**Tēriņu apkopojuma izveide(1. attēls)**

Šajā diagrammā attēlots finanšu izsekošanas programmas darbības process, kur lietotājs pieslēdzas savam kontam un saņem iespēju apskatīt pēdējo septiņu dienu tēriņu apkopojumu. Kad lietotājs apstiprina šo darbību, sistēmas loģikas daļa pieprasa no datu bāzes pēdējo 7 dienu tēriņu informāciju.

Datu bāze atgriež pieprasītos datus, ietverot tēriņus un katrai grupai atvēlēto budžetu. Tālāk loģikas sistēma aprēķina tēriņus katras grupas ietvaros, salīdzinot tos ar budžeta ierobežojumiem. Ar paplašinājuma "Seaborn" palīdzību tiek ģenerēta vizualizācija, kas atspoguļo tēriņu sadalījumu un savstarpējo salīdzinājumu.

Oriģinālais plāns datu vizualizācijai bija izmantot paplašinājumu “matplotlib”, taču pēc dziļākas alternatīvu izpētes nonācām pie secinājuma, ka paplašinājums “Seaborn” būtu piemērotāks situācijai. Šāds secinājums tika veikts pamatojoties uz faktu, ka tas ir stilistiski pievilcīgāks, kā arī veidotājiem draudzīgāks nekā tā alternatīva “matplotlib”, vizualizāciju konstruēšanai pieprasot mazāku koda apjomu. Citas alternatīvas kā “plotly” netika izvēlēti, lai arī radītās diagrammas ir interaktīvākas, tas ir sarežģītāks un vairākas funkcijas ir par maksu, kas liedz pieeju unikālām funkcijām kas šo paplašinājumu izceļ uz citu fona.

Pamatojoties uz aprēķiniem, sistēma sniedz ieteikumus par budžeta izmaiņām un izceļ grupas, kurās tēriņi ir lielāki vai mazāki nekā paredzēts. Lietotājs saņem šo vizualizāciju un var pieņemt informētus lēmumus par savu finanšu pārvaldību.

**Jaunas grupas izveide(2. attēls)**

Šī diagramma ilustrē finanšu izsekošanas programmas loģiku jaunās izdevumu grupas izveidei. Process sākas ar to, ka lietotājs pieprasa izveidot jaunu izdevumu grupu ar konkrētu nosaukumu, piemēram, "X". Sistēma atbild ar pieprasījumu ievadīt grupas nosaukumu, un lietotājs to ievada.

Kad nosaukums ir ievadīts, loģikas sistēma nosūta pieprasījumu datu bāzei, lai pārbaudītu, vai grupas nosaukums "X" jau eksistē. Datu bāze atbild, un loģikas sistēma analizē šo atbildi, lai noteiktu turpmākās darbības :

Ja datu bāze neatklāj jau esošu grupu ar šo nosaukumu, sistēma izveido jauno grupu un saglabā to datu bāzē. Pēc veiksmīgas saglabāšanas datu bāze apstiprina darbību, un lietotājs saņem ziņojumu, ka grupa ar nosaukumu "X" ir veiksmīgi izveidota.

Taču, ja grupa ar nosaukumu “X” tiek konstatēta, sistēma atgriež kļūdas ziņojumu, informējot lietotāju, ka šāda grupa jau ir reģistrēta un jaunu grupu ar to pašu nosaukumu izveidot nav iespējams. Šāda validācija novērš dublēšanos un palīdz uzturēt kārtību lietotāja finanšu uzskaitē.

Šis process nodrošina efektīvu un organizētu pieeju finanšu grupu pārvaldībai, ļaujot lietotājam viegli sakārtot izdevumus pēc kategorijām, vienlaikus novēršot dublētus ierakstus, kas varētu radīt neskaidrības budžeta pārskatā.

**Budžeta ievade/rediģēšana(3. attēls)**

Šī diagramma attēlo budžeta ievades un rediģēšanas procesu. Process sākas ar lietotāja pieprasījumu ievadīt vai rediģēt kopējā budžeta summu. Sistēma atbild, pieprasot lietotājam ievadīt vēlamo budžeta apjomu.

Pēc budžeta summas ievades sistēma lūdz lietotājam norādīt budžeta sadalījumu starp dažādām grupām, ievērojot decimālskaitļu formātu (piemēram, 0.2 jeb 20%, 0.6 jeb 60% utt.). Lietotājs sniedz vēlamo sadalījumu, ko sistēma tālāk apstrādā un saglabā.

Kad sistēma saņem budžeta sadalījuma datus, tā tos saglabā datu bāzē, pievienojot katrai grupai attiecīgo budžeta daļu. Pēc veiksmīgas saglabāšanas datu bāze nosūta apstiprinājumu, un lietotājs saņem paziņojumu, ka budžeta iedalījums ir veiksmīgi saglabāts.

Šis process lietotājam nodrošina efektīvu budžeta sadalījumu starp dažādām izdevumu grupām, izmantojot lietotājam draudzīgu un pārskatāmu pieeju budžeta pārvaldībai. Ja lietotājs nākotnē vēlēsies apskatīt savu budžetu, sistēma, pēc pieprasījuma, izvadīs kopējo budžetu kā arī atlikuma bilanci.

**Tēriņu ievade un saglabāšana (4.attēls)**

Šajā diagrammā ir attēlots tēriņu ievades process. Šī funkcija tiek lietotājam piedāvāta tikai tad, ja lietotājs ir reģistrējies, lai datubāzei būtu iespējams saglabāt datus. Kad lietotājs izvēlas ievadīt jaunus tēriņus, datubāze saņem pieprasījumu nosūtīt jau saglabātās tēriņu kategorijas, kuras tiek piedāvātas lietotājam ar drop-down list. Ja neviena no kategorijām neatbilst nepieciešamajai, tad lietotājs izveido jaunu kategoriju (skat. Jaunas grupas izveide) un tikai pēc tam izvēlas ievadīt jaunus tēriņus. Lietotājs izvēlas kategoriju un ievada summu, un šī informācija, kā arī pašreizējais datums, tiek nosūtīti datubāzei un saglabāta pie konkrētā lietotāja. Pēc saglabāšanas loģika pieprasa datubāzei pēdējos reģistrētos datus, lai pārbaudītu, vai saglabāšana ir veiksmīgi. Ja pēdējie reģistrētie dati sakrīt ar tikko ievadītajiem, tad loģika nosūta apstiprinājumu lietotājam.

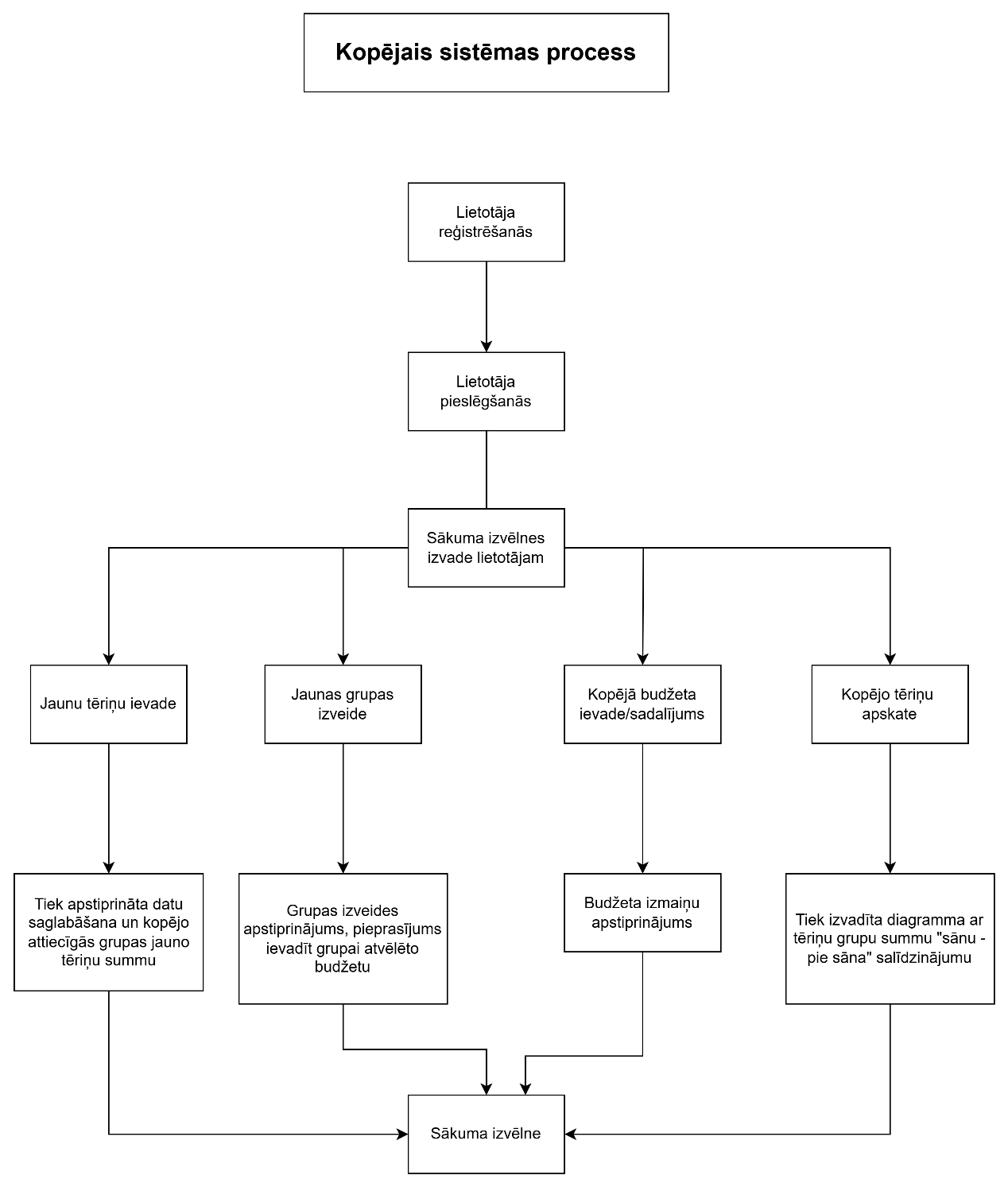
**Lietotāja reģistrācija (5.attēls)**

Šajā diagrammā tiek attēlota lietotāja reģistrācija, kurā loģika pieprasa lietotājam ievadīt lietotājvārdu, paroli, plānoto budžetu un izvēlēties laika posmu, pēc kura būs iespējams saņemt tēriņu pārskatu. Visi šie dati tiek nosūtīti datubāzei, kura tos saglabā. Šo procesu ir iespējams papildināt ar reCaptcha vai e-pasta verifikāciju, lai programmā reģistrētos tikai reāli lietotāji.

**Diagrammu izveide (6.attēls)**

Lietotājam reģistrējoties tiek piedāvāta opcija saņemt tēriņu pārskatu reizi nedēļā/mēnesī/gadā. Kad lietotājs atver programmu, tiek piedāvāta iespēja apskatīt datus par iztērēto daudzumu dažādās kategorijās. Ja lietotājs izvēlas šo opciju, loģika nolasa pašreizējo datumu un datubāzei tiek nosūtīts pieprasījums pēc datiem no pēdējās nedēļas/mēneša/gada atkarībā no lietotāja izvēles, no kuriem loģika izveido diagrammas. Lietotājam tiek piedāvātas diagrammas, kurās tiek attēlots cik ir iztērēts katrā kategorijā, tēriņu salīdzinājums pa dienām, kā arī tēriņu daudzuma salīdzinājums ir iepriekš izvēlēto budžetu. Dati par kategorijām tiktu attēloti stabiņu diagrammā, bet tēriņu salīdzinājumu pa dienām būtu ērtāk attēlot līniju diagrammā.

**Sistēmas apkopojuma diagramma**

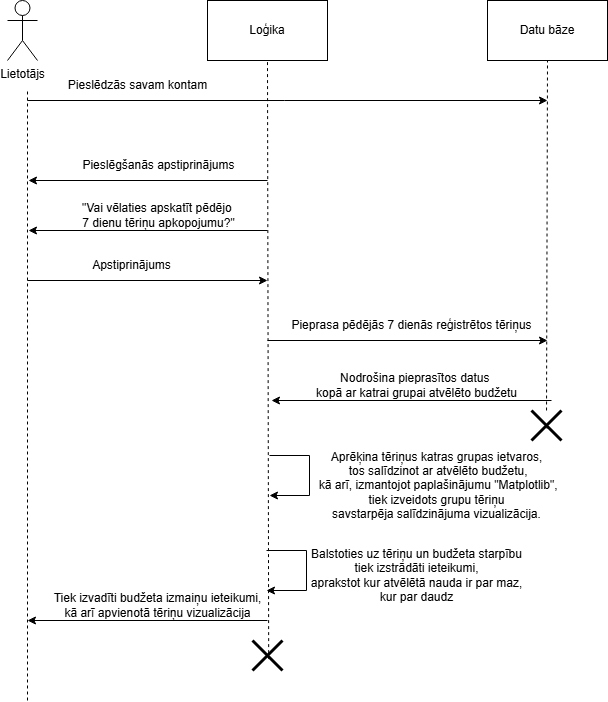
****

**Paplašinājumu izvērtējums**

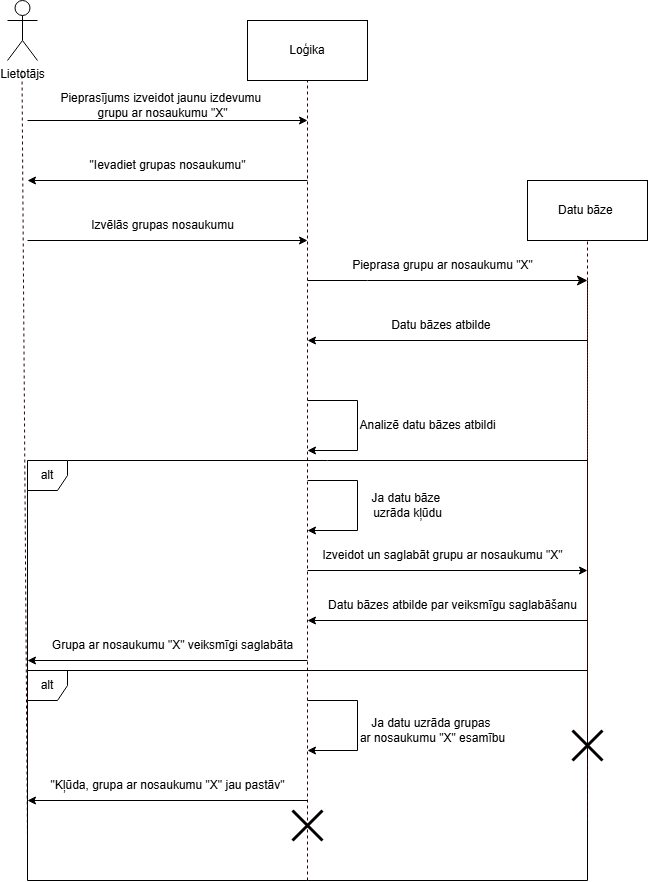
Izvērtējot pieejamās datubāzes, secinājām, ka “PostgreSQL” bibliotēka vislabāk sader ar projekta funkcijām un mērķiem, piedāvājot sarežģītu datu apstrādes funkcijas, tajā pašā laikā nodrošinot drošas datu transakcijas starp lietotāju un datu bāzi. Alternatīvi kā “MySQL” un “SQLite” mums neder tā iemesla dēļ, ka “SQLite” datubāze strādā kā fails uz datora, kas nav optimāls formāts ja ir nepieciešama vairāku lietotāju datu saglabāšana un attālināta piekļuve tiem. “MySQL” nav spējīgs apstrādāt datus tādā līmenī un apjomā, kādu pieprasa attiecīgais projekts, naturāli padarot “PostgreSQL” kā acīmredzmo izvēli.

Apskatot ar “Python” saderīgos ietvarus lietotņu izstrādei, mēs izvirzījām 3 galvenos pretendentus, kuri ir “Kivy”, “Tkinter” un “PyQT”. Izvērtējot visu ietvaru piedāvātās funkcijas, nonācām pie secinājuma, ka ietvars “Kivy” vislabāk saderēs ar mūsu programmu. Lai gan “PyQT” piedāvā profesionālas kvalitātes saskarnes, tas nāk ar daudz apjomīgāku mācīšanās līkni. “Tkinter” ir vienkāršs un viegli apgūstams, taču tā piedāvātais interfeiss arī ir attiecīgi primitīvs. Ietvars “Kivy” ir kā viduspunkts starp abiem, relatīvi vienkārša izstrādes procesa rezultātā piedāvājot stilistiski pievilcīgas saskarnes.

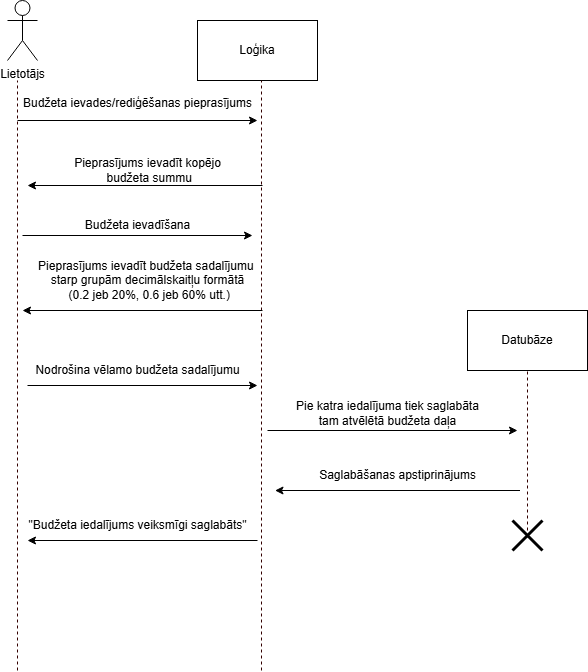
**Attēli**

****

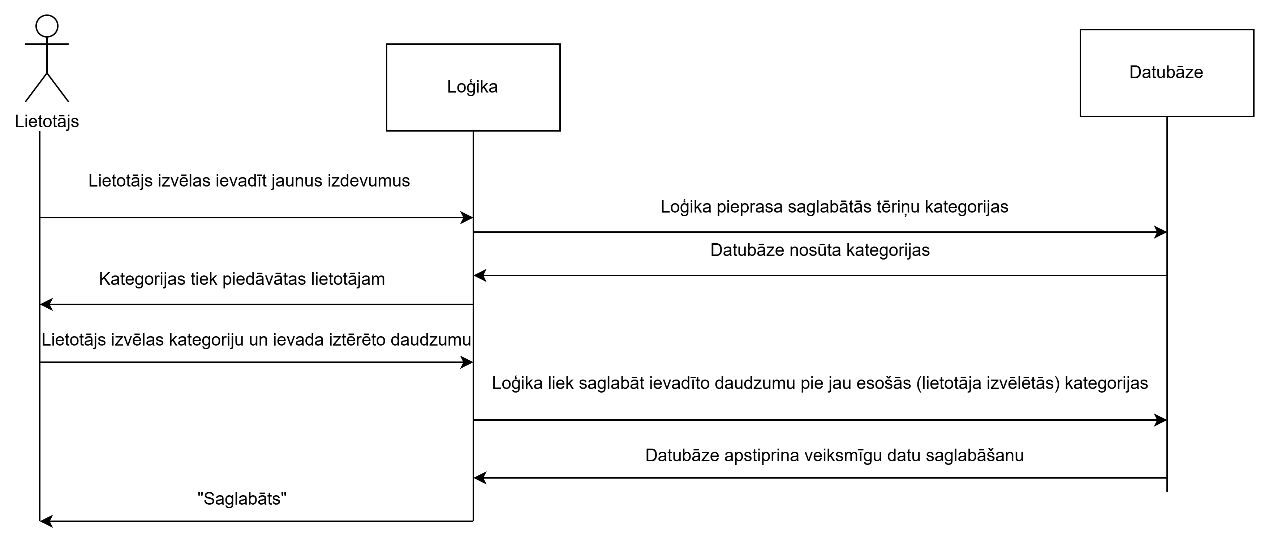
*1. attēls, tēriņu apkopojums*



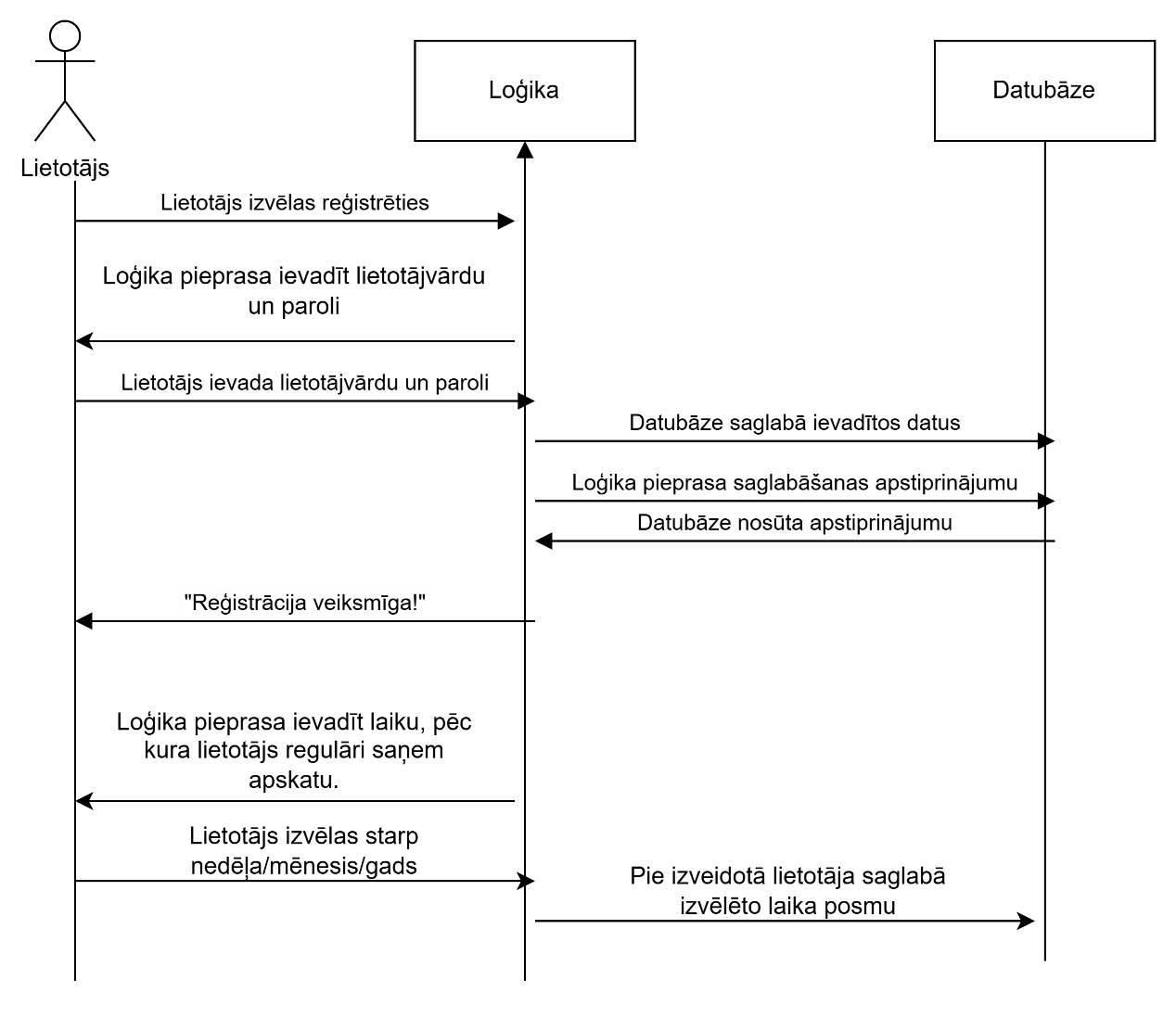
*2. attēls, jaunas grupas izveide*

**

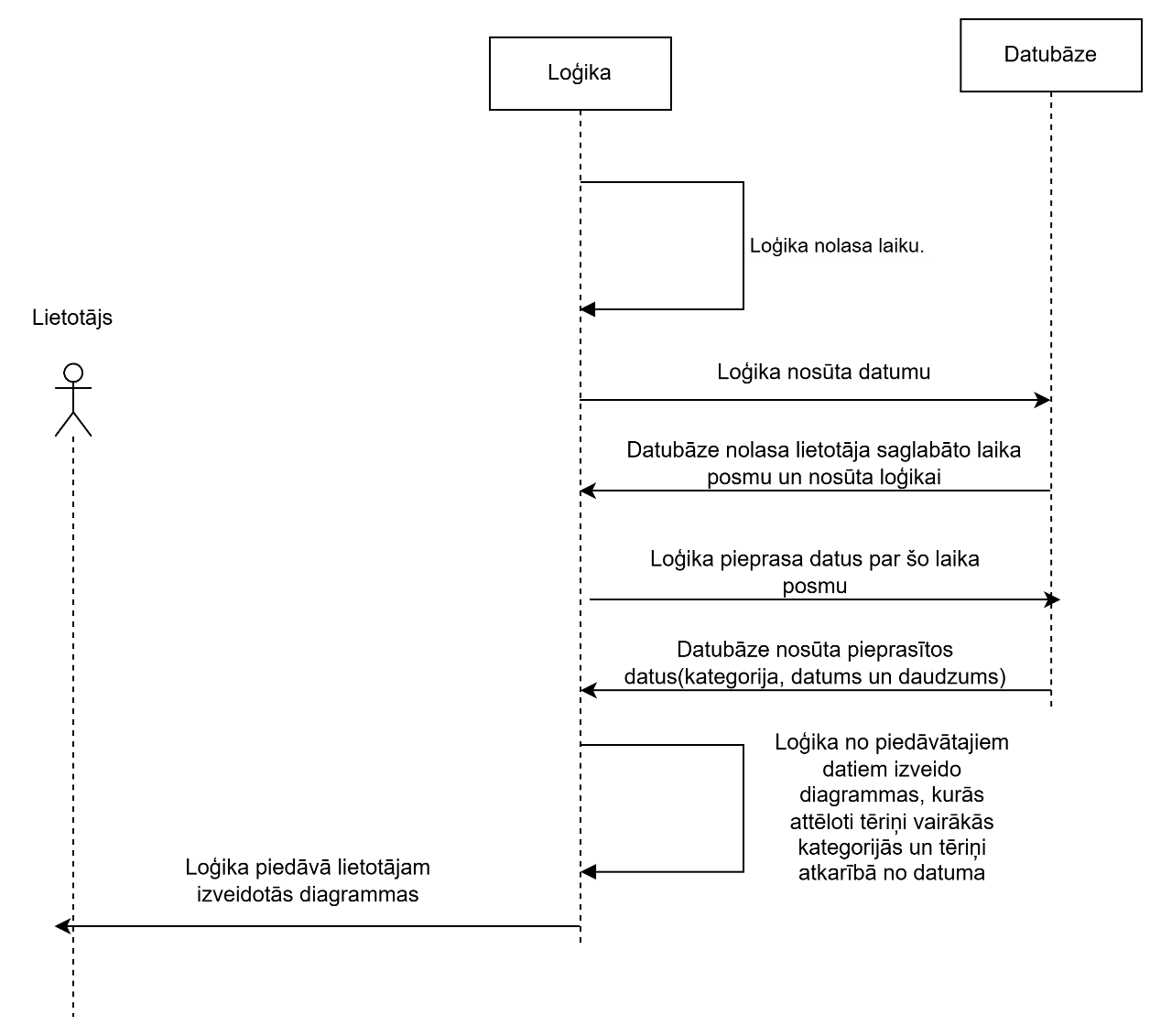
*3. attēls, budžeta ievade un sadalījums*

**

*4. attēls, tēriņu ievade un saglabāšana*



*5.attēls, lietotāja reģistrācijas process*



*6.attēls, tēriņu apkopojuma izveide diagrammu formā*