahel.

Lisaks sellele on Slackis tehtud lihtsaks erinevate rakenduste integreerimine suhtluskeskkonda. Niteks on vimalik koodis juhtuvaid erindeid kinni puda Bugsnagi abil ning lasta need saata Slack kanalisse #bugs [2]. See annab arendajatele reaajas teada koodis esinenud vigadest ilma kasutajate poolse tagasisideta, mille tulemusena on vimalik need vead kiiremini parandada.

Rakendused Slackis on oma olemuselt nagu teised kasutajad, kuid nende lisamisel tokeskkonda neb tokeskkonna haldaja, mis iguseid integratsioon tokeskkonnas vajab. Kui integratsioonile on vastavad igused antud, saab see lugeda ja kirjutada kanalitesse, alustada vestluseid kasutajatega vi isegi hallata tokeskkonda. Lisaks saab integratsiooniga suhelda kaldkriipskskluste (ingl *slash command*) abil.

Kaldkriipsksklused on mugavad, sest neid saab kasutada olenemata sellest, mis kanalis vi vestluses kasutaja parasjagu on. Saadavalolevaid kskluseid ning nende kirjeldusi on vimalik nha rakenduse ja kasutaja vahelises privaatvestluses.

3. Soovitussüsteemid

Soovitussüsteemid on tarkvaralised informatsiooni filtreerimise lahendused, mis esitavad kasutajale tema mingite tunnuste põhjal pakkumisi, mis võiksid talle kasulikud olla [3]. Soovitussüsteemidel on väga lai kasutusala, seetõttu võib soovitatud pakkumiseks olla ükskõik, mis: toode, teenus, lõunapakkumine või midagi muud.

Soovitussüsteemid on muu hulgas kasutusel internetipoodides, kus kasutajale pakutakse e-kirja teel varem vaadatud toodetele sarnaseid tooteid või eelnevalt ostetud toodete uuemaid väljalaskeid [4]. Masinõppega soovitussüsteeme kasutatakse ka Netflix'is [5] või YouTube'is [6], et näidata kasutajale soovituste hulgas videosid või filme, mida nad kõige suurema tõenäosusega järgmisena vaadata soovivad.

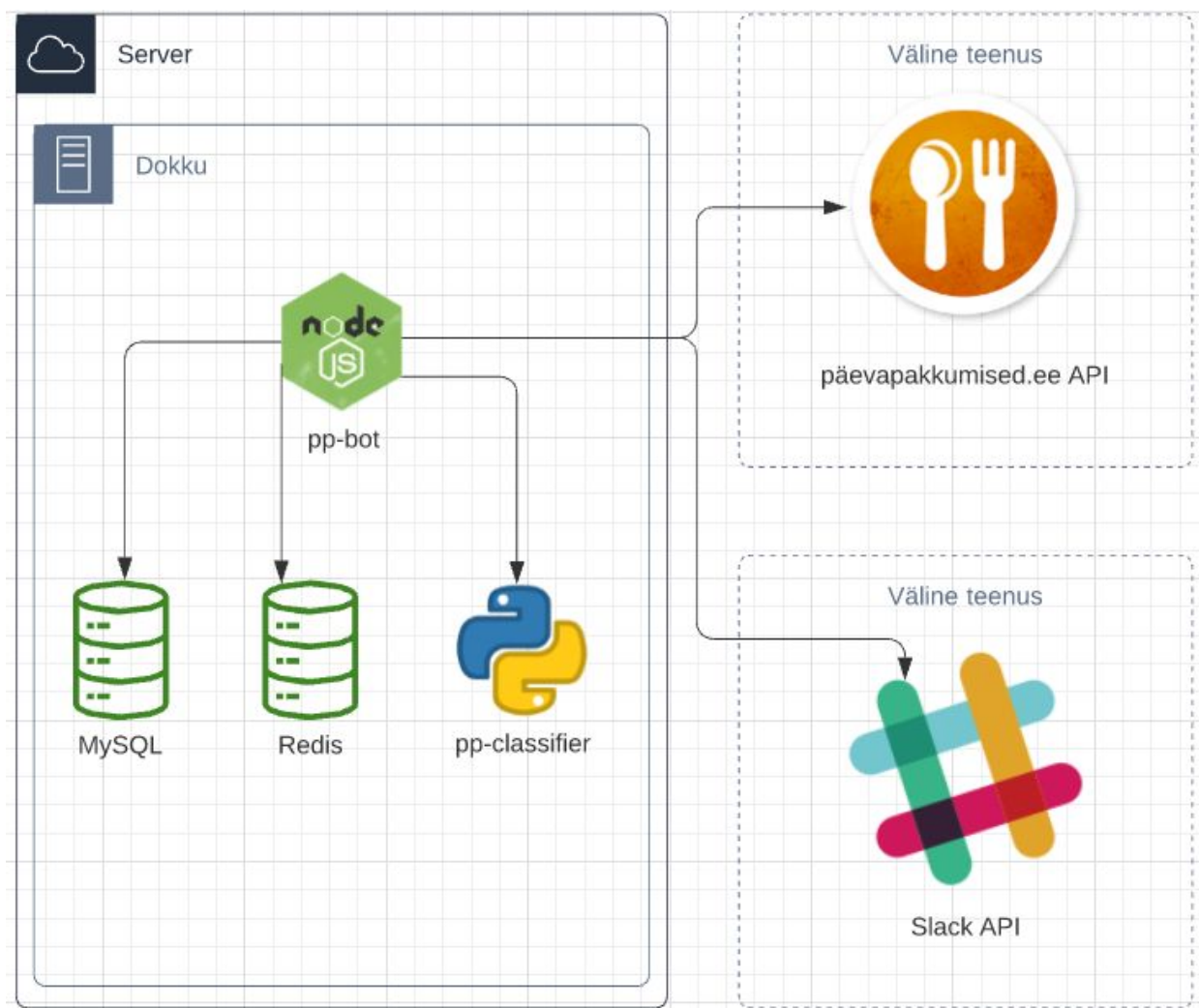
Marco de Gemmis jt kirjutavad, et tüüpilise soovitussüsteemi kõige olulisem samm on kasutaja eelistuste õppimine. Õppida saab nii kasutajalt otse küsides kui ka kasutaja käitumist jälgides. Nende sõnul on asjalike soovituste tegemise probleemi põhjalikult uuritud ning on esile tõusnud kaks tüüpilisemat mudelit. Sisupõhised soovitussüsteemid teevad soovitusi põhiliselt kasutaja eelnevate valikute põhjal ning kollaboratiivsed soovitussüsteemid teevad soovitusi olenevalt teiste sarnaste kasutajate tehtud valikutest. Kasutuses on ka hübriidsoovitussüsteemid, mis kasutavad ära mõlema tüüpilise mudeli tugevusi [7].

Külma stardi probleem

Külma stardi probleemiks nimetatakse soovitussüsteemide puhul olukorda, kus uuele kasutajale ei ole võimalik täpseid soovitusi teha, sest tema kohta puudub vajalik varasemate valikute informatsioon [8]. Lõputöös valminud juturoboti puhul peab probleemi lahendamiseks tegema kasutaja paar esimest valikut ise, kasutades talle saadavaks tehtud pakkumiste või restoranide otsingu funktsionaalsust.

4. Lahenduse kirjeldus

Käesolev peatükk annab ülevaate lõputöö tulemusena valminud teenuste ülesannetest, funktsionaalsustest, arendusprotsessist ja kasutatud tehnoloogiatest. Juturobot koosneb kahest sisemisest ja kahest välimisest teenusest (vt joonis 1).



Joonis 1. Juturoboti ülesehitus sisemiste ja välimiste teenuste näol.

Sisemisteks teenusteks on *Node.js HTTP* serveri peale ehitatud juturobot (*pp-bot*) ja *Python Flask* raamistiku peale ehitatud päevapakkumiste soovitusüsteem (*pp-classifier*). Välimisteks teenusteks on **päevapakkumised.ee API**, kust saadakse info tänaste päevapakkumiste kohta, ja **Slack API**, mille kaudu toimetatakse soovitused kasutajatele ning võetakse vastu kasutajapoolset sisendit.

4.1 Juturoboti kasutamine

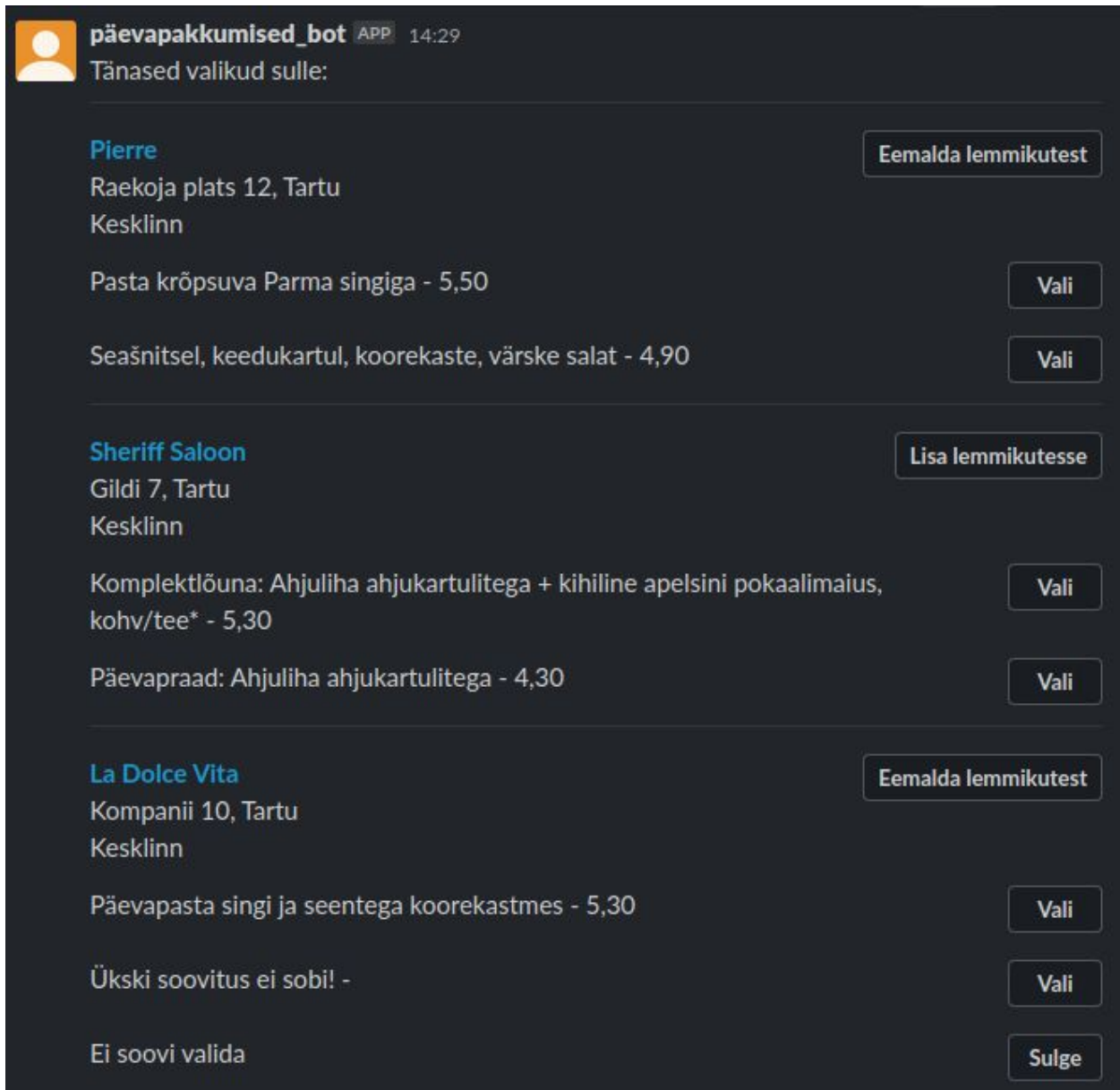
Juturoboti lisamiseks Slacki töökeskonda tuleb töökeskkonna administraatoril külastada <https://pp-bot.lall.ee/> ning vajutada nupule “*Sign in with Slack*” (“Logi sisse Slacki kasutajaga”). Seejärel avaneb vaade, mis annab teada, mis õiguseid rakendus nõuab ja mida rakendus Slacki töökeskkonnas teha saab.

Pärast rakenduse lisamist töökeskkonda on kõikidel töökeskkonnas olevatel kasutajatel võimalik juturobotiga suhelda, kasutades järgmisi kaldkriipskäskluseid:

1. **/start** - registreerib kasutaja saamaks igapäevaseid päevapakkumiste soovitusi. Kui kasutaja registreerib esimest korda, palutakse tal teha ka asukoha valik. Kui kasutaja on vahepeal kirjutanud **/stop**, kuid tahab uuesti soovitusi saada, siis on asukoha valik juba salvestatud ja seda enam ei küsita.
2. **/otsi otsingusõna** - laseb kasutajal otsida restorane nime järgi.
3. **/tahan otsingusõna** - laseb kasutajal otsida päevapakkumisi pakkumises sisalduva sõna järgi.
4. **/lemmikud** - näitab kasutajale tema lemmikutesse lisatud söögikohtasid koos nende pakkumistega.
5. **/soovita** - teeb kasutajale päevapakkumiste soovitused, näiteks juhul, kui kasutaja tahab soovitusi saada varem kui kell 12.
6. **/seaded** - laseb kasutajal muuta enda asukohta või igapäevaste soovituste allikat, milleks võib olla soovitusüsteem või kasutaja lemmikrestoranide pakkumised.

7. **/stop** - peatab kasutajale igapäevaste pakkumiste saatmise. Kasutajal on endiselt võimalik kasutada kõiki teisi kaldkriipskäsklusi, et leida endale sobiv söögikoht.

Kasutajad, kes on sisestanud käsu “start” ning valinud enda linna, hakkavad saama soovitusi iga päev kell 12. Külma stardi probleemi vältimiseks peaksid uued kasutajad kasutama esimesed paar päeva käsklusi “otsi” ja “tahan”, et leida endale sobivad päevapakkumised.



Joonis 2. Juturoboti sõnum, mis näitab kasutajale restorane ning nende päevapakkumisi ja valikuid.

Käsud “otsi”, “tahan”, “lemmikud” ning “soovita” tagastavad kasutajale sarnasel kujul tulemuse (vt joonis 2). Päevapakkumise kõrval oleva nupu “vali” vajutamisel salvestatakse kasutaja valik andmebaasi ning järgmisel soovitamisel võetakse seda juba arvesse. Ülejäänud pakkumiste valikute nupud kustutatakse ning valik tehakse kasutajale nähtavaks. Restorani nime peale vajutades suunatakse kasutaja päevapakkumiste lehele, kus on võimalik näha restorani kohta lisainfot. Restorani kõrval oleva nupu “Lisa lemmikutesse” vajutamisel salvestatakse restoran kasutaja lemmikuks. Kasutajal on võimalik oma lemmikute restoranide päevapakkumisi näha kirjutades käsu **/lemmikud**. Teine võimalus on see, kui kasutaja on seadetest määranud, et soovib lõuna ajal saada pakkumisi lemmikutest, mitte soovitusüsteemilt. Nupu “Sulge” vajutamisel kustutab juturobot oma sõnumi.

4.2 Päevapakkumiste API

```
{
  "offerings": [
    {
      "meals": [
        {
          "price": "3,50",
          "description": "Guljašš sealihast, keedetud kartul ja värsked salat",
          "id": "1"
        },
        {
          "price": "3,50",
          "description": "Guljašš sealihast, keedetud kartul ja värsked salat",
          "id": "2"
        }
      ],
      "diner": {
        "id": 2087492860,
        "name": "Vanameistri PUBI",
        "address": "Rahu 15A, Jõhvi",
        "district": null,
        "homepage": "http://www.vanameistripubi.ee",
        "coordinates": {
          "h": "59.361572",
          "l": "27.400728"
        },
        "nameAsUrl": "vanameistri-pubi",
        "id": "2"
      }
    }
  ]
}
```

Joonis 3. <https://paevapakkumised.ee/andmed/slack/linnad/linn> vastuse näidis, kus antud lõputöö raames ebaoluline informatsioon on asendatud punktidega.

Päevapakkumised.ee avaldab ühe juturoboti jaoks vajaliku HTTP (ingl *Hypertext Transfer Protocol*) päringu GET meetodiga marsruudi (ingl *URL*), kus aktsepteeritavateks linna sisenditeks on “tallinn”, “tartu”, “parnu”, “haapsalu”, “kuressaare”, “rakvere”, “viljandi”, “johvi”, “valga” ja “muu-eesti”. Marsruut tagastab JSON kujul vastuse (vt joonis 3).

4.3 Juturoboti server (*pp-bot*)

Juturoboti serveri ülesandeks on suhtlemine kõikide teiste rakenduse töötamiseks vajalike teenustega. Põhiline ülesanne on igapäevaselt lõuna ajal läbida rutiin, mis toimetab registreeritud kasutajatele Slackis päevapakkumised, millest nad huvitatud võiksid olla. Peale selle kuulab server Slackist kasutaja poolt tehtud käsked ja salvestab kasutaja valikud andmebaasi.

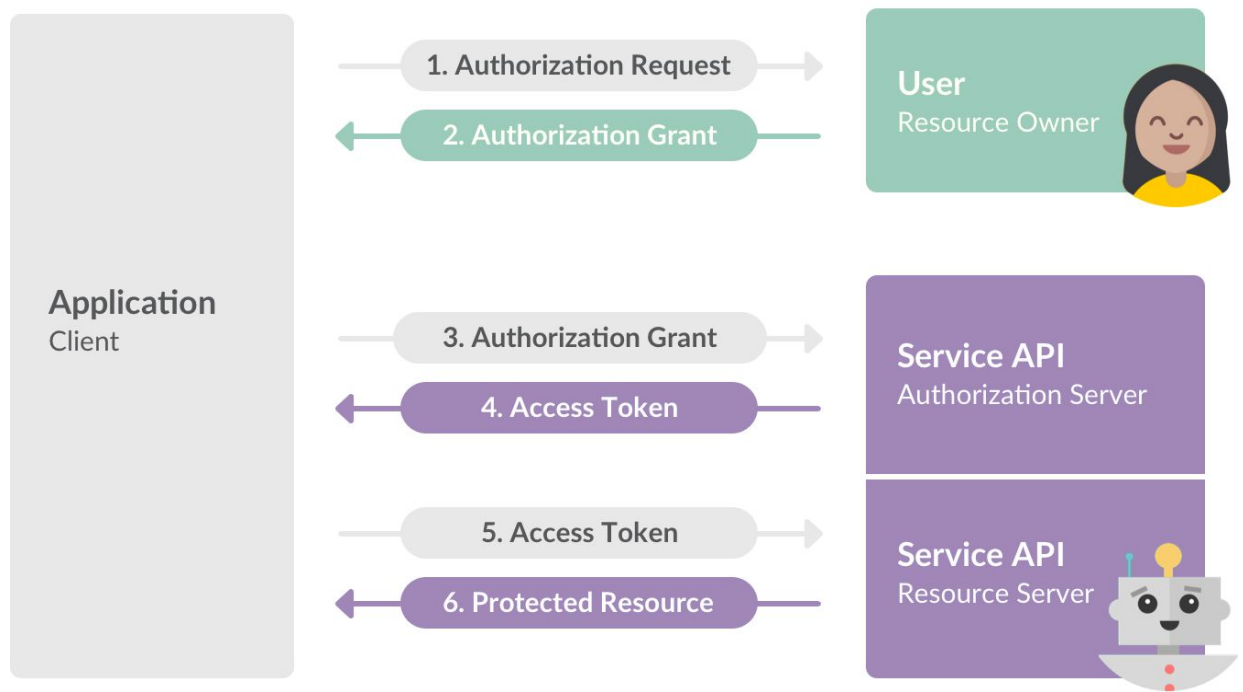
4.3.1 Slack integratsioon

Juturoboti lisamiseks Slacki töökeskkonda on avaldatud serveris kaks HTTP GET marsruuti. Serveri juurmarsruut (*pp-bot.lall.ee*) tagastab veebilehe, kus asub “*Sign in with Slack*” nupp. Nupp sisaldab informatsiooni nõutavate õiguste kohta ja Slacki poolt antud rakenduse tuvastamise koodi. Nupu vajutamisel suunatakse kasutaja edasi Slacki domeenil olevale rakenduse volitamise (ingl *authorization*) lehele. Juturoboti rakendus küsib Slacki töökeskkonnalt kaks õigust (vt joonis 4 punkt 1).

1. **chat:write** - see õigus lubab juturoboti rakendusel saata kasutajatele privaatsõnumeid.
2. **commands** - see õigus lubab juturoboti rakendusel kuulata Slacki töökeskkonnas kaldkriipskäskluseid.

Slacki volitamise leht suunab peale kasutaja nõustumist või mitte nõustumist tagasi marsruudile */auth*, tagastades serverile Slacki töökeskkonna ligipääsu saamiseks vajaliku ajutise ligipääsu nõusoleku koodi (ingl *authorization grant*) (vt joonis 4 punkt 2). Server saadab selle koodi koos rakenduse tuvastamise koodi ja rakenduse salajase võtmega tagasi Slacki API-le, ning saab vastu Slacki töökeskkonna ligipääsu loa (ingl *authorization token*) (vt joonis 4 punkt 3).

ja 4). Ligipääsu luba kasutab juturoboti server Slacki töökeskkonna kasutajatele sõnumite saatmiseks.



Joonis 4. Slacki rakenduse volitamise protsess [9].

Kasutaja sisendi kuulamiseks on juturoboti serveris avaldatud kaheksa HTTP POST marsruuti - üks iga kaldkriipskäskluse jaoks ning “/interact”, mis võtab vastu kasutaja nupuvajutuste sõnumeid Slackist ning saadab neile vastuse.

4.3.2 Päevapakkumiste hankimise rutiin

Serveri rutiin käivitatakse automaatselt igal tööpäeval mitu korda kella 11 ja 13 vahel. Rutiini alguses küsib server päevapakkumised.ee API-st tänased päevapakkumised. Server salvestab tulemuse andmebaasi ning saadab päevapakkumised edasi soovitusüsteemi teenusele, mis leiab iga päevapakkumise kirjelduse alusel soovitusüsteemi jaoks vajalikud tunnused. Need saadetakse tagasi serverisse ja salvestatakse andmebaasi. Päevapakkumiste hankimise rutiin on lõppenud.

Rutiini on vajalik käivitada mitu korda, sest kasutaja võib tahta otsida pakkumisi enne, kui need automaatselt välja saadetakse. Samuti on neid vajalik küsida pärast kella 12, sest mõned restoranid jäävad oma pakkumiste lisamisega hiljaks.

4.3.3 Soovituste saatmine

Kell 12 käivitatakse serveris päevapakkumiste kasutajatele soovitamise rutiin. Server laeb andmebaasist linnade kaupa grupeeritult kõik kasutajad, kes soovivad regulaarselt soovitusi saada. Järgmiseks laeb server andmebaasist kõik selle linna restoranid, nende tänased päevapakkumised ning kasutajate varasemad valikud. Kasutajate varasemad valikud ja tänased päevapakkumised saadetakse kõik korruga edasi soovitussüsteemi teenusele. Igale kasutajale, kes pole veel juturobotilt soovitusi saanud, koostatakse olenevalt kasutaja seadetest kas tagastatud soovituste järgi või tema valitud lemmikute järgi Slacki API-le sobiva kujuga sõnum. See sõnum saadetakse Slacki päevapakkumiste juturoboti ning kasutaja vahelisse privaatvestlusesse. Kui soovituste saatmine õnnestus, salvestatakse andmebaasi kasutajate tabelisse viimane õnnestunud soovitamise kuupäev. Soovituste saatmise rutiin on lõppenud.

4.4 Soovitussüsteemi teenus (*pp-classifier*)

Soovitussüsteemi teenusel on kaks ülesannet. Mõlema ülesande jaoks avaldatakse HTTP POST meetodiga marsruut, mille sisendiks ja väljundiks on JSON vormingus andmed. Esimene ülesanne on iga päevapakkumise kohta leida soovituste tegemiseks vajalikud tunnused. Neid tunnuseid kasutab teenus hiljem, et võrrelda päevapakkumiste sarnasust ning sobivust kasutajaga. Teine soovitussüsteemi ülesanne on päevapakkumiste järjestamine igale kasutajale nende omavahelise sobivuse järgi.

Ülesanded on jagatud kaheks, sest päevapakkumiste tunnuste otsimine on ajakulukas ning seda on võimalik teha kohe pärast päevapakkumiste hankimist. Sellisel moel on võimalik soovitusi teha kiiremini ja efektiivsemalt, sest tunnused leitakse ainult ühe korra - enne, kui on aeg soovitusi saata.

4.4.1 Tunnuste leidmine (*/classify*)

Tunnusteks (ingl *feature*) nimetatakse asjade või nähtuste iseloomulikke omadusi, mille alusel on neid võimalik omavahel võrrelda või eristada [10]. Päevapakkumiste puhul võiks olulisteks tunnusteks lugeda näiteks päevapakkumise kirjelduse igat sõna või seda, kas tegemist on lihavaba toitumist harrastavale sobiva pakkumisega. Marsruut võtab JSON vormingus sisendiks joonisel 5 välja toodud *ClassifierRequest* andmestruktuuri ning tagastab JSON vormingus *ClassifierResponse* andmestruktuuri.

```
export interface ClassifierRequest {
  meals: Array<{
    id: number;
    description: string;
  }>;
}

export interface ResultMeal {
  brackets: Array<string>;
  category: string | null;
  classifier: string;
  description: string;
  id: number;
  ingredients: Array<string>;
  lemmas: Array<string>;
  meat: Array<string>;
  vegetarian: boolean | null;
  error?: boolean;
}

export interface ClassifierResponse {
  meals: Array<ResultMeal>;
}
```

Joonis 5. Marsruudi */classify* sisendi ja väljundi andmetüübid.

Tunnuste töötlemine

Päevapakkumise kirjelduse (ingl *description*) puhul on tegemist vabas vormis kirjutatud tekstiga. Kui sõnahaaval võrrelda “praekartulid koos hapukapsaga” ja “hapukapsas praekartulitega”, siis näevad need arvuti jaoks välja kui kaks erinevat päevapakkumist, kuigi tegelikkuses sisaldavad kirjeldused sama informatsiooni. Sellise probleemi lahendamiseks rakendatakse loomuliku

keele töötlemise meetodeid, nagu kirjelduse lemmatiseerimine, väiketähestamine (ingl *converting to lowercase*) ja stoppsõnade eemaldamine.

Lemmatiseerimine on sõnade viimine nende algvormi [11]. Stoppsõnadeks nimetatakse keeletehnoloogias sõnu, mis teksti sisu analüüsimisel ütlevad selle sisu kohta vähe. Päevapakkumise puhul võib stoppsõnadeks lugeda kõiki sõnu, mis ei kirjelda kuidagi pakutavat toitu. Lisaks eemaldatakse kirjelduse hulgast kõik kirjavahemärgid ning muud sümbolid. Kui eelmises lõigus antud näited lemmatiseerida, jääb alles “praekartul koos hapukapsas” ning “hapukapsas praekartul”. Kuna “koos” ei ütle midagi toidu kohta, võime selle sõna eemaldada. Nüüd kui võrrelda pakkumiste tunnuseid, on näha, et mõlemas pakkumises esineb sõna hapukapsas ja praekartul ning et tegu on sarnaste pakkumistega.

Andmete uurimisel selgus, et tihti peale märgitakse sulgudesse pakkumise kohta lisainformatsiooni. Näiteks “peekoniburger (180g loomahakkliha pihv, cheddar juust, jäasalat, peekon, western remulaad ja bbq kaste)”. Sulgudest leitud lemmasid käitleb soovitusüsteem eraldi tunnustena kirjeldusest leitud lemmadest. Sedasi on hiljem soovitusüsteemil võimalik teada, mis oli pakkumises välja toodud esmase ning mis teisejärgulise teabena.

Käsitsi märgendamine

Töö käigus töötati läbi ligi 7000 päevapakkumist, et määrata erinevatele tunnustele vastavaid lemmades esinevaid märksõnasid. Tunnuste ja märksõnade vahelised seosed on töö autori poolt käsitsi defineeritud. Näiteks lemma “kanasupp” sisaldab märksõna “kana” ja märksõna “supp”, mille tulemusel märgitakse pakkumise kategooriaks “supp”, lihakategooriaks “kana” ja taimetoidu tunnus negatiivseks. Kokku tehti vahet kümne kategooria vahel ja kolme lihakategooria vahel, mida kirjeldatakse järgmises alampeatükis.

Kirjeldusest tuletatud tunnused

Ainult lemmade võrdlemisest ei piisa, et näha seost näiteks seljanka ja borši vahel. Sellel põhjusel sai lisatud üheks otsitavaks tunnuseks päevapakkumise kategooria. Tunnuse võimalikeks väärtusteks said valitud järgmised kategooriad: “*sushi*”, “supp”, “pitsa”, “pasta”,

“puder”, “riis”, “magustoit”, “burger”, “salat”, “praad”. Kategooriad on töö autori subjektiivne üldistus toitude liigitamiseks. Näiteks liigitatakse burgeri kategooriasse ka *wrap*-id, einesaiad, *taco*-d ning muud autori arvates sarnased toidud.

Kategooria määramiseks otsib soovitusüsteem päevapakkumise lemmade seest etteantud märksõnasid. Samuti arvestatakse eelnevalt leitud liha kategooriate ja taimetoidu tunnustega. Kui toitu ei õnnestu kategoriseerida, jääb kategooria tunnus määramata.

Märksõnu otsides määrab soovitusüsteem päevapakkumisele ka liha kategooriad ning taimetoitlasele sobiva pakkumise tunnuse. Liha kategooriateks said valitud “lind”, “kala” ja “loom”. Üks toit võib sisaldada mitme liha tunnust. Näiteks oleks “kanaburger peekoniga” nii looma- kui linnuliha tunnusega. Taimetoitlasele sobivuse tunnus määratakse selle järgi negatiivseks, kui pakkumine sisaldas mõnda liha tunnust. Tunnus määratakse positiivseks, kui lemmades on mõni kindlalt taimetoitlasele sobivust tähistav märksõna, näiteks “vegan”, “vege” või “taimne”. Märksõnade puudumisel jäetakse tunnus määramata.

4.4.2 Päevapakkumiste soovitamine (*/recommend*)

Marsruudi sisendiks on joonisel 6 välja toodud JSON vormingus andmestruktuur *RecommenderRequest*. Päevapakkumiste soovitamise puhul kasutatakse sisupõhist soovitamist. Igale kasutajale määratakse tema eelnevate päevapakkumiste valikute põhjal kasutajale omased tunnused. Tabel 1 esitab kasutajate tunnuste nimetusi ja nende kirjeldusi.

Tabel 1. Kasutajatele päevapakkumiste valikute põhjal määratud tunnused.

Tunnuse nimetus	Kirjeldus
lemma_freq	Valitud päevapakkumiste lemmade sagedustabel. Sagedustabel näitab, mitu korda esines iga lemma kasutaja valikute seas.
category_freq	Valitud päevapakkumiste kategooriate sagedustabel. Sagedustabel näitab, mitu korda esines iga kategooria kasutaja valikute seas.
l_coordinate	Valitud päevapakkumiste restoranide aritmeetiline keskmine laiuskraad.
h_coordinate	Valitud päevapakkumiste restoranide aritmeetiline keskmine pikkuskraad.
max_distance	Valitud päevapakkumiste restoranide suurim kaugus restoranide keskmistest koordinaatidest meetrites.
min_distance	Valitud päevapakkumiste restoranide vähim kaugus restoranide keskmistest koordinaatidest meetrites.
avg_distance	Valitud päevapakkumiste restoranide keskmine kaugus restoranide keskmistest koordinaatidest meetrites.
choices_count	Tunnuste leidmiseks kasutatud valikute arv.
vegan_count	Positiivse taimetoidu tunnusega valikute arv.
vegan_maybe_count	Määramata taimetoidu tunnusega valikute arv.
meat_count	Suvalist liha tunnust sisaldavate valikute arv.
meat_freq	Valitud päevapakkumiste liha tunnuste sagedustabel. Sagedustabel näitab, mitu korda esines iga liha kategooria kasutaja valikute seas.

Valikutega antakse lisaks eelnevalt leitud päevapakkumiste tunnustele kaasa ka restorani koordinaadid (vt joonis 6 *DbClassifiedMeal*). Seejärel võrreldakse päevapakkumiste tunnuseid kasutaja tunnustega ning arvutatakse pakkumise sobivus kasutajale.

```

export interface Suggestions {
  user_id: string;
  team_id: string;
  meal_ids: Array<number>;
  debug?: any;
}

export interface RecommenderResponse {
  suggestions: Array<Suggestions>;
  errors: Array<any>;
}

export interface RecommenderRequest {
  users: Array<{
    user_id: string;
    team_id: string;
  }>;
  meals: DbClassifiedMeal[];
  choices: DbClassifiedChoice[];
  debug: boolean;
}

export interface DbClassifiedMeal {
  id: number;
  diner_id: number;
  price: string;
  description: string;
  classifier: string;
  category: string;
  lemmas: Array<string>;
  brackets: Array<string>;
  meat: Array<string>;
  ingredients: Array<string>;
  vegetarian: boolean | null;
  l_coordinate: number | null;
  h_coordinate: number | null;
}

export interface DbClassifiedChoice extends DbClassifiedMeal {
  user_id: string;
  team_id: string;
  suggestions: Array<number>;
  created_at: string;
}

```

Joonis 6. Marsruudi */recommend* sisendi (*RecommenderRequest*) ja väljundi (*RecommenderResponse*) andmetüübid.

Päevapakkumise sobivuse arvutamiseks eemaldatakse pakkumises korduvad lemmad ning liidetakse *lemma_freq* sagedustabeli alusel kokku nende esinemiste arv valikutest. Seejärel jagatakse saadud summa läbi päevapakkumise lemmade arvu ja *lemma_freq* sagedustabelis esinevate unikaalsete lemmade arvu summaga. See tagab, et sõnarohkemad päevapakkumiste kirjeldused ei saa suuremat sobivust tänu oma sõnarohkusele. Tulemuseks on *lemma_score* (vt joonis 7).

```

lemma_score = 0
lemmas = set(meal['lemmas'])
for lemma in lemmas:
  lemma_score += features['lemma_freq'][lemma]
lemma_score /= len(lemmas) + len(features['lemma_freq'])

```

Joonis 7. Skoori *lemma_score* arvutamine. Muutuja “*lemmas*” on päevapakkumise lemmade hulk, kust korduvad lemmad on eemaldatud. Muutuja “*features*” on kasutaja tunnuste kogum (ingl *dictionary*).

Järgmiseks võetakse *category_freq* sagedustabelist päevapakkumise kategooria esinemise arv ning jagatakse läbi tunnuste leidmiseks kasutatud valikute arvuga *choices_count*. Tulemuseks on *category_score*.

Mõne restorani kohta puudub koordinaatide informatsioon. Sellisel juhul on päevapakkumise *distance_score* (vt joonis 8) tulemuseks null. Vastasel juhul arvutatakse kasutaja valikute keskmisi koordinaate ning päevapakkumise koordinaate kasutades päevapakkumise kaugus. See lahutatakse kasutaja keskmisest valikute kaugusest *avg_distance* ning saadud tulemus jagatakse läbi kasutaja suurima valikute kaugusega *max_distance*.

```
has_coordinates = meal['l_coordinate'] is not None and meal['h_coordinate'] is not None and features['max_distance'] != 0
if has_coordinates:
    distance = self._getCoordinatesDistanceInMeters(
        features['l_coordinate'], features['h_coordinate'],
        meal['l_coordinate'], meal['h_coordinate'])
    distance_score = (features['avg_distance'] - distance) / features['max_distance']
else:
    distance_score = 0
```

Joonis 8. Skoori *distance_score* arvutamine. Muutuja “*meal*” sisaldab päevapakkumise tunnuseid. Muutuja “*features*” on kasutaja tunnuste kogum.

Soovitussüsteemis ei arvestata, millist lihakategooriat kasutaja eelistab. Selle asemel arvutatakse, mitu protsenti kasutaja valikutest ei olnud liha tunnustega (*veg_percent*) ning kui kindlalt sisaldub pakkumises liha (*meat_prob*). Kui päevapakkumisel on liha tunnused, on *meat_prob* võrdne ühega. Kui päevapakkumisel ei ole liha tunnuseid ja tema taimetoidu tunnus on negatiivne, siis *meat_prob*=0,75. Kui päevapakkumisel ei ole liha tunnuseid ja tema taimetoidu tunnus on määramata, siis *meat_prob*=0,5. Kõikidel teistel juhtudel on *meat_prob* null. Sellise valemi (vt joonis 9) kasutamisel annab lihavaba toit lihasööjale negatiivse, taimetoitlasele aga positiivse tulemuse.

```

if len(meal['meat']) > 0:
    meat_prob = 1
elif meal['vegetarian'] is False:
    meat_prob = 0.75
elif meal['vegetarian'] is None:
    meat_prob = 0.5
else:
    meat_prob = 0

veg_percent = (features['vegan_count'] + features['vegan_maybe_count']) / features['choices_count']
meat_score = 4 * (meat_prob - 0.5) * (0.5 - veg_percent)

```

Joonis 9. Skoori *meat_score* arvutamine. Muutuja “*features*” on kasutaja tunnuste kogum.

Lõpptulemuse saamiseks liidetakse *lemma_score*, *category_score*, *distance_score* ja *meat_score* kokku. Pakkumised reastatakse lõpptulemuse järgi ning tagastatakse JSON vormingus joonisel 6 välja toodud *RecommenderResponse* andmestruktuurina.

4.5 Andmebaas

Tabel 2. Andmebaasi tabelite ja nende sisu kirjeldused.

Tabel	Kirjeldus
oauth	Sisaldab Slack töökeskkondade ligipääsu lubasid ja informatsiooni töökeskkonna kasutaja kohta, kes juturoboti sinna lisas.
users	Sisaldab kasutajate informatsiooni, kes on läbinud registreerimisprotsessi (<i>/start</i>) juturoboti kasutamiseks.
favorites	Sisaldab kasutajate salvestatud lemmikrestorane.
choices	Sisaldab kasutajate valikuid ning soovitatud pakkumiste ID numbreid.
classifications	Sisaldab soovitusüsteemi poolt leitud päevapakkumiste tunnuseid.
meals	Sisaldab Päevapakkumised.ee API-st hangitud päevapakkumiste kirjeldust, hinda ja restorani ID numbrit.
diners	Sisaldab Päevapakkumised.ee API-st hangitud restoranide andmeid.

Andmebaasis hoiustatakse Slacki töökeskkondade ligipääsude lubasid, päevapakkumiste kirjeldusi ja muid rakenduse tööks vajalikke andmeid. Andmebaasi tabelid ja tabeli veergude vahelised seosed (vt lisa 1) on defineeritud struktuurpäringukeeles. Tabel 2 esitab andmebaasis kasutatavate tabelite nimesid ja kirjeldusi.

5. Kasutatud tehnoloogiad

Antud peatükis tehakse ülevaade juturoboti realiseerimiseks kasutatud tehnoloogiatest ning nende seostest.

5.1 TypeScript

Päevapakkumiste juturobot (*pp-bot*) on kirjutatud TypeScript uusimat versiooni 3.7. Tegemist on avatud lähtekoodiga staatilise tüübikontrolli võimalustega programmeerimiskeelega, mille kasutamine teeb koodi kirjutamise, lugemise ning mõistmise lihtsamaks tänu muutujate ja funktsioonide tüüpimisele ning kompilaatori tasemel veakontrollide tegemisele [12]. Staatilise tüübikontrolli abil on koodis võimalik vead tuvastada enne tema käivitamist, mis lihtsustab arendamist. TypeScriptis on kehtiv ka JavaScripti kood ning kuna ta kompileeritakse JavaScriptiks on tema käitusaegne käitumine võrdne JavaScriptiga [13].

5.2 JavaScript

JavaScript on programmeerimiskeel, mis on kõige tuntum veebilehtede skriptimiskeelena. Enamus veebilehti kasutavad veebilehele funktsionaalsuse lisamiseks JavaScripti [14] ning kõik laialtkasutatavad veebilehitsejad võimaldavad JavaScripti kasutamist [15]. Sellest hoolimata kasutatakse JavaScripti ka paljudes veebilehitseja välistes keskkondades, näiteks Node.js-is.

5.3 Node.js

Juturoboti JavaScriptiks kompileeritud kood käivitatakse serveris Node.js abil. Node.js on JavaScripti käitussüsteem, mis põhineb Chrome V8 mootoril. Chrome V8 mootor kompileerib JavaScripti programmi käimise ajal masinkoodiks just siis kui vaja (ingl *just-in-time compilation*) [16]. Node.js on asünkroonse sündmujuhitava arhitektuuriga, mis on disainitud skaleeruvate serverite ehitamiseks [17].

Node.js sisaldab ka teekide haldussüsteemi (ingl *package manager*) npm, millega on võimalik vaevata lisada enda projekti vajalike teeke ning skripte rakenduse käivitamiseks või kompileerimiseks.

5.4 Bull

Bull on Redis-el põhinev ülesannete täitmise teek Node.js-ile, mis võimaldab põhirakendusest ülesandeid lisada ning täita neid erinevates protsessides, vältimaks põhirakenduse blokeerimist. Bulli kasutatakse juturoboti serveris rutiinide õigeaegseks käivitamiseks ning nende täitmiseks eraldi Node.js protsessides, mida nimetatakse töötajateks (ingl *worker*).

Ülesannete lisamisel saab määrata, millal neid täita on vaja ja kuidas neid korrata, kui nende täitmine ei õnnestu. Ülesandeid täitvaid protsesse on võimalik käivitada mitu. Vead, mis juhtuvad ülesannet täitvas protsessis, ei katkesta põhirakenduse tööd. Ülesanded salvestatakse Redis andmebaasi ning igal ajahetkel on võimalik näha kõiki parasjagu täitmisel olevaid, täitmist ootavaid või ebaõnnestunud ülesandeid.

Ülesannete täitmise teegi kasutamise vajadus tekkis sellest, et andmebaasist suure hulga andmete pärimisel võib blokeeruda põhirakenduse töö [18], mille tulemusel ei saa samal ajal Slackist tehtud kasutaja päringud piisavalt kiiresti vastust ning Slack näitab kasutajale veateadet.

5.5 Redis

Redis on mälu põhine võti-väärtus andmebaas, mida kasutatakse sõnumivahendajana, andmebaasina või puhvrina. Bull kasutab seda ülesannete salvestamiseks. Redis on ideaalne ülesannete järjekorra hoidmiseks, sest tema operatsioonid on atomaarsed — kõik toimingud andmetega sooritatakse ühe sammuna. Tänu sellele ei saa juhtuda, et erinevad ülesandeid täitvad protsessid saavad sama ülesande.

5.6 MySQL

Andmete hoiustamiseks kasutatakse MySQL-i. MySQL on avatud lähtekoodiga relatsioonilise andmebaasi haldamise süsteem [19]. Andmebaasi ülesandeks on andmete hoiustamine ja

tagastamine struktuurpäringukeele (ingl *SQL*) abil. Relatsioonilises andmebaasis on andmed salvestatud erinevatesse omavahel seostatud tabelitesse.

5.7 Python

Python on interpreteeritav dünaamiliste andmetüüpidega programmeerimiskeel. Päevapakkumiste soovitusüsteem kasutab Pythoni versiooni 3.5, mis on soovitatud EstNLTK kasutamiseks [20]. Pythoni teekide haldamiseks kasutatakse soovitusüsteemis Condat.

5.8 Conda

Conda on teekide ja käituskeskkonna haldussüsteem. Kuigi alates Python versioonist 3.4 kaasneb temaga haldussüsteem pip, siis Conda ja pip vahel on suur erinevus. Pip on võimeline installeerima ainult Pythoni teeke. Condaga on võimalik installeerida teeke või tarkvara, mis on kirjutatud suvalises keeles. Conda teegid on binaarfailid, tänu sellele pole installeerimiseks vaja midagi kompileerida. See lihtsustab EstNLTK installeerimist [21].

5.9 EstNLTK

EstNLTK ehk *Estonian Natural Language ToolKit* on soovitusüsteemis kasutusel olev Pythonis kirjutatud kogumik teeke, mis on mõeldud eestikeelsete tekstide töötlemiseks. Soovitusüsteemis kasutatakse seda lemmatiseerimiseks. EstNLTK võimaldab veel näiteks määrata sõnaliike ja teha morfoloogilist analüüsi [20], millega oleks võimalik soovitusüsteemi tunnuste määramist parandada.

5.10 Flask

Päevapakkumiste soovitusüsteemi API jaoks kasutatakse *Flask*-i. Flask on minimaalne Pythonis kirjutatud veebiraamistik.

5.11 Dokku

Kogu päevapakkumiste juturoboti lahenduse majutamiseks kasutatakse Dokkut. Dokku on lihtsasti laiendatav avatud lähtekoodiga PaaS (ingl *Platform as a Service*) mida saab paigaldada enda serverisse. Dokku teeb rakenduste serverisse üleslaadimise ja uuendamise sama lihtsaks, kui versioonihaldustarkvara *Git* käsu *push* kasutamine. Igale rakendusele loodakse (ingl *build*) *Heroku buildpack*-i või arendaja poolt antud *Dockerfile*-i järgi eraldiseisev Docker konteiner, millesse installeeritakse kõik rakenduse eeldused ning milles rakendus käivitatakse.

5.12 Docker

Docker võimaldab käivitada rakendusi isoleeritud käituskeskkonnas, mida nimetatakse konteineriks. Erinevalt virtuaalmasinatest (ingl *virtual machine*), milles on virtualiseeritud kõik alates riistvarast, on konteinerites virtualiseeritud operatsioonisüsteem. Konteinerid on selle võrra ressursikasutuses efektiivsemad.

6. Edasiarendamise võimalused

Lõputöös valminud juturobot omab kõiki vajalikke põhifunktsionaalsusi, et täita oma eesmärki. Käesolevas peatükis kirjeldatakse juturoboti ning soovitusüsteemi edasiarendamise võimalusi.

6.1 Tunnuste leidmine

Lõputöö raames valminud soovitusüsteemis valiti käsitsi märgendatud päevapakkumiste lemmade seast märksõnad, mille põhjal edaspidi soovitusüsteem vastavad tunnused tõeseks märgib. Sellise lähenemisega märksõnade otsimine on ajakulukas ja ei ole mõistlik suuremas pildis. Selle asemel võiks päevapakkumise kirjelduse järgi käsitsi ainult tunnused määrata ja seosed tunnuste ning lemmade vahel leida masinõppe abil.

Suur lisaväärtus võiks tulla päevapakkumiste kirjelduste põhjal toidu koostisosade määramisest. Selle informatsiooni olemasolul saaksid kasutajad valida, et nad ei soovi peeti, sibulat või mõnda muud toiduainet.

Lemmade leidmisel päevapakkumise kirjeldusest tuleks ka liitsõnad poolitada, et luua soovitusüsteemi jaoks seos näiteks “keedukartulite” ja “keedetud kartulite” vahel.

Lisaks võiks täiendada päevapakkumiste kirjelduse eeltöötlemise osa. Näiteks saaks luua süsteemi, kus üldpildis harva toitude kirjelduses esinevad sõnad või fraasid tulevad veebikeskkonda, kus juturoboti haldaja saab valida millised harva esinevatest sõnadest ei ole toitu kirjeldavad ning lisade need vajadusel stoppsõnade hulka. See aitaks tuvastada ja eemaldada pakkumisi, kus restoran on päevapakkumise kirjelduse asemel kirjutanud reklaami või muud informatsiooni.

6.2 Soovituste tegemine

Lõputöö raames valminud soovitusüsteemis kasutatakse soovituste tegemiseks tehisintellekti, mille käitumine on meetoodiline ning ei ole tõeliselt iseõppiv. Kasutajale omaste tunnuste asemel, või neile lisaks, võiks kasutaja eelnevad valikud koos tehtud soovitustega olla masinõppe mudeli õppimisandmeteks ning seejärel võiks mudel hinnata pakkumiste sobivust kasutajale.

Hetkel ei arvesta soovitusüsteem, mis olid soovitused kui kasutaja valis, et talle ükski pakkumine ei sobi. Sobimatute soovitude põhjal võiks langetada vastavate soovitude tunnuste hinnangut ja teha kasutajale koheselt uued kohandatud pakkumised.

Lisaks võiks olla tunnuseks valiku tegemise nädalapäev. Autori töökaaslased käivad peaaegu igal reedel ainult burgerit söömas. Sellise info põhjal saaks soovitusüsteem vähendada burgeri tähtsust muudel nädalapäevadel ning suurendada seda reedel.

Kasutajakogemuse parandamiseks võiks kaldkriipskäsklus “tahan” olla rohkem otsingumootori omadustega, kui lihtsalt pakkumise seast osasõne otsimine. Samuti parandaks otsingu tulemust suuresti nende järjestamine kauguse järgi kasutaja valikute keskmisest asukohast.

Külma stardi probleemi lahendamiseks võiks uue kasutaja puhul kasutada tema töökeskkonnas olevate teiste kasutajate valikuid, justkui need oleksid tema valikud, või soovitada populaarsemaid pakkumisi või restorane.

6.3 Erinevate suhtlusvahendite toetamine

Slacki kasutamise ainsaks põhjuseks oli see, et see on autori töökohas põhiliseks suhtlusvahendiks. Päevapakkumiste juturobotit saaks integreerida igasse juturoboteid toetavasse suhtlusvahendisse, nagu näiteks Microsoft Teams, Facebook Messenger, Telegram, Fleep ja paljud teised. Selle jaoks oleks vaja tihedalt Slackiga seotud kood ümber kirjutada ja ka andmebaasis mõningad muudatused teha.

Kokkuvõte

Valminud rakendus aitab optimeerida inimesel lõunapausi ajal söögikoha valikuks kuluvat aega, soovitades päevapakkumiste tunnuste (toidu kategooria, restorani kaugus jms) ja kasutaja eelnevate valikute põhjal kasutajale võimalikke sobivaid päevapakkumisi. Kasutajale kuvatakse igal tööpäeval kuni 10 (Tartus keskmiselt 299 päevapakkumise seast) sobivamat päevapakkumist, mis järjestatakse sobivushinnangu alusel. Sobivuse määramiseks on vajalik kasutajapoolne panus, mis seisneb igapäevaste soovitude seast sobiva pakkumise valimises. Töö autor töötas läbi ligi 7000 päevapakkumist ning nendest leitud lemmad, määrates päevapakkumiste kohta käivad tunnused ning valides välja märksõnad, mille alusel neid tunnused määratakse töös valminud soovitusüsteemis. Hetkel töötab juturobot ainult Slackis, kuid seda on võimalik ilma suurema vaevata liidestada ka teiste juturoboteid toetavate suhtlusvahenditega.

Viidatud kirjandus

- [1] Lebeuf, C., Storey, M., Zagalsky, A. Software Bots. *IEEE Software*, volume 35, 2017, IEEE, 18-23. <https://ieeexplore.ieee.org/document/8239928> (12.11.2019)
- [2] Slack, Bugsnag Documentation. <https://docs.bugsnag.com/product/integrations/slack/> (06.05.2020)
- [3] Ricci, F., Rokach, L., Shapira, B. Introduction to Recommender Systems Handbook, Springer. 2011. <http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf> (15.12.2019)
- [4] Mangalindan, JP. *Fortune*. Amazon's recommendation secret. 2012. <https://fortune.com/2012/07/30/amazons-recommendation-secret/> (14.12.2019)
- [5] How Netflix's Recommendations System Works? <https://help.netflix.com/en/node/100639> (14.12.2019)
- [6] Covington, P., Adams, J., Sargin, E. Deep Neural Networks for YouTube Recommendations. 2016. <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf> (15.12.2019)
- [7] De Gemmis, M., Iaquinta, L., Lops, P., Musto, C., Narducci, F., Semeraro, G. Preference Learning (PL-09) ECML/PKDD-09 Workshop. 2009. <http://www.ke.tu-darmstadt.de/events/PL-09/06-deGemmis.pdf> (15.12.19)
- [8] Bobadilla, J., Ortega, F., Hernando, A. Bernal, J. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, volume 26, 2012, Elsevier, 225-238. http://oa.upm.es/15302/1/INVE_MEM_2012_123432.pdf (06.05.2020)

- [9] Installing with Oauth, Slack API. <https://api.slack.com/authentication/oauth-v2> (06.05.2020)
- [10] "Eesti keele seletav sõnaraamat" 2009. <http://eki.ee/dict/ekss/index.cgi?Q=tunnus&F=M> (06.05.2020)
- [11] Uiboaed, K. Tekstikaeve terminid, Sissejuhatus tekstikaevesse, 2018.
<https://kristel.gitbooks.io/sissejuhatus-tekstikaevesse/content/tekstikaeve-terminid.html> (06.05.2020)
- [12] TypeScript - Javascript that scales. <https://www.typescriptlang.org/> (06.05.2020)
- [13] TypeScript for the New Programmer.
<https://www.typescriptlang.org/v2/docs/handbook/typescript-from-scratch.html> (06.05.2020)
- [14] Usage statistics of JavaScript as client-side programming language on websites, W3Techs World Wide Web Technology Surveys, 2020.
<https://w3techs.com/technologies/details/cp-javascript/> (06.05.2020)
- [15] JavaScript Versions, W3Schools. https://www.w3schools.com/js/js_versions.asp (06.05.2020)
- [16] Gruber, J. JIT-less V8, V8 Javascript engine development blog, 2019
<https://v8.dev/blog/jitless> (06.05.2020)
- [17] About, Node.js. <https://nodejs.org/en/about/> (06.05.2020)
- [18] Driver blocking node event loop. <https://github.com/mysqljs/mysql/issues/233> (06.05.2020)

[19] What is MySQL?, MySQL 8.0 Reference Manual

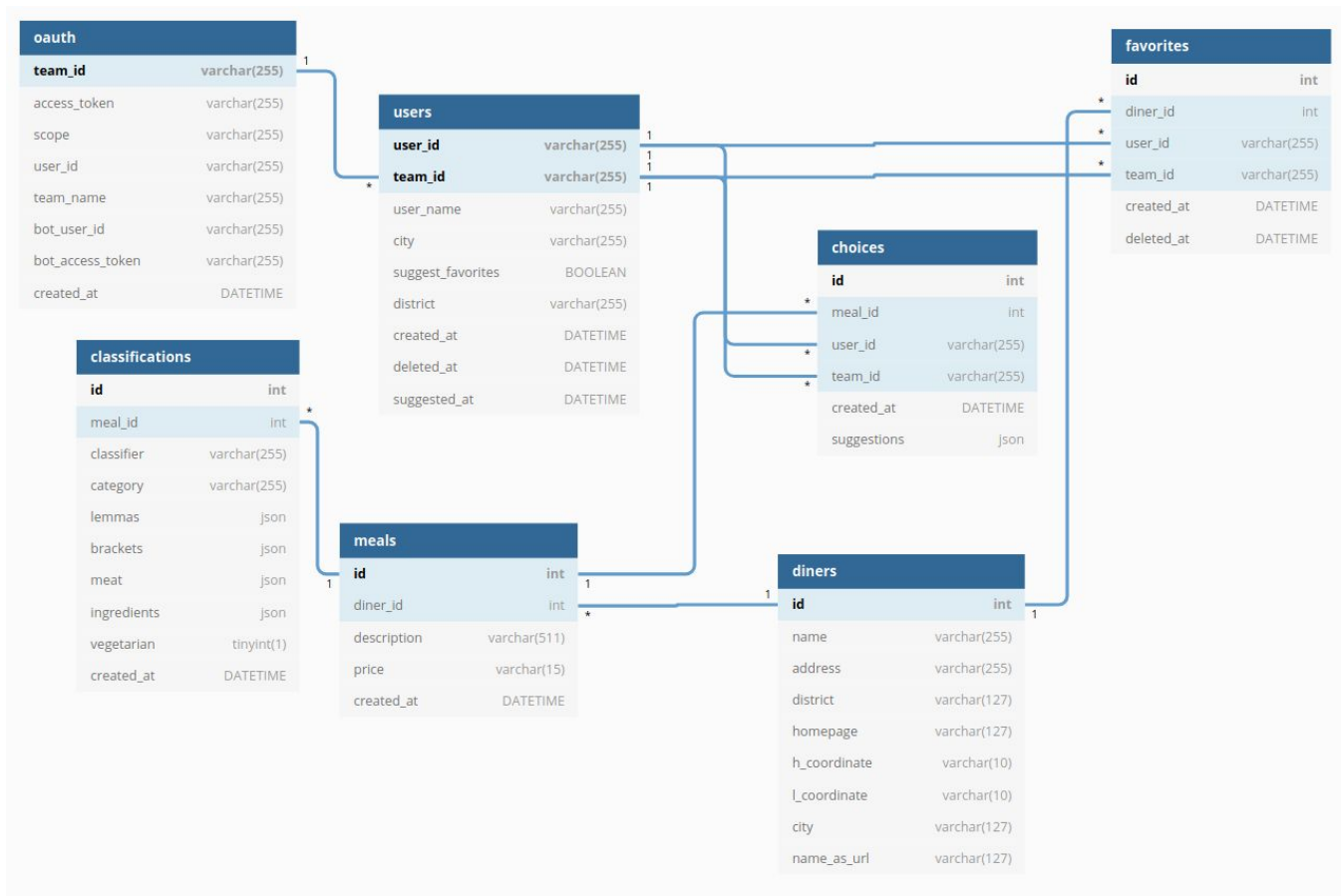
<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html> (06.05.2020)

[20] Orasmaa, S., Petmanson, T., Tkachenko, A., Laur, S., Kaalep, H.-J. EstNLTK - NLP Toolkit for Estonian, Proceedings of the Tenth International Conference on Language Resources and Evaluation, 2016. <https://estnltk.github.io/estnltk/1.4.1/> (06.05.2020)

[21] Helmus, J. Understanding Conda and Pip.

<https://www.anaconda.com/blog/understanding-conda-and-pip> (06.05.2020)

Lisad



Lisa 1. Andmebaasi tabelite veerud, nende andmetüübid ja seosed.

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Verner Läll,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Päevapakkumiste soovitusüsteem juturoboti kujul**, mille juhendajateks on Laura Ruusmann ja Mark Fišel, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Tartus, **08.05.2020**