

Vilniaus universitetas

Matematikos ir informatikos fakultetas

Informatikos studijų programa

**Penktasis laboratorinis darbas - Maisto Užsakymo Sistemos Dokumentacija**

Ataskaitą parengė: Vasarė Pratuzaitė ir Dovilė Vaiginytė 3 kurso, 2 grupės informatikos studentės

Vilnius, 2024

Turinys

[Dokumentacijos aprašymas 3](#_Toc185103788)

[Sistemos API 4](#_Toc185103789)

[Duomenų modelio diagrama 6](#_Toc185103790)

## Dokumentacijos aprašymas

Apžvelgus keturias duomenų bazes, padarėme išvadą, kad maisto užsakymo sistemai įgyvendinti geriausias būdas bus pasinaudoti Redis ir Mongo duomenų bazėmis. Ši sistema sėkmingai įgyvendinama, kai pagrindinė struktūrizuotų duomenų bazė yra Mongo, o duomenų talpyklai pasirenkame Redis duomenų bazę. Redžio pasirinkimas mums teikė didelę naudą, dėl itin didelio greičio ir paprasto raktų saugojimo modelio. Taip pat duomenų bazė leidžia išsaugoti dažnai naudojamus duomenis, tokius kaip rekomendacijos ar paieškos rezultatus, ir juos greitai pasiekti, sumažinant pagrindinės duomenų bazės apkrovą. Redžio vienas iš privalumų, kad palaiko įvairiausių duomenų struktūras, tokias kaip: sąrašai ir rinkiniai, kurie būtų surūšiuoti arba nebūtinai, todėl tokios struktūros leidžia efektyviai įgyvendinti reitingavimą, paieškos užklausų kešavimą (laikinai saugant duomenis greitoje atmintyje, kad paskui juos būtų galima pasiekti greičiau). Taip pat, Redžio teikiama nauda yra didelė, nes vienu metu prie sistemos gali būti prisijungę daugiau vartotojų, nes ši duomenų bazė toleruoja didelį sistemos apkrovimą. Šios programos tikslas: leisti registruoti restoranus, klientus, patiekalus, kurti užsakymus, pateikti rekomendacijas bei atlikti paiešką. Sistema buvo sukurta moduliniu principu, kad būtų lengva praplėsti funkcionalumą: restoranų ir meniu valdymas leidžia registruoti restoranus, jų meniu elementus, bei valdyti informaciją apie patiekalus, užsakymų valdymas, kurio tikslas, kad klientas galėtų registruotis, pateikti užsakymus tiek išsinešimui, tiek pristatymui į kitą vietą, rekomendacijų sistemos plėtra ir paieška, kuri leidžia atlikti patiekalų ir restoranų paiešką.

MongoDB galima įdiegti replikų rinkinius, siekiant užtikrinti aukštą prieinamumą. Pagrindinė replikos kopija tvarko rašymo užklausas, o antroji – skaitymo. Tuo tarpu Redis turi galimybę įgalinti „Redis Sentinel“ ar „Cluster“, kurie gali užtikrinti duomenų pasiekiamumą. Skaidymas MongoDB gali restorano ir kliento duomenų bazes padalyti duomenis, naudojant „restaurant\_id“ arba „customer\_id“ kaip skaidymo raktus. Na, o Redis, panaudojus klasterius, gali paskirstyti talpyklos apkrovą tarp mazgų.

Kalbant apie indeksus, jų pasirinkimas svarbus, norint užtikrinti efektyvų duomenų pasiekimą: pilno teksto paiešką bei greitesnį užklausų panaudojimą. Tai yra svarbu, nes naudojamas tekstinis indeksas „name“, „menu.name“ ir „menu.description“ laukams, kurie leidžia atlikti greitą paiešką pagal patiekalų ir restoranų pavadinimus ar net aprašymus, o indeksai laukuose „menu\_id“ ir „orders.items.menu\_item\_id“ gali paspartinti užklausas, skirtas užklausų informacijai ar patiekalams gauti. Prasmingų pirminių raktų pasirinkimas irgi turi didelę įtaką. Nors ši sistema naudoja automatiškai sugeneruotus pirminius raktus iš „counters“ kolekcijos, tai užtikrina prasmingą panaudojimą, unikalumą ir paprastą valdymą. „\_id“ panaudojamas MongoDB dokumentų identifikavimui, „restaurant\_id“ yra sugeneruojamas, menu\_id leidžia atskirai identifikuoti kiekvieną patiekalą, na o, „customer\_id“ bei „order\_id“ garantuoja kiekvieno dokumento unikalumą ir lengvą skaidymą.

Apibendrinant galima pasakyti, kad ši sistema užtikrina aukštą pasiekiamumą, efektyvumą ir lankstumą. Naudodama MongoDB ir Redis, nes ji palaiko didelius duomenų srautus, o skaidymo ir replikavimo strategijos užtikrina, kad ji būtų lengvai pritaikoma prie augančių poreikių. Pridėjus naujas funkcijas, tokias kaip lokalizuota paieška ar realaus laiko užsakymų stebėjimas, tokiu būdu projektas turėtų didesnias plėtros galimybes.

## Sistemos API

Paveikslėlio įkėlimas:

/upload-image [POST]

400 - No file part in the request

400- No selected file

400- Invalid file type. Only .jpg, /jpeg, ad .png files are allowed

201- Image uploaded successfully!

Restoranai:

/restaurants [PUT]

400 - Invalid input. Mandatory attributes are missing

201 - Restaurant registered successfully!

Restoranai:

/restaurants [GET]

404 - No restaurants were found

200 – without message

Restoranai pagal ID:

/restaurants/<restaurantId> [DELETE]

204 - Restaurant deleted

404 - Restaurant not found

Restoranų meniu pagal restorano ID:

/restaurants/<restaurantId>/menu [PUT]

400 - Invalid input, mising name, description, or price

400 - Price must be a positive number

201 - Menu item added successfully!

Gaunam restoranų meniu pagal restorano ID:

/restaurants/<restaurantId>/menu [GET]

404 - Restaurant not found

404 - Restaurant not found

200 – Without message

Statinis paveikslio pateikimas:

/uploads/<filename> [GET]

Klientai:

/customers [PUT]

400 - Invalid input. Mandatory attributes are missing

200 – Without message

Klientai:

/customers [GET]

400 - No customers were found

200 – Without message

Klientas pagal ID:

/customers/<customerId> [DELETE]

204 - Customer deleted

404 - Customer not found

Užsakymai pagal ID:

/orders/<customerId> [PUT]

400 - Invalid input. 'items' is required and must be a non-empty list

400 - Invalid input. 'order\_type' ir required

400 - Invalid 'order\_type. Must be one of {valid\_order\_types}

400 - Invalid input. 'address' is required for 'delivery' orders.

400 - Each item must include 'restaurant\_id' and 'menu\_item\_id'

400 - Each item must include valid 'quantity' greaten than 0

404 - Restaurant with ID {item['restaurant\_id']} not found

400 - Menu item ID {item['menu\_item\_id']} does not exist in restaurant {item['restaurant\_id']

404 - Customer not found

201 - Order created successfully!

Užsakymas pagal kliento ID:

/customers/<customerId>/orders [GET]

404 - Customer not found

404 - No orders were found for this customer

200 – Without message

Rekomendacijos:

/recomendations/<customerId> [GET]

200 – Without message

404 - Customer not found

Paieška:

/search [GET]

400 - Query parameter ir required

200 – Without message

Išvalymas:

/cleanup [POST]

200 - Cleanup completed

500 - An error occurred while clearing the database

## Duomenų modelio diagrama

A diagram of a restaurant

Description automatically generated

pav