



Liepājas Valsts tehnikums

Nekustamā īpašuma nomas sistēma "EstateRent"

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Profesionālā kvalifikācija

Programmēšanas tehniks

Grupas nosaukums

4 PT

Projekta izstrādātājs

/vārds, uzvārds, paraksts/

Eksāmena datums 2024. gada 20.jūnijs

Liepāja 2024

Satura rādītājs

Ievads	Error! Bookmark not defined.
1. Uzdevuma formulējums	4
2. Programmatūras prasību specifikācija	5
2.1. Produkta perspektīva	5
2.2. Sistēmas funkcionālās prasības.....	5
2.3. Sistēmas nefunkcionālās prasības	15
2.4. Gala lietotāja raksturiezīmes.....	16
3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums	17
3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts	17
3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts	18
4. Sistēmas modelēšana un projektēšana	21
4.1. Sistēmas struktūras modelis.....	22
4.2. Klašu diagramma / ER diagramma (Obligāti, ja sistēmā ir klases un/vai tabulas)	27
4.3. Funkcionālais un dinamiskais sistēmas modelis..	Error! Bookmark not defined.
4.4. Aktivitāšu diagramma (Activity)	28
4.5. Lietojumgadījumu diagramma (Use Case).....	29
4.6. Sistēmas moduļu apraksts un algoritmu shēmas..	Error! Bookmark not defined.
5. Lietotāju ceļvedis.....	32
6. Testēšanas dokumentācija	45
6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums.....	45
6.2. Testpiemēru kopa	46
6.3. Testēšanas žurnāl	55
7. Secinājumi	58
8. Lietoto terminu un saīsinājumu skaidrojumi.....	59
9. Literatūras un informācijas avotu saraksts	60
PIELIKUMS.....	61

Ievads

Šodienas mūsdienu dzīves ritmā kvalitatīvai un ērtai dzīvesvietai ir kļuvusi pieejamāka un pievilcīgāka nekā jebkad agrāk, pateicoties nekustamā īpašuma nozares tehnoloģiskajai progresijai un inovācijām. Šis dokumentācijas materiāls sniedz plašu ieskatu un aicinājumu izpētīt un izmantot mūsu piedāvāto dzīvokļu un māju īres programmu, kas atver jaunas unikālas iespējas atrast un īrēt ideālu mājokli, atbilstoši jūsu vēlmēm un prasībām.

Mūsu platforma piedāvā ērtu un intuitīvu veidu izpētīt dažādus nekustamos īpašumus, sākot no mājīgiem dzīvokļiem pilsētas centrā līdz ekskluzīviem lauku mājokļiem ar skatu uz ainavu. Šī dokumentācija ir padziļināts materiāls par mūsu nekustamā īpašuma nomas programmu, kas izstrādāta, ņemot vērā mūsdienu prasības un inovatīvas pieejas nekustamā īpašuma jomā.

Turpinot lasīt šo dokumentu, jūs iegūsiet ne tikai pilnu aprakstu par programmas funkcionalitātēm, bet arī detalizētas diagrammas, kas vizualizē tās struktūru un komponentu mijiedarbību. Turklāt jūs atradīsiet pārskatu par testēšanas procesiem, kas garantē augstu uzticamības un drošības līmeni, lietojot mūsu programmu.

Būtiska šīs dokumentācijas daļa ir programmas lietotāju ceļvedis, kas izstrādāts, lai palīdzētu jums izprast katru funkcionalitāti un maksimāli izmantot visas nekustamā īpašuma izīrēšanas programmas priekšrocības. Neatkarīgi no jūsu pieredzes nekustamā īpašuma vai tehnoloģiju jomā, šajā dokumentā ir sniegti galvenie rīki un norādījumi, lai veiksmīgi lietotu mūsu programmu.

Lūdzu, uzmanīgi izlasiet šo dokumentu, lai pilnībā izprastu piedāvātās nekustamā īpašuma nomas programmatūras funkcijas, īpašības un priekšrocības, kas varēs ievērojami uzlabot jūsu dzīves kvalitāti un atvieglot meklējot mājokli.

1. Uzdevuma formulējums

Uzdevuma mērķis ir radīt visaptverošu mājas īres platformu, kuras galvenais uzdevums ir nodrošināt lietotājiem vienkāršu un efektīvu veidu, kā atrast un īrēt īres objektus. Platformas funkcionalitāte ietver lietotāja reģistrāciju, kas var notikt ar e-pasta adresi un paroli. Lai lietotājiem būtu viegli atrast vajadzīgo, ir izstrādāta detalizēta meklēšanas un filtrēšanas sistēma. Lietotāji var ievadīt dažādus kritērijus, piemēram, atrašanās vietu, cenu dienā, istabu skaitu un citus, lai atlasītu vislabāk atbilstošos īres objektus. Platforma piedāvā iespēju īpašniekiem pievienot sludinājumus ar detalizētu informāciju, attēliem un norādītu īres noteikumus. Lietotāji var viegli veikt rezervācijas, izvēloties pieejamos datumus un ievadot apmaksas datus. Pēc katras īres pieredzes ir iespējams atstāt atsauksmes un novērtējumus, tādējādi veidojot uzticamības sistēmu starp lietotājiem. Taču mūsu programma nepiedāvā tikai informāciju par dzīvokļiem, bet arī nodrošina tiešu mijiedarbību ar nekustamā īpašuma īpašniekiem. Viegli sazinieties ar īpašniekiem, uzdodiet jautājumus un apspriežiet īres nosacījumus. Mēs cenšamies radīt ērtu un drošu vidi abām pusēm, palīdzot orientēties īres procesā un noslēgt izdevīgu vienošanos. Turklāt mūsu platforma piedāvā arī iespēju īrēt savu pašu nekustamo īpašumu, pēc administrācijas apstiprinājuma. Tas nodrošina, ka jūsu īres pieredze ir pārredzama un droša ne tikai īrētājiem, bet arī īpašniekiem. Ļaujiet mūsu programmai kļūt par jūsu uzticamu partneri, meklējot un īrējot dzīvokļus, nodrošinot jums vienkāršu piekļuvi daudziem nekustamā īpašuma piedāvājumiem, kas kļūs par jūsu mājīgu patvērumu. Mēs esam pārliecināti, ka jūsu dzīvokļa vajadzības tiks apmierinātas ar mūsu palīdzību, un jūsu dzīvojamais piedzīvojums kļūs par patīkamu un apmierinošu pieredzi. Laipni lūgti pasaulē, kur jums ir pieejams dzīvoklis pēc jūsu gaumes un vēlmēm, un kur jūs varat piedāvāt savu nekustamo īpašumu īrei pēc administrācijas apstiprinājuma. Lai padarītu platformu pilnībā darboties, ir ieviesta droša maksājumu integrācija, kas ļauj lietotājiem veikt ērtas un drošas transakcijas tiešsaistē. Personīgais profils lietotājiem ļauj viegli pārvaldīt savu informāciju, rediģēt rezervācijas un sekot notikumu vēsturei. Paziņojumu sistēma nodrošina, ka lietotāji tiek informēti par svarīgiem notikumiem, piemēram, rezervāciju apstiprinājumiem un jaunajiem piedāvājumiem. Platformas pilnveidošanai varētu ieviest vairākas inovatīvas funkcijas, kas uzlabotu lietotāju pieredzi un padarītu šo mājas īres platformu vēl pievilcīgāku. Piemēram, integrējot tehnoloģijas, kas atpazīst lietotāju priekšmetus un automātiski aizpilda sludinājuma informāciju, īpašniekiem būtu vienkāršāk un ātrāk pievienot piedāvājumus. Turklāt var apsvērt ekoloģiski draudzīgas īres objektu kategoriju, kas uzsvērtu vides ilgtspējību un piesaistītu lietotājus ar interesi par ekoloģiski atbildīgām izvēlēm.

2. Programmatūras prasību specifikācija

Šajā modulī tiks aprakstītas programmas funkcijas, kā arī tās perspektīva

2.1. Produkta perspektīva

Programmu var uzlabot, izmantojot jaunākas tehnoloģijas, un laika gaitā var izveidot ērtāku un lietotājam draudzīgāku saskarni. Ieviesiet analītikas sistēmu, lai izsekotu lietotāju uzvedībai un dažādu funkciju efektivitātei. Apkopojiet atsauksmes no lietotājiem un aktīvi izmantojiet tās, lai turpinātu uzlabot produktu.

2.2. Sistēmas funkcionālās prasības

1. Lietotāju reģistrācija:

Lietotājiem ir jāievada e-pastu, vārdu, uzvārdu un paroli reģistrācijas procesā, nodrošinot drošu un viegli saprotamu pieredzi.

Lietotājiem ir iespēja pēc reģistrācijas pilnībā rediģēt savu profilu, pievienojot vai atjauninot informāciju par sevi un attēlu.

- Ievaddati: Lietotājiem jāievada e-pasts, vārds, uzvārds un parole reģistrācijas procesā.
- Apstrāde: Pārbauda un validē ievadītos datus, pēc tam reģistrē lietotāju datu bāzē.
- Izvaddati: Paziņo par veiksmīgu reģistrāciju vai parāda kļūdu paziņojumu, ja reģistrācija neizdodas.

2. Lietotāju autentifikācija:

autentifikācija:

Lietotājiem ir jāiraksta savus e-pasta un paroles datus autentifikācijas procesā lai varetu izmantot visus tīmekļa vietnes iespējas

- Ievaddati: Lietotājiem ir jāievada savs e-pasts un parole autentifikācijas procesā.
- Apstrāde: Pārbauda, vai ievadītie dati atbilst esošajiem reģistrētajiem datiem un nodrošina piekļuvi sistēmai.
- Izvaddati: Pāradresē lietotāju uz sākulapu vai sniedz piekļuvi kontam pēc veiksmīgas autentifikācijas

3. Rezervāciju pārvaldība:

Kalendāra integrācija:

Īpašniekiem ir iespēja viegli apskatīt rezervāciju kalendāru un pārvaldīt īpašuma pieejamību.

Sistēma automātiski atspoguļo rezervācijas un pieejamību, integrējoties ar ārējiem kalendāriem.

Rezervāciju apstiprināšana un noraidīšana:

Īpašniekiem ir tiesības apstiprināt vai noraidīt rezervācijas pieprasījumus, sniedzot elastību īpašumu pārvaldībā.

- Ievaddati: Lietotājiem jānorāda rezervācijas datumus un citi svarīgi rezervācijas parametri.
- Apstrāde: Pārbauda rezervācijas pieejamību un veic rezervāciju apstiprināšanu vai noraidīšanu.
- Izvaddati: Paziņo par veiksmīgu rezervācijas veikšanu vai parāda kļūdas paziņojumu, ja rezervācija nav veiksmīga.

4. Maksājumu pārvaldība:

Droša maksājumu apstrāde:

Sistēma nodrošina drošu un šifrētu maksājumu apstrādi, lai aizsargātu lietotāju finanšu informāciju.

Var iekļaut integrāciju ar populāriem maksājumu pakalpojumiem.

Maksājumu vēsture un statuss:

Lietotājiem ir piekļuve pārskatam par veiktajiem maksājumiem, to vēsturi un pašreizējo maksājumu statusu.

- Ievad dati: Lietotājiem jāievada maksājuma informācija, tostarp kredītkartes detaļas.
- Apstrāde: Veic drošu maksājumu apstrādi, izmantojot šifrēšanas tehnoloģijas un integrāciju ar maksājumu pakalpojumiem.
- Izvaddati: Paziņo par veiksmīgu maksājuma veikšanu vai parāda kļūdas paziņojumu, ja maksājums nav veikts veiksmīgi.

5. Paziņojumu un atgādinājumu sistēma:

Pielāgoti paziņojumi:

Lietotājiem ir iespēja pielāgot paziņojumus, lai saņemtu informāciju par rezervācijām, atsauksmēm un citiem svarīgiem notikumiem.

Atgādinājumi var tikt nosūtīti caur vairākiem komunikācijas kanāliem, tostarp e-pastu un mobilajiem paziņojumiem.

- Ievad dati: Lietotājiem ir iespēja pielāgot paziņojumus, lai saņemtu informāciju par rezervācijām, atsauksmēm un citiem svarīgiem notikumiem.

- Apstrāde: Pārbauda lietotāja izvēlētos paziņojumus un sagatavo un nosūta atbilstošos paziņojumus.
- Izvaddati: Nosūta pielāgotus paziņojumus, lai informētu lietotājus par svarīgiem notikumiem vai darbībām.

6. Meklēšana un filtrēšana:

Detalizēti meklēšanas filtri:

Lietotājiem ir pieejams plašs klāsts filtru, lai ērti atrastu piemērotus nekustamos īpašumus.

Var piedāvāt īpašus filtra kritērijus, piemēram, mājdzīvnieku draudzīgumu vai īpašumu veidu.

Kartes meklēšana:

Iespēja meklēt īpašumus, izmantojot interaktīvu karti, kas lietotājiem sniedz vizuālu informāciju par īpašumu atrašanās vietu.

- Ievad dati: Lietotājiem ir pieejams plašs klāsts filtru, lai atrastu piemērotus nekustamos īpašumus.
- Apstrāde: Analizē lietotāja izvēlētos filtrus un izvēlas atbilstošos nekustamos īpašumus.
- Izvaddati: Parāda rezultātus, kas atbilst lietotāja meklēšanas kritērijiem.

7. Atsauksmju un vērtējumu sistēma:

Atsauksmju publicēšana un pārvaldība:

Īrniekiem ir iespēja atstāt atsauksmes un vērtējumus par īpašumiem un īpašniekiem.

Īpašniekiem ir iespēja reaģēt uz atsauksmēm un uzlabot savu piedāvājumu atbilstoši saņemtajai atgriezeniskajai informācijai.

- Ievad dati: Īrniekiem ir iespēja atstāt atsauksmes un vērtējumus par īpašumiem un īpašniekiem, kas sniedz iespēju citiem lietotājiem iegūt informāciju par īpašuma kvalitāti un īpašnieka uzvedību.
- Apstrāde: Sistēma saglabā atsauksmes un vērtējumus, lai tos varētu parādīt citiem lietotājiem un sniegt tiem papildu informāciju par konkrētu īpašumu vai īpašnieku.
- Izvaddati: Parāda atsauksmes un vērtējumus, lai lietotāji varētu izvēlēties īpašumus un īpašniekus, kuriem ir laba reputācija, un īpašniekiem sniedz iespēju uzlabot savus piedāvājumus, reaģējot uz saņemtajām atsauksmēm un vērtējumiem.

8. Iespēja pievienot pieejamību un noteikumus:

Klienta vēlmju saraksts:

Īpašniekiem ir iespēja pievienot informāciju par īpašuma īpatnībām, kas var būt svarīgas īrniekiem, piemēram, mājdzīvnieku draudzīgums vai īpašuma īpašie nosacījumi.

- Ievad dati: Īpašniekiem ir iespēja pievienot informāciju par īpašuma īpatnībām un noteikumiem, kas var ietekmēt īrnieku izvēli.
- Apstrāde: Sistēma saglabā īpašnieku pievienotos datus un noteikumus, lai tos parādītu potenciālajiem īrniekiem rezervēšanas laikā.
- Izvad dati: Parāda īrniekiem informāciju par pieejamību un īpašuma noteikumiem, lai palīdzētu viņiem izvēlēties piemērotu nekustamo īpašumu

9. E-pasta paziņojumi:

Personalizētie paziņojumi:

Lietotājiem var būt iespēja pielāgot e-pasta paziņojumus, izvēloties saņemt ziņojumus tikai par konkrētiem notikumiem vai darbībām.

- Personalizētie paziņojumi: Lietotāji var pielāgot e-pasta paziņojumus, lai saņemtu ziņas tikai par tiem notikumiem vai darbībām, kas viņiem ir svarīgi.
- Apstrāde: Sistēma izsūta personalizētus e-pasta paziņojumus, pamatojoties uz lietotāju iestatījumiem un notikumu atbilstību.
- Izvad dati: Lietotāji saņem pielāgotus e-pasta paziņojumus, kas atbilst viņu izvēlētajiem kritērijiem un interesēm.

10. Rezervāciju anulēšana un elastīga atcelšanas politika:

Sistēmai ir jānodrošina ērtības lietotājiem, kuri vēlas anulēt rezervāciju, nodrošinot viņiem intuitīvu procesu rezervācijas pārskatīšanai un atcelšanai.

Lietotājiem ir jābūt iespējai ātri un vienkārši piekļūt informācijai par rezervācijām un jāvar veikt atcelšanu, izmantojot skaidri norādītus soļus.

Elastīga atcelšanas politika nodrošina, ka lietotāji var viegli pārskatīt un mainīt rezervācijas termiņus vai pat pilnībā atcelt tās, ievērojot noteikumus, kuri samierinātu gan lietotāju vēlmes, gan sistēmas darbību.

Paziņojumi un apstiprinājumi par atceltajām rezervācijām tiek nodrošināti, lai lietotāji būtu informēti par veiktajām izmaiņām un iespējamām saistītām sekām.

Elastīgā atcelšanas politika veicina lietotāju uzticību sistēmai, nodrošinot saprātīgus un viegli saprotamus noteikumus rezervāciju pārvaldībai un atcelšanai.

- Ievaddati: Sistēmai ir jānodrošina ērtības lietotājiem, kuri vēlas anulēt rezervāciju, nodrošinot saprotamu un viegli izpildāmu procesu.
- Apstrāde: Sistēma nodrošina skaidru un saprotamu rezervācijas pārskatīšanas un atcelšanas procesu, lai lietotāji varētu viegli izdarīt izmaiņas saviem rezervācijas plāniem.
- Izvaddati: Paziņo lietotājiem par veiksmīgi veiktajām rezervāciju izmaiņām un nodrošina paziņojumus par atceltajām rezervācijām, lai viņi būtu informēti par veiktajām izmaiņām un to sekām.

11. Mājas pievienošana:

Lai pievienotu savu māju, lietotājiem no personīgā kabineta ir jāizvēlas sadaļa "Mana māja", kur būs pogas, lai pievienotu jaunu māju.

Tiek pārvietots uz lapu, kur nepieciešams ievadīt mājas informāciju, ieskaitot adresi, ērtības, cilvēku skaitu, fotogrāfijas un ģeolokāciju, kā arī cenu.

- Ievaddati: Lietotājiem no personīgā kabineta ir jāizvēlas sadaļa "Mana māja", kur būs pieejamas pogas, lai pievienotu jaunu māju.
- Apstrāde: Tiek pārvietots uz lapu, kur nepieciešams ievadīt mājas informāciju, tostarp adresi, ērtības, cilvēku skaitu, fotogrāfijas un ģeolokāciju, kā arī cenu.
- Izvaddati: Pēc veiksmīgas pievienošanas lietotājs var redzēt savu māju sarakstā un citi lietotāji to var atrast meklēšanas rezultātos un pārlūkošanas lapās.

12. Mājas izvadīšana

Pēc administrācijas apstiprinājuma mājas tiek publiskotas un kļūst pieejamas vietnes lietotājiem. Lietotāji var tos atrast meklēšanas rezultātos un pārlūkošanas lapās, kur tie tiek norādīti kā pieejami piedāvājumi. Tas palielina redzamību un pieejamību potenciālajiem īrniekiem, kuri var izvēlēties un rezervēt interesējošos nekustamos īpašumus, piedāvājot papildu iespējas saiknes un komunikācijas veidā ar īpašniekiem, ja nepieciešams. Kopumā mājas apstiprināšana administrācijas pusē ir būtisks solis, lai nodrošinātu vietnes kvalitāti un nodrošinātu lietotājiem uzticamu un drošu pieredzi.

- Ievaddati: Pēc administrācijas apstiprinājuma mājas tiek publiskotas un kļūst pieejamas vietnes lietotājiem.
- Apstrāde: Mājas tiek norādītas kā pieejami piedāvājumi meklēšanas rezultātos un pārlūkošanas lapās.

- Izvaddati: Tas palielina redzamību un pieejamību potenciālajiem īrniekiem un nodrošina papildu iespējas saiknes un komunikācijas veidā ar īpašniekiem.

13. Reģistrētie īpašumi

Šeit tiek attēlots saraksts ar visiem īpašumiem, ko lietotājs ir reģistrējis. Katram īpašumam ir sniegta detalizēta informācija, ieskaitot adresi, ērtības, pieejamību un cenu.

- Ievaddati: Šeit tiek attēlots saraksts ar visiem lietotāja reģistrētajiem īpašumiem.
- Apstrāde: Katram īpašumam ir sniegta detalizēta informācija, ieskaitot adresi, ērtības, pieejamību un cenu.
- Izvaddati: Lietotājs var pārskatīt savu īpašumu sarakstu un saņemt informāciju par katru īpašumu individuāli.

14. Īpašumu vēsture

Šajā sadaļā lietotāji var pārskatīt visu savu īpašumu vēsturi, tostarp iepriekšējās rezervācijas, saņemtās atsauksmes un jebkādas citus saistītos notikumus. Tas nodrošina īpašniekiem pārskatu par viņu īpašumu darbību un ļauj izsekot visiem svarīgajiem notikumiem un rezervāciju vēsturei.

- Ievad dati: Šeit tiek attēlots saraksts ar visiem lietotāja reģistrētajiem īpašumiem.
- Apstrāde: Katram īpašumam ir sniegta detalizēta informācija, ieskaitot adresi, ērtības, pieejamību un cenu.
- Izvaddati: Lietotājs var pārskatīt savu īpašumu sarakstu un saņemt informāciju par katru īpašumu individuāli.

15. Mājas rezervācijas pārbaude

Funkcija pārbauda, vai izvēlētie datumi ir pieejami rezervācijai, balstoties uz jau esošajām rezervācijām. Ja datumi sakrīt ar jau esošajām rezervācijām, tie tiek atzīmēti kā aizņemti un kalendārā tiek atspējoti, lai lietotājs nevarētu veikt rezervāciju uz šiem datumiem.

- Ievad dati: Funkcija pārbauda, vai izvēlētie datumi ir pieejami rezervācijai, balstoties uz jau esošajām rezervācijām.
- Apstrāde: Ja datumi sakrīt ar jau esošajām rezervācijām, tie tiek atzīmēti kā aizņemti un kalendārā tiek atspējoti.
- Izvaddati: Lietotājiem tiek sniegta informācija par rezervācijas iespējamību noteiktajos datuma diapazonos, kas nodrošina skaidrību un samazina rezervācijas konfliktu iespējamību.

16. Māju piedāvājumus izvade uz atsevišķas lapas ar to identifikatoriem

Funkcija iegūst visus pieejamos māju piedāvājumus no datu bāzes un tos izvada uz atsevišķas lapas. Katrs piedāvājums tiek attēlots kopā ar tā identifikatoru, lai lietotājs varētu identificēt konkrēto māju un pieprasīt tās detaļas.

- Ievad dati: Funkcija iegūst visus pieejamos māju piedāvājumus no datu bāzes.
- Apstrāde: Katrs piedāvājums tiek izvadīts uz atsevišķas lapas kopā ar tā identifikatoru.
- Izvad dati: Lietotāji var identificēt konkrēto māju un pieprasīt tās detaļas, jo katrs piedāvājums ir attēlots ar savu identifikatoru.

17 Visu lietotāja piedāvājumu izvade

Šī funkcija ir atbildīga par attēlojot visus lietotāja izvietotos piedāvājumus AG Grid tabulā. Lai to paveiktu, funkcija sazinās ar Firestore datubāzi, lai iegūtu visus lietotāja piedāvājumus un tos pēc tam attēlo tabulā.

Funkcija sāk ar definēšanu un inicializēšanu, tai tiek izmantota React Hooks useState un useEffect, lai saglabātu piedāvājumu datus un izpildītu datu ieguves darbību, kad komponente tiek izveidota. Pēc tam funkcija izveido AG Grid tabulu, kurai ir definētas kolonnas, kuras attēlos katru piedāvājuma atribūtu, piemēram, ID, nosaukumu un cenu.

Visbeidzot, funkcija atgriež AG Grid komponenti, kas attēlo tabulu ar visiem lietotāja izvietotajiem piedāvājumiem.

- Ievad dati: Funkcija sazinās ar Firestore datubāzi, lai iegūtu visus lietotāja piedāvājumus.
- Apstrāde: Attiecīgie dati tiek saglabāti un attēloti AG Grid tabulā.
- Izvad dati: Lietotāji var redzēt visus savus izvietotos piedāvājumus tabulā.

18. Karte

Šī funkcija ir atbildīga par attēlošanu un ļauj lietotājam norādīt savas mājas atrašanās vietu uz kartes. Lietotājam ir iespēja noklikšķināt uz kartes, lai norādītu precīzu atrašanās vietu.

Funkcija sākas ar inicializēšanu un definēšanu, tai ir stāvoklis, kas saglabā lietotāja norādīto atrašanās vietu uz kartes.

Tālāk funkcija attēlo karti, kurā lietotājs var norādīt mājas atrašanās vietu, izmantojot marķieri un kartes noklikšķināšanu.

- Ievad dati: Lietotāja norādītā atrašanās vieta uz kartes.
- Apstrāde: Attiecīgie stāvokļi saglabā lietotāja norādīto atrašanās vietu uz kartes.
- Izvaddati: Lietotāji var norādīt savas mājas atrašanās vietu uz kartes, izmantojot marķieri un kartes noklikšķināšanu.

19. Čats

Šī funkcija nodrošina iespēju lietotājam sūtīt un saņemt ziņas no dzīvokļa īpašnieka, lai pārrunātu nomas detaļas vai izteiktu jautājumus par dzīvokli.

Funkcija sākas ar definēšanu un inicializēšanu. Tajā tiek izmantoti divi stāvokļi - viens, lai uzglabātu jaunās ziņas tekstu, un otrs, lai saglabātu visu čata vēsturi.

Pēc tam funkcija attēlo čata logu, kurā redzami visi līdz šim saņemtie un nosūtītie ziņojumi.

Papildus tam tiek attēlots ievades lauks, kurā lietotājs var rakstīt un nosūtīt jaunas ziņas.

Kad lietotājs nosūta jaunu ziņu, funkcija to pievieno pie čata vēstures un iztukšo ievades lauku, lai ļautu lietotājam sūtīt nākamo ziņu.

Šī funkcija ir noderīga, lai nodrošinātu komunikāciju starp nomas dzīvokļa īpašnieku un nomnieku, tādējādi ļaujot pārrunāt nomas detaļas, atbildēt uz jautājumiem vai izteikt komentārus par dzīvokli.

- Ievad dati: Funkcija sākas ar definēšanu un inicializēšanu, izmantojot divus stāvokļus.
- Apstrāde: Attēlo čata logu ar visiem saņemtajiem un nosūtītajiem ziņojumiem.
- Izvades parametri: Lietotājiem ir iespēja sūtīt un saņemt ziņas no dzīvokļa īpašnieka, lai pārrunātu nomas detaļas vai izteiktu jautājumus par dzīvokli.

20. Profils

Šī funkcija ļauj lietotājam iegūt piekļuvi citu lietotāju profilam, lai apskatītu informāciju par šo lietotāju.

Funkcija sākas ar definēšanu un inicializēšanu. Tās galvenais mērķis ir nodrošināt piekļuvi citu lietotāju profiliem.

Lietotājs var norādīt citu lietotāju lietotājevārdu vai ID, lai atrastu un apskatītu attiecīgo profilu.

Pēc tam funkcija izpilda pieprasījumu, lai iegūtu informāciju par izvēlēto lietotāju profilu.

Informācija, ko varētu attēlot par citu lietotāju profilu, ietver lietotāja vārdu, uzvārdu, e-pasta adresi, dzimšanas datumu, profila bildi, sociālo saikni un citus svarīgus datus.

Funkcija nodrošina iespēju apskatīt un pārbaudīt citu lietotāju profilu informāciju, lai lietotāji varētu iepazīties ar citu lietotāju datiem vai meklēt konkrētu informāciju par noteiktu lietotāju.

- Ievad dati: Lietotājs norāda citu lietotāju lietotājvārdu vai ID, lai atrastu un apskatītu attiecīgo profilu.
- Apstrāde: Izpilda pieprasījumu, lai iegūtu informāciju par izvēlēto lietotāju profilu.
- Izvades parametri: Lietotājiem tiek piedāvāta iespēja apskatīt un pārbaudīt citu lietotāju profilu informāciju, kā arī atrast nepieciešamo informāciju par noteiktu lietotāju.

21. Cenas aprēķināšana

Aprēķinā cenu nekustamā īpašuma vienībai, iekļaujot 12% komisijas maksu.

- Ievades dati: Cena nekustamā īpašuma.
- Apstrāde: Aprēķināt 12% no norādītās cenas un pievienot to norādītajai cenu summai.
- Izvaddati: Kopējā cena, iekļaujot 12% komisijas maksu.

22. Neeksistējošais URL

Parādīt ziņojumu, ja lietotājs mēģina piekļūt neeksistējošam URL.

- Ievades dati: Nav nepieciešami.
- Apstrāde: Izvadīt ziņojumu "Atvainojiet, jūsu meklētā lapa neeksistē".
- Izvaddati: Ziņojums "Atvainojiet, jūsu meklētā lapa neeksistē".

23. Administratoru reģistrs

Lietotājiem, kuri nav administrators, nav atļauts piekļūt lietotāju reģistra un īpašumu reģistra lapām.

- Ievades dati: Nav nepieciešami.
- Apstrāde: Pārbaudīt lietotāja lomu (administratoru vai parastu lietotāju).
- Izvaddati: Atļaut vai aizliegt piekļuvi lietotāju reģistra un īpašumu reģistra lapām atkarībā no lietotāja lomas.

24. Konta deaktivizēšana

Lietotājs nevar piekļūt savam kontam, ja tas ir deaktivizēts.

- Ievades dati: Lietotāja konta statuss (aktīvs vai deaktivizēts).
- Apstrāde: Pārbaudīt lietotāja konta statusu. Ja tas ir deaktivizēts, bloķēt piekļuvi.
- Izvaddati: Ziņa vai darbība, kas norāda, ka piekļuve ir bloķēta.

25. Kalendārs sākumlapā

Galvenajā lapā jābūt kalendāram, kas atrod pieejamas nekustamā īpašuma vienības atbilstoši ievadītajiem datumiem.

- Ievades dati: Pieejamie nekustamā īpašuma vienību datumi.
- Apstrāde: Atjaunināt kalendāru ar atbilstošajiem datumiem par pieejamajām nekustamā īpašuma vienībām.
- Izvaddati: Kalendārs ar atjauninātām nekustamā īpašuma vienībām atbilstoši ievadītajiem datumiem.

26. Pieejas ierobežojumi neautorizētiem lietotājiem

Lietotājs nevar ierēt māju, pievienot savu māju, skatīt savu profilu un sūtīt ziņojumus, ja viņš nav autorizējies sistēmā.

- Ievades dati: Autorizācijas statuss (autentificēts vai neautentificēts).
- Apstrāde: Pārbaudīt lietotāja autorizācijas statusu. Ja lietotājs nav autentificējies, ierobežot piekļuvi noteiktām darbībām.
- Izvaddati: Ziņa vai darbība, kas norāda, ka noteiktas darbības ir ierobežotas neautorizētiem lietotājiem.

27. Nekustamā īpašuma pārvaldība:

Informācijas pievienošana un rediģēšana:

Īpašniekiem ir piekļuve intuitīvam interfeisam, lai pievienotu un rediģētu nekustamā īpašuma informāciju, tādu kā adrese, īpašuma apraksts, maksimāli pieļaujamais cilvēku skaits mājā, geo lokācija, mājas ērtību saraksts, un arī var augšupielādēt attēlus, kas piedāvā lietotājiem vizuālu priekšstatu par īpašumu.

- Ievad dati: Īpašniekiem ir piekļuve intuitīvam interfeisam, kur tie pievieno un rediģē nekustamā īpašuma informāciju, piemēram, adresi, aprakstu un attēlus.
- Apstrāde: Pārbauda un saglabā informāciju par nekustamā īpašuma pievienošanu vai rediģēšanu datu bāzē.
- Izvaddati: Paziņo par veiksmīgu nekustamā īpašuma informācijas pievienošanu vai rediģēšanu, vai parāda kļūdas paziņojumu, ja process neizdodas.

2.3. Sistēmas nefunkcionālās prasības

1. Veiktspēja:

Lai lietotājiem nodrošinātu ātru piekļuvi informācijai, lapas ielādes laiks nedrīkst pārsniegt 3 sekundes.

Sistēmai jāatbalsta vismaz 500 vienlaicīgu lietotāju bez ievērojamas veiktspējas pasliktināšanās.

2. Pieejamība:

Vietnei jābūt pieejamai lietotājiem vismaz 99 % laika, ieskaitot uzturēšanas periodus.

Datu dublēšana jāveic katru nedēļu, un datu atjaunošanai jābūt iespējamai 24 stundu laikā.

3. Drošība:

Sistēmā jāizmanto HTTPS protokols, lai aizsargātu sensitīvu datu pārraidi.

Lietotāju paroles jāglabā šifrētā veidā.

Nodrošināt aizsardzības pasākumus pret XSS (cross-site scripting) un citiem izplatītiem uzbrukumu veidiem.

4. Pieejamība:

Vietnei jābūt optimizētai darbam dažādās pārlūkprogrammās, piemēram, Chrome, Firefox un Internet Explorer.

Nodrošiniet savietojamību ar mobilajām ierīcēm, lai lietotāji varētu ērti izmantot vietni dažādās platformās.

5. Kļūdu pārvaldība:

Sniegt informatīvus un lietotājam saprotamus kļūdu ziņojumus.

Uzturēšanas izmaksas:

Noteikt un uzturēt budžetu sistēmas uzturēšanai, tostarp hostinga, atjauninājumu un atbalsta izmaksām.

6.Lietderība:

Saskarnes dizainam jābūt intuitīvam un draudzīgam dažādām lietotāju kategorijām.

7.Dizains un lietojamība:

- Vietnes galvenajām krāsām jābūt tumšām, tuvāk pelēkajām tonējumiem.
- Sadaļā "Mans profils" navigācijas joslai jāatrodas kreisajā pusē vertikālā izkārtojumā.
- Pie ērtību saraksta jābūt pievienotām ikonām, kas atbilst ērtībām.

8.Sistēmas valoda:

Visa vietne jāatbalsta latviešu valodā, ieskaitot tekstu, pogas, virsrakstus un citu saturu.

2.4. Gala lietotāja raksturiezīmes

Gala lietotājiem ir viegli saprast un izmantot programmu, pateicoties vienkāršajai un draudzīgai lietotāja saskarnei.

Personalizēta pieredze:

Programma nodrošina iespēju pielāgot meklēšanas kritērijus, lai lietotāji varētu atrast nekustamo īpašumu, kas pilnībā atbilst viņu individuālajām vajadzībām.

Notifikācijas un atgādinājumi:

Lietotāji saņem paziņojumus par jauniem piedāvājumiem, izmaiņām cenās vai citiem svarīgiem notikumiem, tādējādi palīdzot uzturēt interesi par piedāvājumiem tirgū.

Drošība un privātums:

Gala lietotāji var būt pārliecināti par savu datu un finanšu drošību, izmantojot programmu, kura ievēro augstus drošības standartus.

Finanšu pārvaldība:

Lietotājiem ir pieejami rīki, kas ļauj efektīvi pārvaldīt maksājumus, rēķinus un citas saistītās finanšu darbības tieši no vienas platformas.

Programma kopumā nodrošina pozitīvu un individualizētu pieredzi, padarot nekustamā īpašuma meklēšanu un īri lietotājiem pieejamu, ērtu un efektīvu.

3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

Izstrādes līdzekļu un rīku izvēle ir būtiska daļa no katra izstrādātāja vai izstrādes komandas darbības, nosakot veidu, kā tiek veidotas programmas un lietojumprogrammas. Katrs izvēlētais instruments un līdzeklis ietekmē ne tikai izstrādes procesa efektivitāti, bet arī gala rezultātu. Šajā kontekstā šī raksta mērķis ir sniegt ieskatu un pamatojumu izvēlētiem izstrādes līdzekļiem un rīkiem, izklāstot to aprakstus un izvēles pamatojumus. Šī analīze aplūko izstrādes vide, valodas, bibliotēkas, koda pārvaldību un citas tehnoloģijas, kas veido pamatu programmēšanas projektam.

3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts

Visual Studio Code:

Apraksts: VSCode ir brīnišķīga, viegli pielāgojama izstrādes vide, ko nodrošina Microsoft. Tas ir izplatīts starp programmētājiem, jo tas ir ātrs, piedāvā bagātīgu paplašinājumu klāstu, un tam ir iebūvētas ērtas funkcijas, piemēram, koda atklāvošana un integrācija ar versiju kontroles sistēmām.

Pamatojums: VSCode tika izvēlēts tā funkcionalitātes un vieglās lietojamības dēļ. Paplašinājumu ekosistēma sniedz iespēju pielāgot vidi un pievienot funkcionalitāti atbilstoši konkrētajiem uzdevumiem.

HTML, CSS, JavaScript:

Apraksts: HTML (HyperText Markup Language), CSS (Cascading Style Sheets), un JavaScript ir pamatvalodas tīmekļa izstrādei. HTML nodrošina struktūru, CSS - izskatu un stilu, bet JavaScript - dinamiskas funkcijas un interaktivitāti tīmekļa vietnēs.

Pamatojums: Šīs valodas ir standarta tīmekļa izstrādē un tiek plaši atbalstītas visās mūsdienu pārlūkprogrammās. Tās ļauj radīt daudzveidīgas un responsīvas tīmekļa vietnes un lietojumprogrammas.

Node.js:

Apraksts: Node.js ir izpildlaiks, kas ļauj izpildīt JavaScript kodu ārpus pārlūka. Tas ir ļoti populārs serveru puses izstrādes rīks, piedāvājot asinkronu kodu izpildi un vieglu skalējamību.

Pamatojums: Node.js tika izvēlēts, lai veiktu servera puses darbības un sadarbību ar datu bāzi, nodrošinot vienotu valodu (JavaScript) gan klienta, gan servera pusē.

React.js:

Apraksts: React.js ir deklarātīva un efektīva JavaScript bibliotēka, ko izmanto lietotāja saskarnes izveidei. Tas piedāvā komponentu orientētu izstrādi, kas atvieglo lietojumprogrammas stāvokļa pārvaldību.

Pamatojums: React.js tika izvēlēts, jo tas piedāvā modernas iespējas lietojumprogrammu izveidei, atsaucīgu dizainu un vienkāršu komponentu atkārtotu izmantošanu, kas palielina izstrādes efektivitāti.

Firestore datu bāze:

Apraksts: Firestore ir platforma, ko nodrošina Google, un tā ietver daudzas funkcijas, tai skaitā reālā laika datu bāzi, autentifikāciju, un tiešsaistes glabātuvi.

Pamatojums: Firestore tika izvēlēts kā vienkāršs un viegli lietojams risinājums, kas ļauj ātri sākt darbu ar datu bāzi un citiem tīmekļa pakalpojumiem, neaizņemoties ar sarežģītu servera infrastruktūru.

GitHub:

Apraksts: GitHub ir platforma koda pārvaldībai, kas ļauj izstrādātājiem glabāt un uzturēt savu kodu, izmantojot versiju kontroles sistēmu Git. Tas piedāvā iespēju sekot līdzi kodam, pārvaldīt izmaiņas un organizēt projektu struktūru.

Pamatojums: GitHub ir noderīgs, pat strādājot viens pats, jo tas nodrošina drošu un centralizētu vietu kodam, atvieglo versiju pārvaldību, un piedāvā dažādas funkcijas, piemēram, projektu izsekotājus, kas var būt noderīgi, ja vēlies sekot līdzi izmaiņām kodā vai atgriezties pie iepriekšējām versijām.

Opera GX pārlūks:

Apraksts: Opera GX ir interneta pārlūks, kas piedāvā vairākas īpašības, galvenokārt vērstas uz spēlētājiem un resursu pārvaldību. Tas ietver funkcijas, piemēram, spēļu aparatūras apstākļu pārvaldību, resursu bloķēšanu, bezspēkā pārvaldītu video un ātru interneta pārlūkošanu.

Pamatojums: Opera GX tika izvēlēta, jo tā piedāvā īpašības, kas ļauj efektīvāk testēt un optimizēt tīmekļa lietojumprogrammu atbilstību šim pārlūkam. Spēļu aparatūras apstākļu pārvaldība ļauj pārbaudīt resursu intensīvus procesus, un bezspēkā pārvaldītu video sniedz iespēju testēt, kā lietojumprogramma darbojas ar dažādiem video formātiem un kvalitāti.

3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts

Visual Studio Code:

Sublime Text: Ālternatīva, kas ir viegli pielāgojama un piedāvā daudz paplašinājumu.

Atom: Brīvi pieejams redaktors ar daudzām pielāgošanas iespējām.

	Visual Studio Code	Sublime Text	Atom	Notepad++	Brackets	Eclipse	NetBeans
Integrēta izstrādes vide	✓	X	X	X	✓	✓	✓
Ērtības web izstrādei	✓	X	X	X	✓	X	X
Realtime kolaborācija	✓	✓	✓	X	X	X	X
Integrēts terminālis	✓	✓	✓	X	X	X	X
Versiju kontrole	✓	X	X	X	X	✓	✓
Paplašinājumu katalogs	✓	✓	✓	X	X	X	X
Pielāgojama izskata tēma	✓	✓	✓	✓	✓	X	X
Java izstrāde pielagosana	X	X	X	X	X	✓	✓

HTML, CSS, JavaScript:

Vue.js: Kā alternatīva React.js, piedāvā vienkāršu integrāciju un efektīvu komponentu sistēmu.

Svelte: Jaunākā pieeja lietotāja saskarnes izstrādei, kas piedāvā vienkāršu un efektīvu kodu.

Node.js:

Deno: Moderns, drošs un viegli pielāgojams alternatīvais risinājums Node.js.

NestJS: Ietvars, kas balstīts uz TypeScript, piedāvā lielu funkcionalitāti serveru izstrādei.

React.js:

Angular: Pilnīgāka ietvara alternatīva React.js, kas piedāvā lielu funkcionalitāti un strukturētību.

Vue.js: Vienkārša un elastīga JavaScript bibliotēka lietotāja saskarnes izveidei.

Firebase datu bāze:

MongoDB: Elastīga datu bāze ar dokumentu oriētētu modeli, piedāvā alternatīvu Firebase datu bāzei.

Firestore: Google Cloud piedāvāta datu bāzes pakalpojuma alternatīva, ko var integrēt ar citiem Google risinājumiem.

GitHub:

GitLab: Cita platforma koda pārvaldībai, piedāvā privātus repozitorijus un CI/CD iespējas.

Bitbucket: Atbilstoša izvēle, piedāvājot kodu pārvaldību un integrāciju ar Jira.

Opera GX pārlūks:

Google Chrome: Plaši izplatīts pārlūks ar labu veiktspēju un attīstītāju rīkiem.

Mozilla Firefox: Stabils un brīvi pieejams pārlūks, piedāvājot plašas pielāgošanas iespējas.

4. Sistēmas modelēšana un projektēšana

Izmantojot šīs sadaļas ievadu, ir skaidri redzams, ka darbības diagrammas sniedz vispārīgu priekšstatu par "Ēku īres sistēmas" darbības loģiku, procesiem un mijiedarbību starp dažādiem elementiem. Šīs diagrammas ir vērtīgas, lai izprastu sistēmas struktūru un funkcionalitāti, un tās tiek izmantotas dažādu aspektu modelēšanai. Esmu izmantojis sekojošas diagrammas, lai nodrošinātu padziļinātu sistēmas izpratni un palīdzētu tās izstrādē:

Sistēmas struktūras modelis - Klasifikācijas diagramma (Class Diagram) un ER-diagramma (Entity-Relationship Diagramma).

Klašu diagramma (Class Diagramma) - strukturēta diagramma, kas parāda sistēmas klases un to attiecības, sniedzot informāciju par klases atribūtiem, metodēm un to savstarpējām saistībām.

ER-diagramma (Entity-Relationship Diagramma) - grafisks attēlojums, kas parāda datu entitātes un to attiecības sistēmā, izprast datu struktūru un savstarpējo saistību starp datu entitātēm.

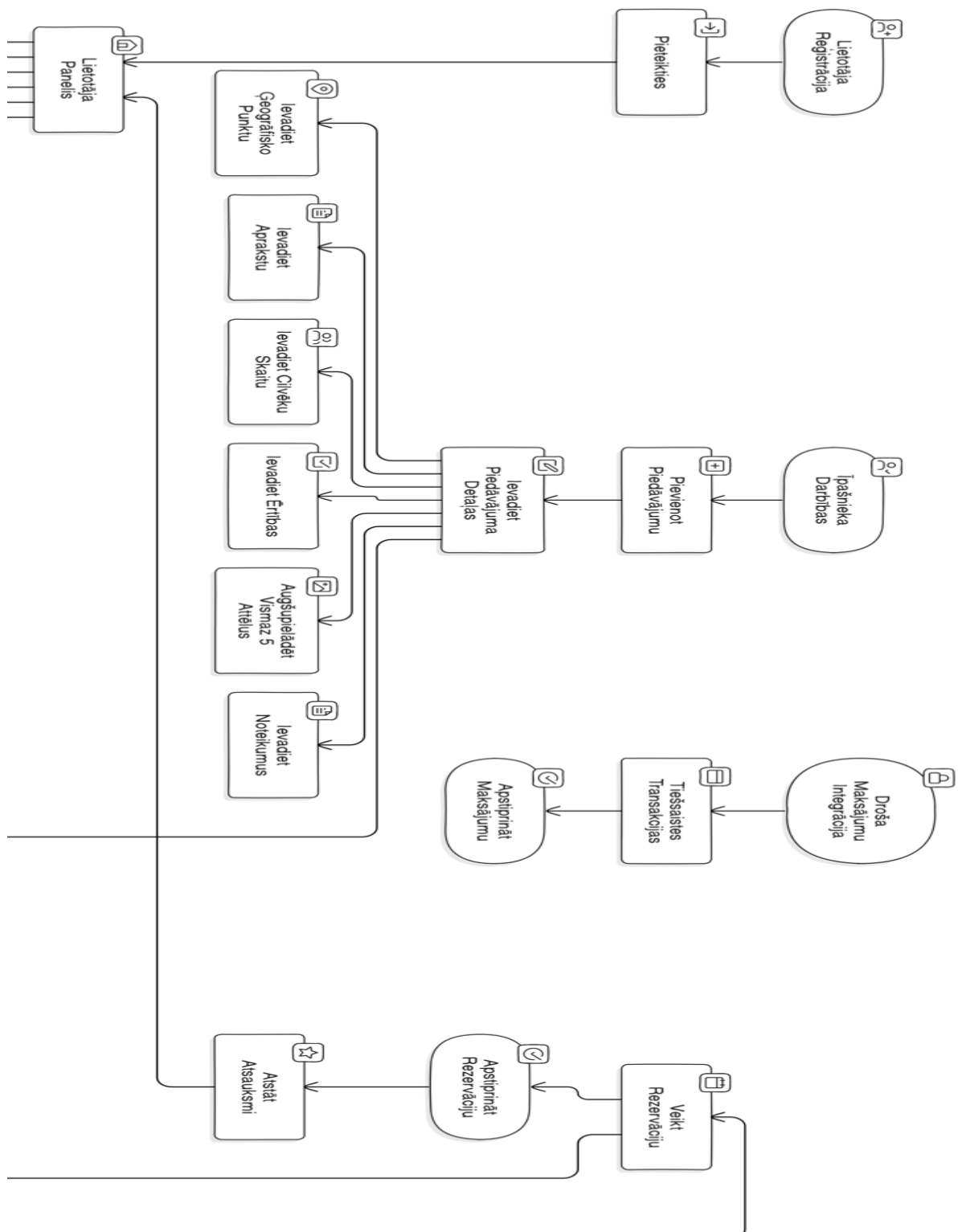
Aktivitāšu diagramma (Activity Diagramma) - diagramma, kas attēlo darbplūsmu un aktivitātes secību sistēmā, ļaujot izsekot procesiem un darbībām.

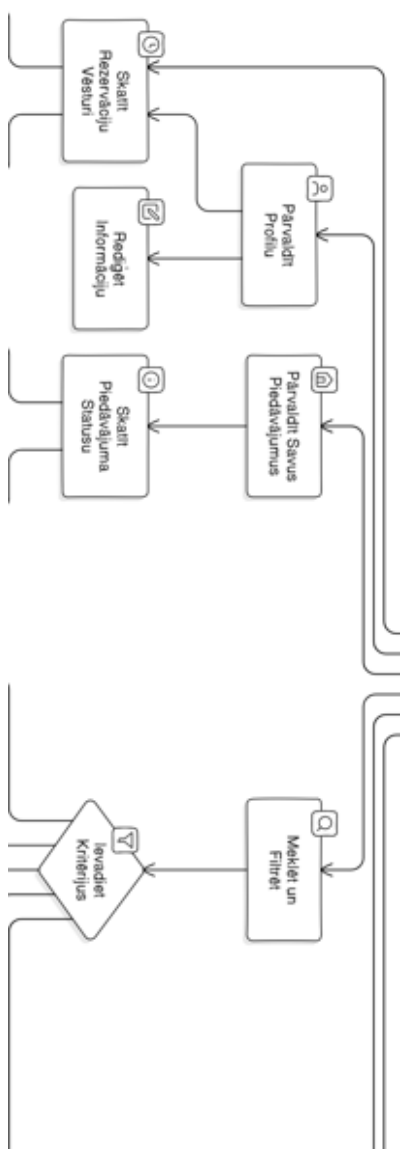
Lietojumgadījumu diagramma (Use Case Diagramma) - grafisks attēlojums, kas parāda sistēmas funkcionalitāti no lietotāja perspektīvas, identificējot un aprakstot lietotāja prasības un to, kā sistēma reaģē uz tām.

Secības diagramma (Sequence Diagramma) - grafisks attēlojums, kas parāda objektu mijiedarbību un komunikāciju sistēmā secībā, sniedzot informāciju par to, kā dažādi sistēmas objekti mijiedarbojas, lai izpildītu konkrētu uzdevumu vai funkcionalitāti.

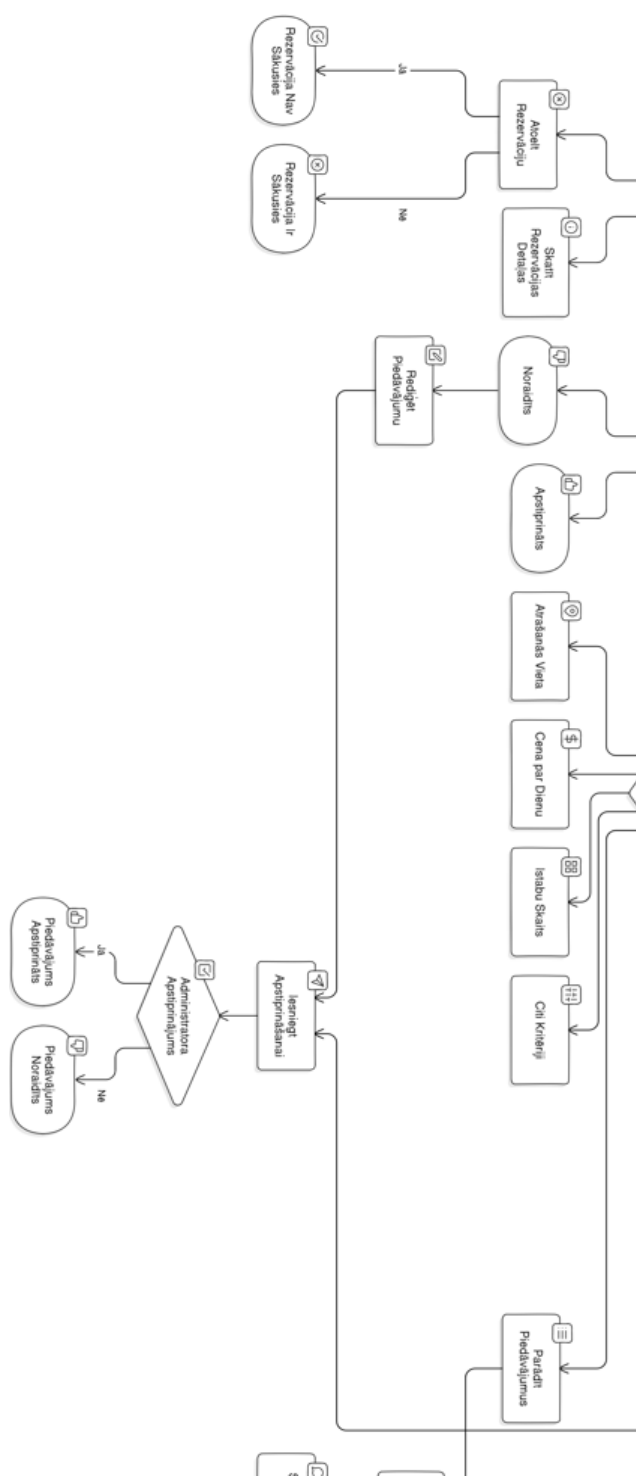
Izmantojot šīs diagrammas, esmu spējis nodrošināt padziļinātu izpratni par "Ēku īres sistēmas" struktūru, darbību un lietojumprogrammu, kas ir bijis noderīgs rīcības plānošanā un sistēmas izstrādē.

4.1. Sistēmas struktūras modelis

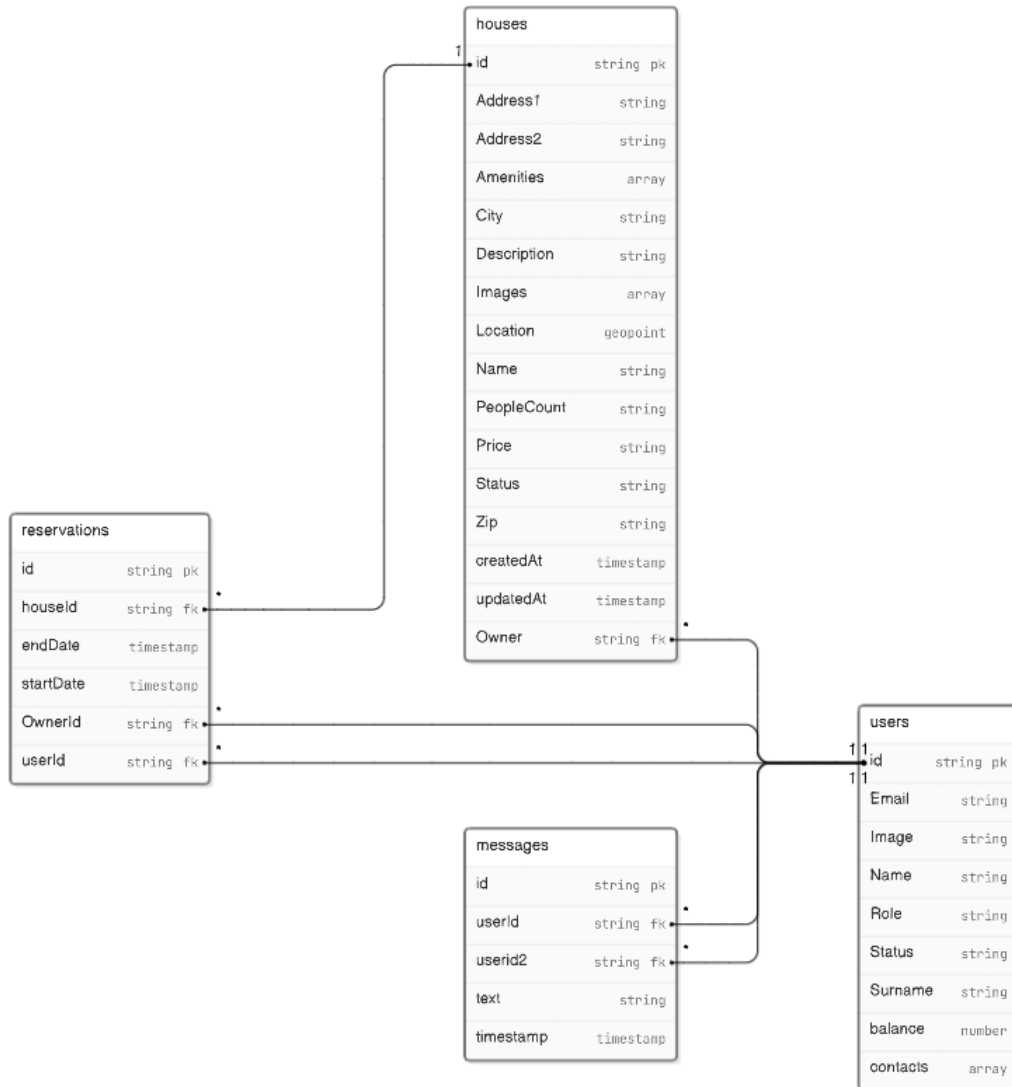




26

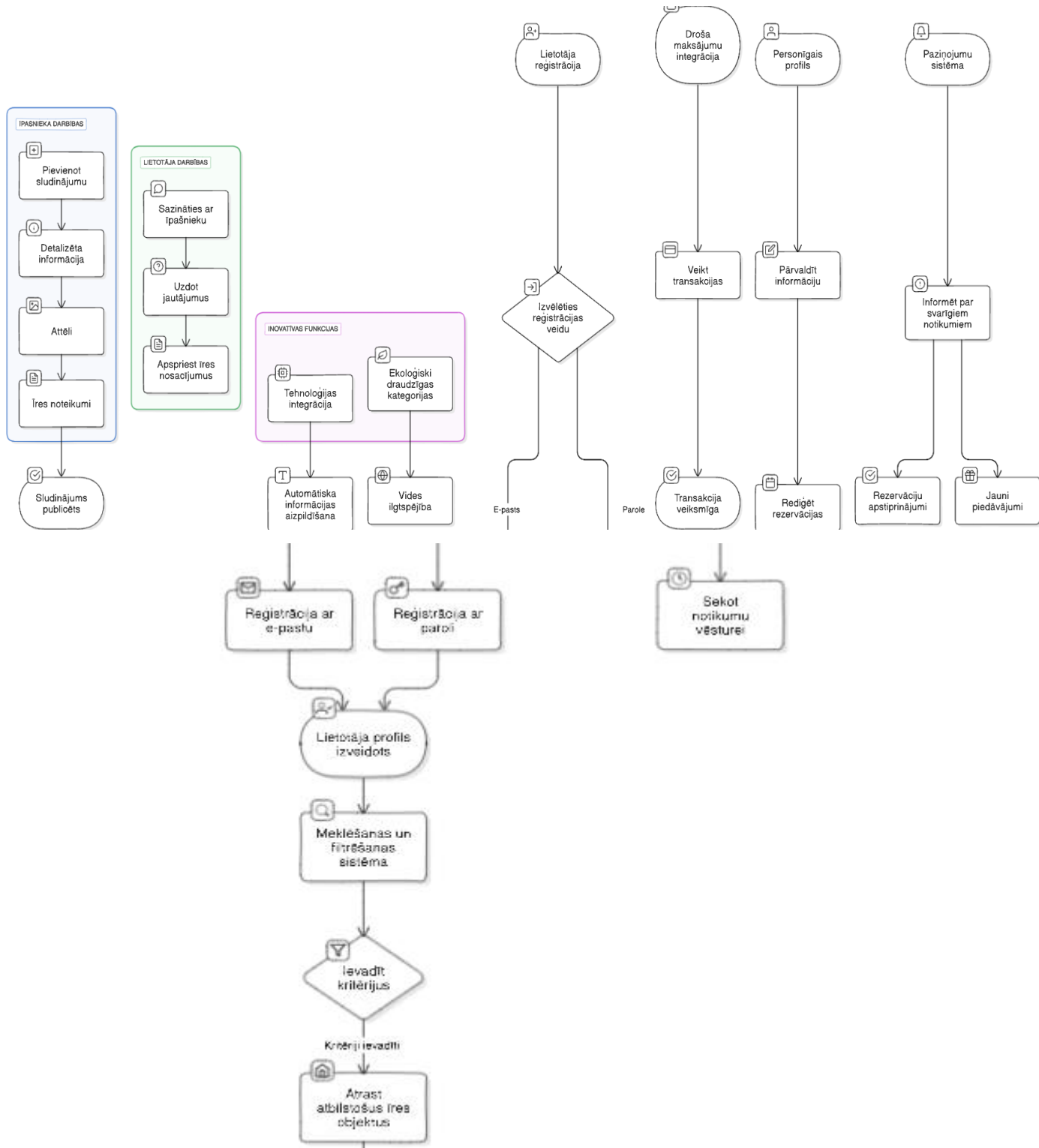


4.2. ER diagramma



2.attels. ER diagramma

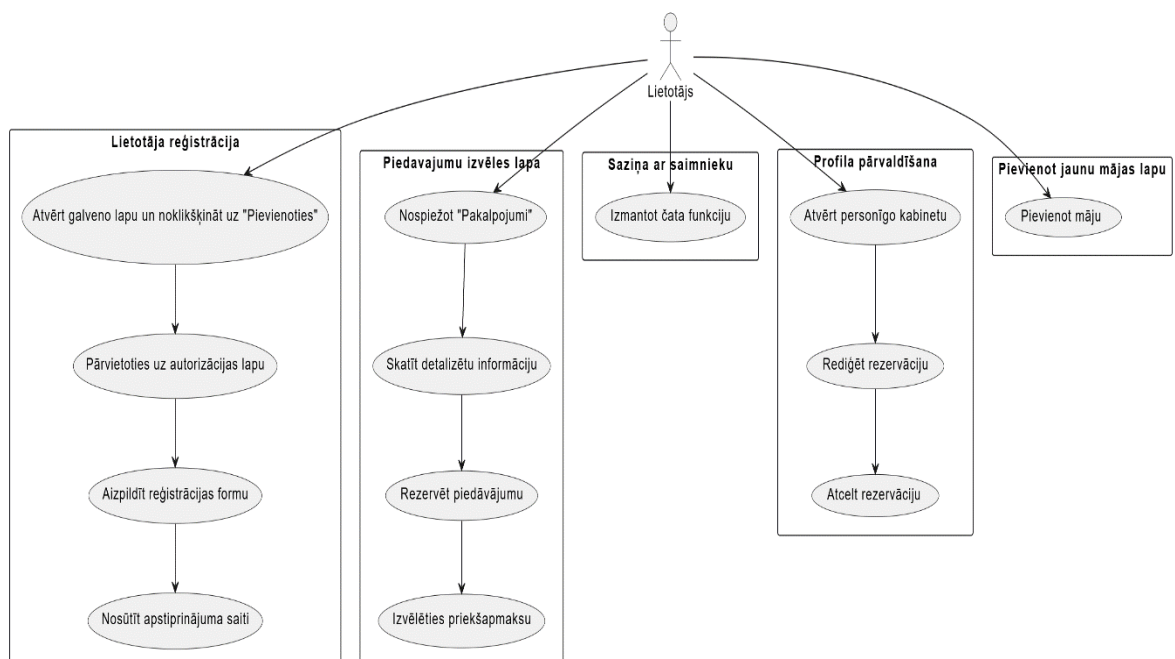
4.3. Aktivitāšu diagramma (Activity)





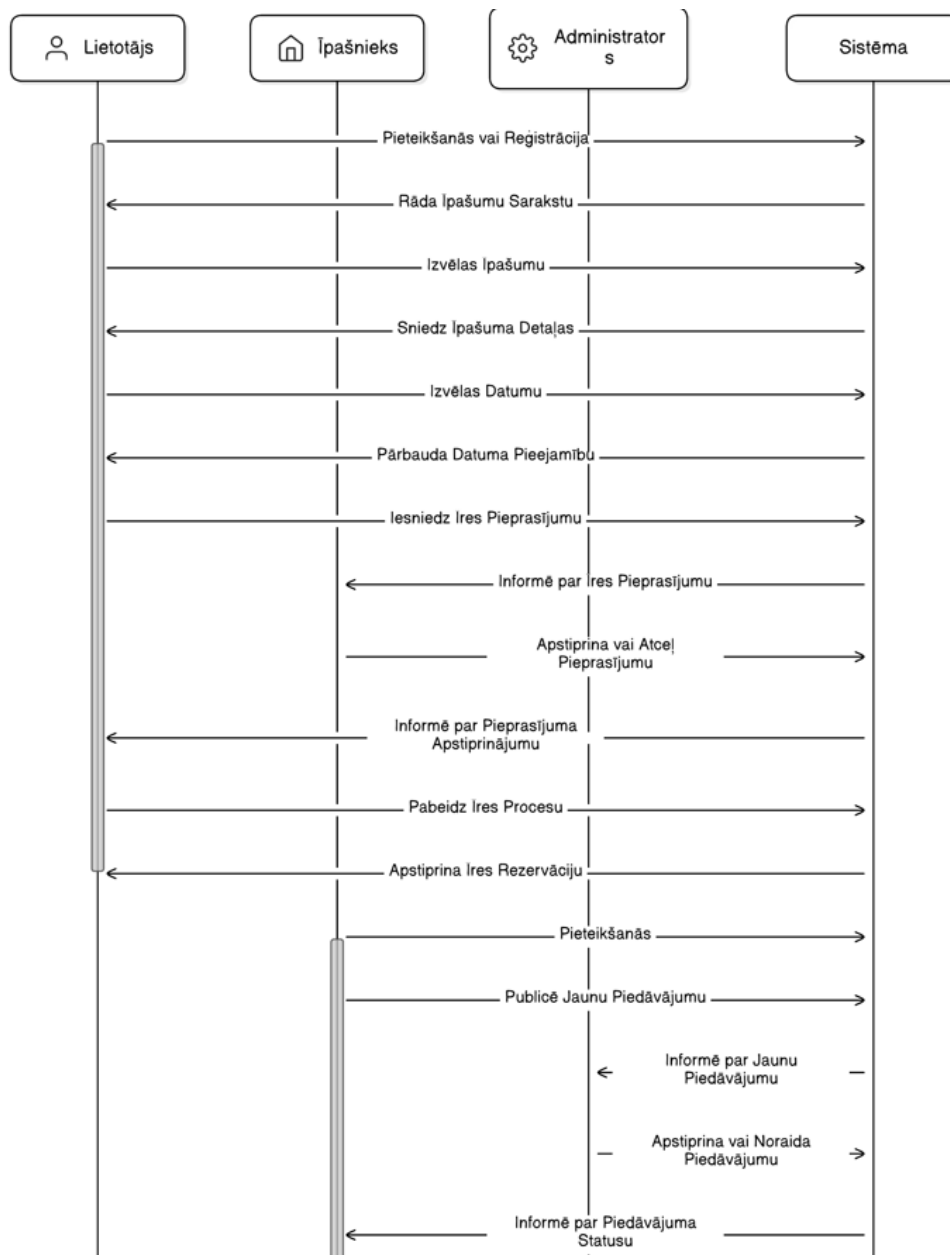
3.attels. Aktivitāšu diagramma

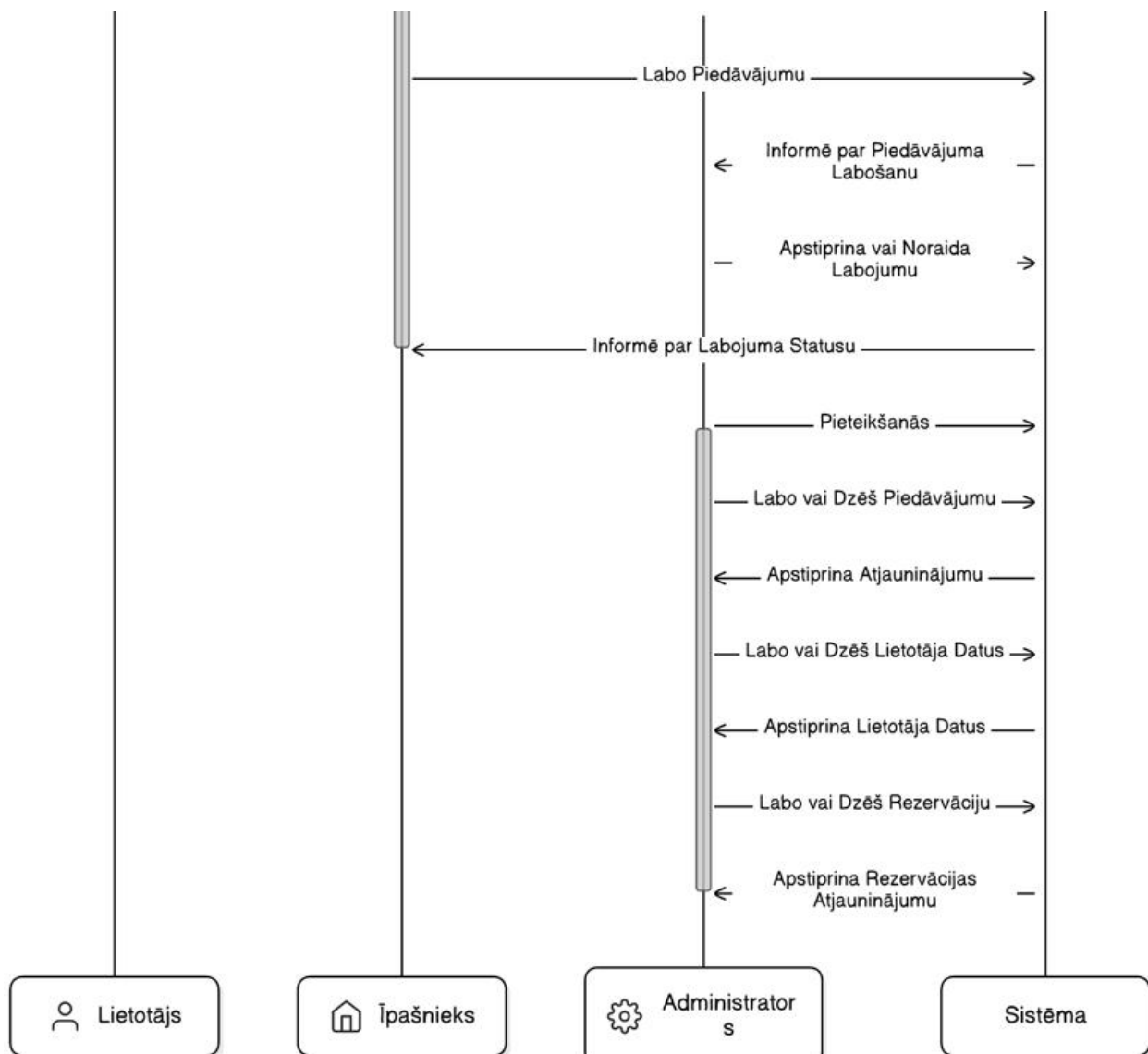
4.4. Lietojumgadījumu diagramma (Use Case)



4.attels. Lietojumgadījumu diagramma

4.5. secības diagramma



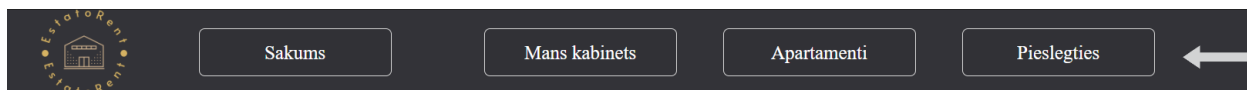


5.attels. secības diagramma

5. Lietotāju ceļvedis

Sākums un autorizācija:

Atver galveno lapu un noklikšķini uz "Pislegties" pogas, kas atrodas galvenajā izvēlnē.



5.attels. Sākums un autorizācija

Tiks pārvietots uz autorizācijas lapu, kur būs divas opcijas: "Reģistrēties" jaunam lietotājam vai "Ienākt" esošam lietotājam.

[Pislegties sistemai](#)

[Registreties](#)

Epasta Adrese *

Parole *

☐ Atcereties mani

PIESLEGITIES

Copyright © EstatoRent 2024.

6.attels. Pieslegšanās logs

Reģistrācija: Ja izvēlies "Reģistrēties", tev būs jāaizpilda reģistrācijas forma ar taviem personiskajiem datiem, tostarp vārdu, uzvārdu, e-pasta adresi un paroli.

Pieslēgties sistamai

Reģistrēties

Vards *

Uzvards *

Epasta Adrese *

Parole *

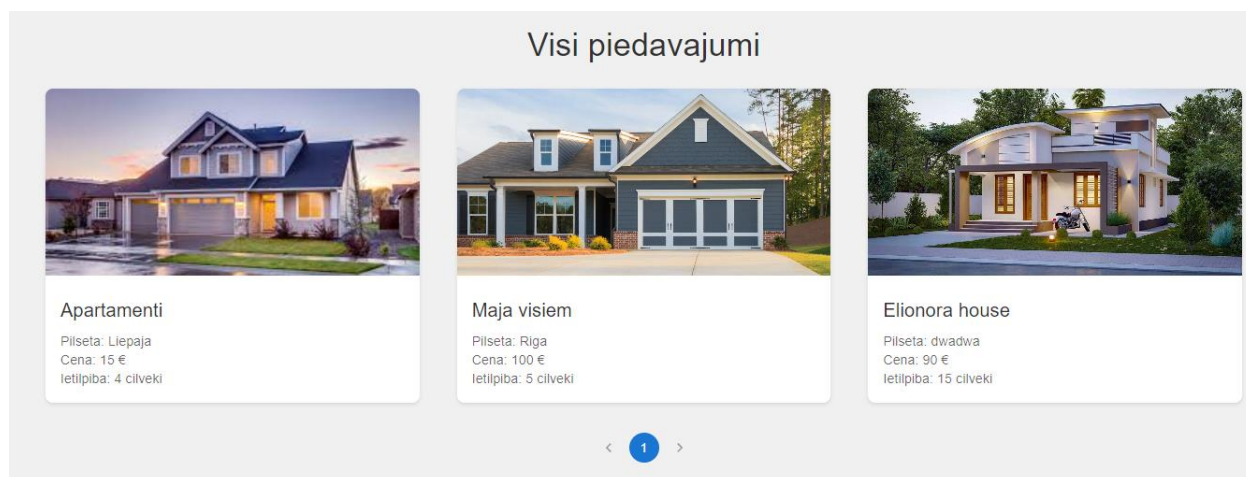
REGISTRETIES

Copyright © [EstateRent](#) 2024.

7.attels. Reģistrācijas

Kad esi aizpildījis formu, nospied pogu "Reģistrēties". Pēc tam tev tiks nosūtīta e-pasta ziņa ar apstiprinājuma saiti, kuru tev būs jānospiež, lai pabeigtu reģistrāciju.

Nospiežot "Apartamenti" galvenajā izvēlnē, atver lapu ar visiem pieejamajiem piedāvājumiem.

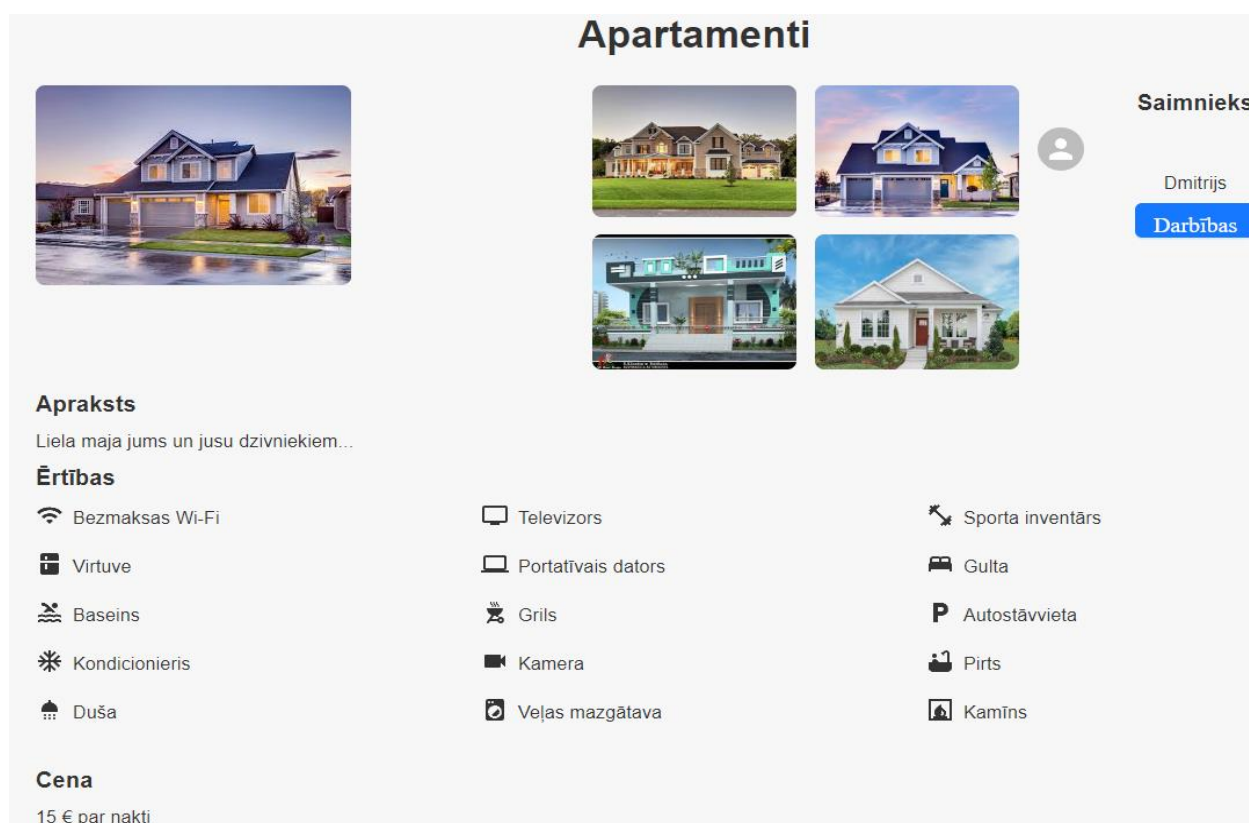


8.attels. Visi piedāvājumi

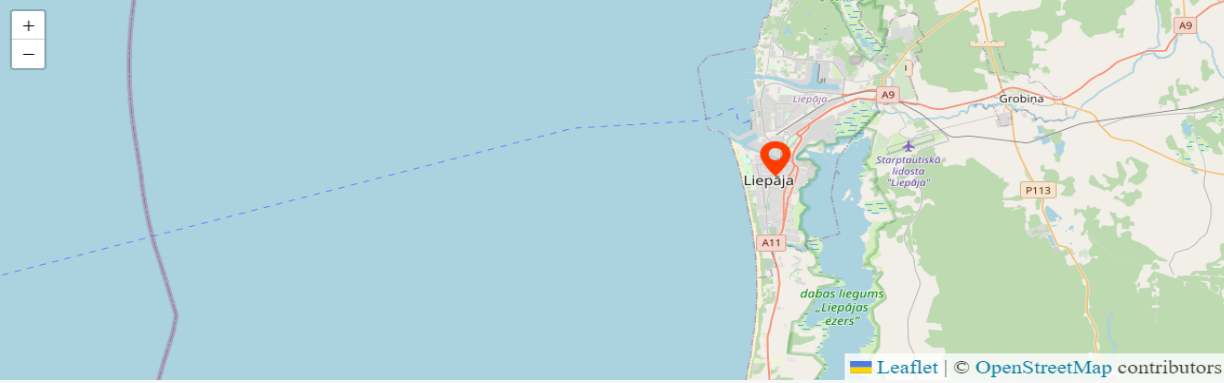
Tur vari pārlūkot sarakstu ar dažādiem piedāvājumiem, tostarp īres vietām vai pakalpojumiem.

Rezervācija:


Kad esi atradis interesējošo piedāvājumu, nospied uz viņu, lai iegūtu detalizētu informāciju.




9.attels. Apartamentu informācija



Atsauksmes



Jānis Bērziņš
 ★★★★★
 Lieliska vieta! Ļoti tīra un ērta. Noteikti ieteikšu draugiem.



Anna Kalniņa
 ★★★★★
 Viss bija perfekti! Ļoti jauka saimniece un brīnišķīga vieta.

10.attels. Karte

Atlasiet datumus, uz kuriem vēlaties iznomāt māju, un tiks parādīta cena par visām dienām.

◀

jūnijs

▼

2024

▶

pirmd.	otrd.	trešd.	ceturtd.	piektd.	sestd.	svētd.
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Kopējā cena

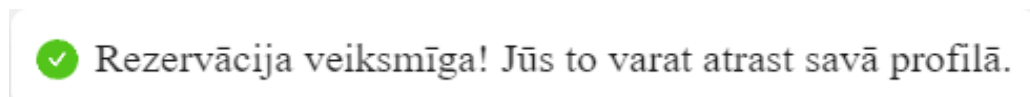
50.40 €

Atpakaļ

Rezervēt

11.attels. Registrācijas

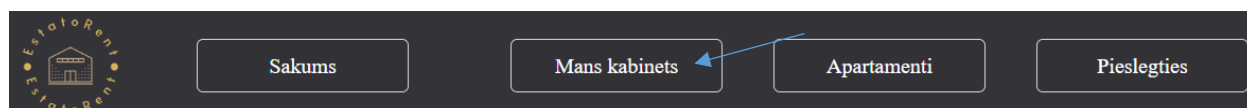
Jā esat gatavi rezervēt, noklikšķiniet uz rezervēt, ja vietnes kontā ir pietiekami daudz līdzekļu, jūs veiksmīgi rezervēsiet māju, un līdzekļi tiks noņemti no jūsu konta. pēc tam rezervēta māja būs redzama jūsu rezervāciju sarakstu tabulā.



12.attels. Veiksmīga rezervācijas paziņojums

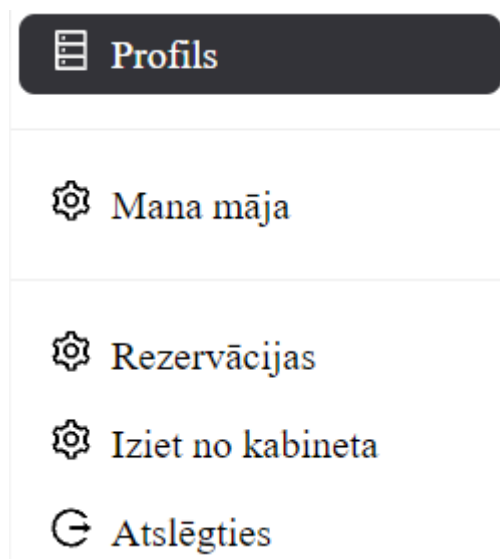
Maksājums tīmekļa vietnē un profila rediģēšana:

Lai papildinātu kontu vietnē, jums jādodas uz cilni mans kabinets.



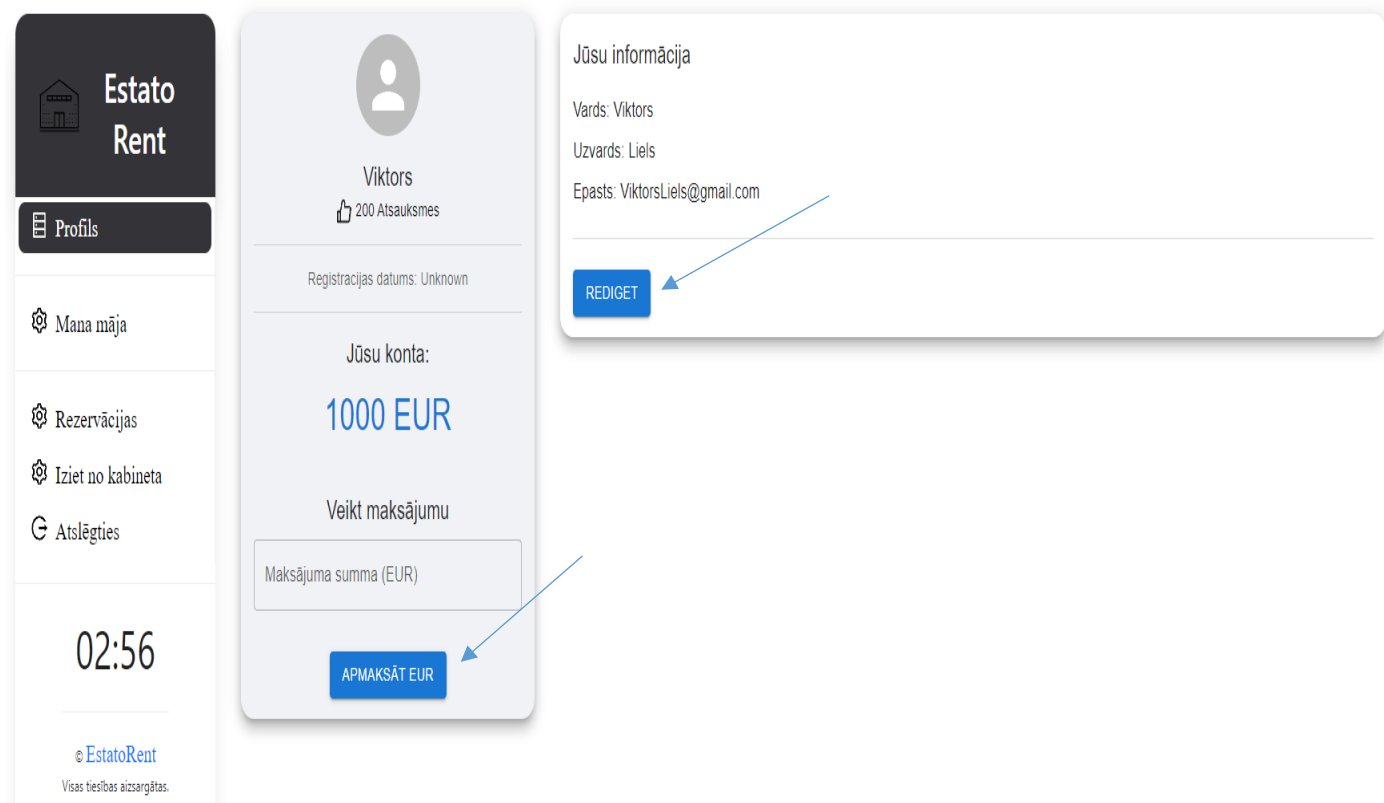
13.attels. Mans kabinets navigācija

Tad jaunā cilnē izvēlamies profila sadaļu.



14.attels. Profila navigācija

Šajā lapā mēs redzam savu profilu, kurā varam papildināt savu kontu un arī rediģēt savu profila informāciju pēc rediģēšanas pogas nospiešanas. Lai papildinātu kontu, noklikšķiniet uz pogas "Maksāt".



15.attels. Profils

pēc noklikšķināšanas uz maksāt tiks atvērta jauna lapa, kurā norādīsiet savas kartes informāciju, lai papildinātu kontu.

E-pasts ggamersshd@gmail.com

Ievadiet maksājuma informāciju **Atcelt**

1234 1234 1234 1234

MM / YY CVC

Kartes īpašnieka vārds

Lisa

Valsts vai reģions

Latvija

Turpinot, jūs piekrītat Link saglabāt savu maksājumu veidu.

Maksāt

16.attels. Apmaksas dati

Noklikšķinot uz pogas rediģēt, tiek atvērts logs, kurā redzama mūsu pašreizējā informācija, ko var rediģēt, kā arī augšupielādēt profila attēlu.


Rediģēt profilu

Vards

Viktors

Uzvards

Liels

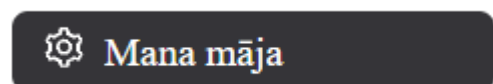
 AUGŠUPLĀDĒT JAUNU ATTĒLU

ATCELT SAGLABAT

17.attels. Profila rediģēšana

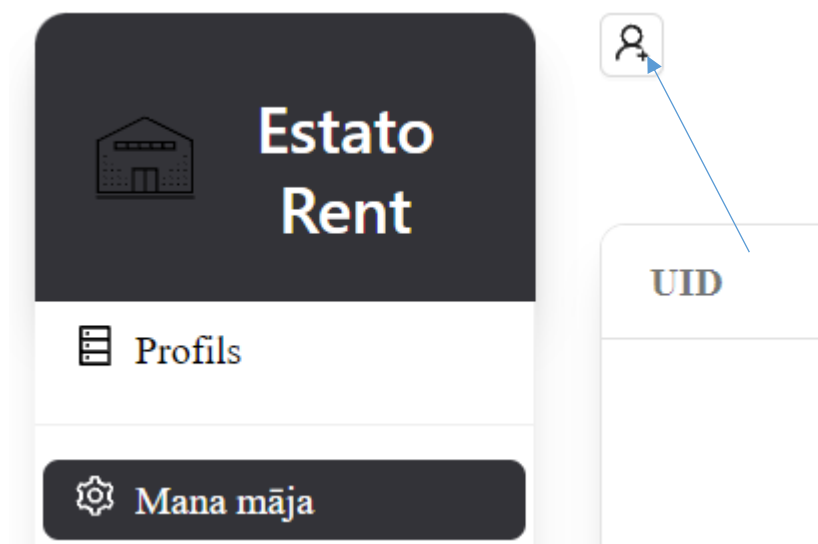
Mājas pievienošana:

lai pievienotu savu māju, panelī kreisajā pusē noklikšķiniet uz "Mana māja".



18.attels. Profila navigācija

pēc tam noklikšķiniet uz pogas ar plusa ikonu atvērtajā logā.



19.attels. Mājas navigācija

mums tiks atvērta lapa, kurā mums būs jāsniedz informācija par māju, ko mēs soli pa solim aizpildām.

1 Adreses dati — 2 Ērtības — 3 Apraksts — 4 Bides — 5 Lokacija

Piedāvājuma nosaukums

Ielas nosaukums un nummurs

Dzīvoklis, korpuss u.c.

Rīga LV-4092

















Cena

← ATPAKĀJ

TURPINĀT >

20.attels. Adreses dati

✓ Adreses dati — 2 Ērtības — 3 Apraksts — 4 Bīdes — 5 Lokacija

 WI-FI	 TELEVIZORS	 GULTA	 PORTATĪVAIS DATORS
 KONDICIONIER	 STĀVVIETA	 BASEINS	 VIRTUVE
 KAMERA	 VEĻAS MAZGĀTAVA	 GRILS	 SPORTA INVENTĀRS
 PIRTS	 DUŠA	 KAMINS	 MUZIKA

← ATPAKAĻ

TURPINĀT >

21.attels. Ērtību saraksts

✓ Adreses dati — ✓ Ērtības — 3 Apraksts — 4 Bīdes — 5 Lokacija

Īsi aprakstiet savu māju

Ievadiet aprakstu savai mājai...

Cilvēku skaits, kas ērti jutīsies jūsu mājā

Ievadiet cilvēku skaitu.

← ATPAKAĻ

TURPINĀT >

22.attels. Apraksts

✓ Adrešes dati — ✓ Ērtības — ✓ Apraksts — 4 Bides — 5 Lokacija

Pievienot fotoattēlus (minimum 3)

AUGŠPIELADET ATTELUS



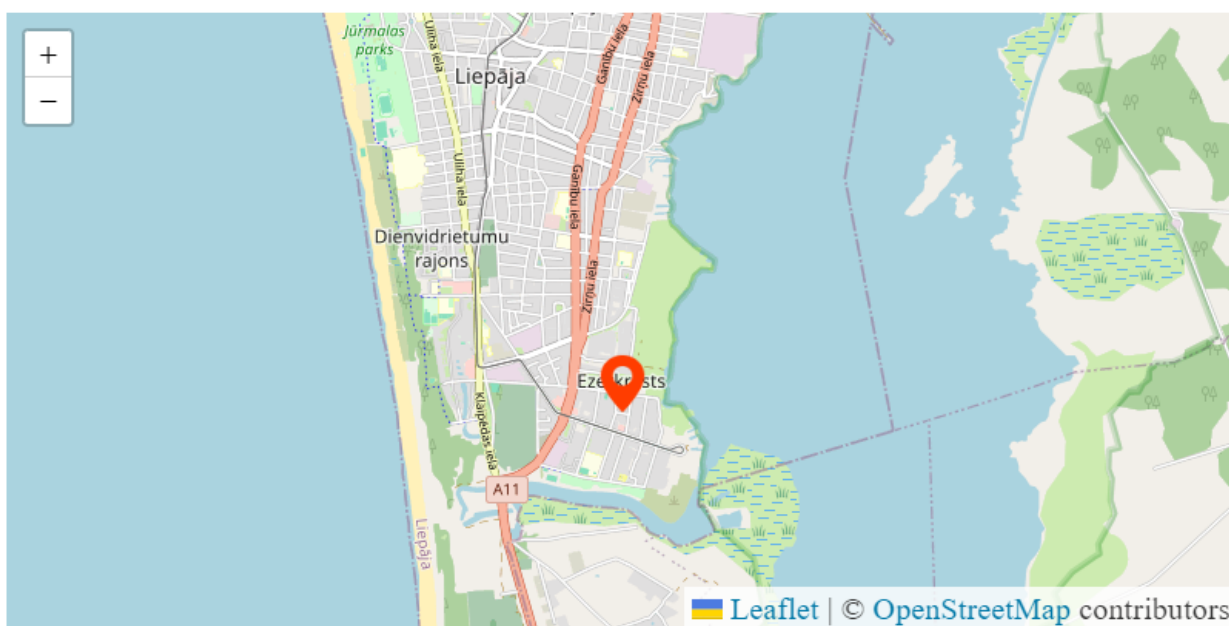
← ATPAKAĻ

TURPINĀT >

23.attels. Bides pievienošana

✓ Adrešes dati — ✓ Ērtības — ✓ Apraksts — ✓ Bides — 5 Lokacija

Izvēlieties atrašanās vietu kartē



← ATPAKAĻ

APSTIPRINĀT >

24.attels. Lokācijas pievienošana





Paldies!

Paldies, ka izmantojat mūsu pakalpojumus, jūsu piedāvājums ir nosūtīts administratoram pārbaudes nolukiem, un drīz to varēs redzēt arī citi lietotāji.

ATGRIEZTIES PROFILĀ

25.attels. Paziņojums par mājas pievienošanu

pēc visu datu aizpildīšanas piedāvājums tiek nosūtīts administratoram pārbaudei, un pēc tam, kad viņš to pārbaudīs un apstiprinās, lietotāji varēs apskatīt un iznomāt jūsu piedāvājumu, un jūsu māja parādīsies tabulā sadaļā "Mana māja", kur mēs varam izņemt piedāvājumu un rediģēt to pēc apstiprināšanas.

UID	Pilseta	Vards	Status
dp7LfVcXfkdaFeXP2LDm	dwa	dwa	Aizsūtīts  

26.attels. Piedāvājuma zivādīšana tabulā

Lietotāju reģistrs (administrators skats)


Lapā "Mans kabinets" navigācijas joslā kreisajā pusē noklikšķiniet uz "Lietotāju reģistrs".





Lietotāju reģistrs

27.attels. Lietotāju reģistra navigācija

Šajā lapā mēs redzam visu vietnē esošo lietotāju sarakstu, no šīs lapas mēs varam pievienot administratoru, deaktivizēt lietotāju un rediģēt viņa datus.

 **Pievienot adminu**

UID	Vārds	Uzvārds	E-pasts	Loma	Statuss	
zaawUbfWHuZN...	Anatolijs	Kap	Mihail@Mihail.com	Admin	Aktīvs	 

28.attels. Lietotāja reģistrs

Lai pievienotu administratoru, augšpusē noklikšķiniet uz pogas "Pievienot adminu", un atvērsies logs, kurā ievadīsim datus.

Jauns administratora kots

×

* Vārds

* Uzvārds

* E-pasts

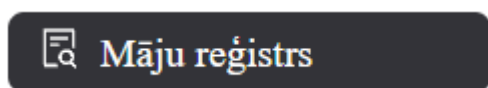
* Parole

PIEVIENOT Adminu

29.attels. Administratora pievienošana

Īpašumu apskate un moderēšana (administratora skats):

Lapā "Mans kabinets" noklikšķiniet uz "Māju reģistrs" navigācijas joslā lapas kreisajā pusē.



30.attels. Māju reģistra navigācija

Tālāk lapā redzams visu māju saraksts.

UID	Pilseta	Vards	Saimnieks	Loma	Uztaisīts	
dp7LfVcXfkdaFe...	dwa	dwa	IvSpJ1IguJgwnV...		Timestamp(second...	<div> <div></div> <div></div> <div></div> </div>
Y6RosQtu72ZTzi...	dwadwa	Elionora house	8jkMvQ3Hq7fCk...		Timestamp(second...	<div> <div></div> <div></div> <div></div> </div>
U76LqxKf4co3rbf...	Riga	Maja visiem	8jkMvQ3Hq7fCk...		Timestamp(second...	<div> <div></div> <div></div> <div></div> </div>
2U8mbJ4x6CcR4...	Liepaja	Apartamenti	8jkMvQ3Hq7fCk...		Timestamp(second...	<div> <div></div> <div></div> <div></div> </div>

31.attels. Māju reģistrs

Lai apskatītu piedāvājumu, noklikšķiniet uz acs ikonas

Piedāvājuma informācija



Nosaukums: dwa

Pilseta: dwa

Saimnieks: IvSpJ1IguJgwnVWXoSnEuF7ZFFF2

Status: Aizsūtīts



Apraksts: dwa

Atpakaļ

32.attels. Piedāvājuma apskate

Noklikšķinot uz ķeksīša, mēs apstiprināsim piedāvājuma izvietošanu visu dzīvokļu lapā,
noklikšķinot uz ķeksīša, mēs noraidīsim piedāvājumu, kas nozīmē, ka citi lietotāji to neredzēs
kopējā dzīvokļu lapā.

6. Testēšanas dokumentācija

Ši sadaļa ir informācija par testešanu, testēšanas metode, rīkiem, kā arī testpimēri ka arī testēšanas žurnāls

6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums

Black box testēšana ir viens no programmēšanas pārbaudes veidiem, kurā testētāji pārbauda programmas darbību bez detalizētas iekšējās kodēšanas zināšanām. Šajā metodē testētāji pievērš uzmanību programmas specifikācijai un funkcionalitātei, nevis kodam, kas to realizē.

Testētāji izmanto programmas interfeisu un ievades datus, lai pārbaudītu, vai programma darbojas pareizi saskaņā ar noteiktajām prasībām. Viņi veic dažādas testēšanas darbības, piemēram, ievada vērtību pārbaudi, darbības secību testēšanu un izņēmuma apstrādi.

Šīs metodes priekšrocība ir tā, ka tā ļauj testēt programmu no lietotāja perspektīvas, kas palīdz identificēt iespējamās problēmas un neparedzētas sekas. Turklāt, tā ļauj testētājiem koncentrēties uz programmas darbību, nevis kodēšanas detaļām, tādējādi palielinot testēšanas efektivitāti un atklājot plašāku klāstu potenciālo kļūdu.

6.2. Testpiemēru kopa

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra apraksts	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra sagaidāmais rezultāts	Prasības ID
Black Box							
TP.REG.01	Reģistreties sistemā	Noklikšķiniet uz pogas pieslēgties vietnes navigācijā, pēc tam atvērsies lapa, kurā jūs izvēlaties reģistrāciju, ievadiet savu vārdu, uzvārdu, e-pastu, paroli.	Lietotāja reģistrācijas datu saglabāšana datubāzē	1)Atveriet autorizācijas un reģistrācijas lapu 2)Ar peles kreiso pogu noklikšķiniet uz reģistrēties 3)Ievadiet savu vārdu 4)Ievadiet savu uzvārdu 5)Ievadiet savu e-pastu 6)Ievadiet savu paroli 7) Kreisās peles pogas klikšķis uz pogu "REGISTRETIES"	Vārds: [Vārds] Uzvārds: [Uzvārds] E-pasts: [E-pasts] Parole: [Parole] Peles kreisā poga: [Kreisās peles pogas klikšķis]	Tiek parādīts ziņojums par veiksmīgu reģistrāciju, dati tiek saglabāti datu bāzē.	PR.01
TP.AUTH.01	Pieslēgties sistemā	Lietotājs atrodas autorizācijas un reģistrācijas lapā, lietotājam ir izveidots konts.	Pārbaudīt lietotāja iespēju pieslēgties sistēmai, ievadot derīgu	Atveriet autorizācijas un reģistrācijas lapu. Ar peles kreiso pogu noklikšķiniet uz pieslēgties. Ievadiet savu e-pastu. Ievadiet savu paroli.	E-pasts: [Ievadiet savu e-pastu] Parole: [Ievadiet savu paroli]	Tiek parādīts ziņojums par veiksmīgu pieslēgšanos, lietotājs tiek novirzīts uz galveno lapu.	PR.02

			e-pastu un paroli.	Kreisās peles pogas klikšķis uz pogu "PIESLĒGTIES".	Peles kreisā poga: [Kreisās peles pogas klikšķis]		
TP.PROF.01	Ieeja lietotāja profilā	Lietotājs ir pieslēdzies sistēmai, lietotājs atrodas vietnes galvenajā lapā.	Pārbaudīt lietotāja iespēju piekļūt savam profilam pēc pieslēgšanās sistēmai.	Atveriet autorizācijas un reģistrācijas lapu. Ar peles kreiso pogu noklikšķiniet uz "Pieslēgties". Ievadiet savu e-pastu. Ievadiet savu paroli. Kreisās peles pogas klikšķis uz pogu "PIESLĒGTIES". Galvenās lapas headerī ar peles kreiso pogu noklikšķiniet uz "Mans profils"	Peles kreisā poga: [Kreisās peles pogas klikšķis]	Tiek parādīta lietotāja profila lapa, kurā redzami lietotāja dati.	PR.01
TP.PROF.02	Rediģēt lietotāja profilu	Lietotājs ir pieslēdzies sistēmai un atrodas lietotāja profila lapā.	Pārbaudīt lietotāja iespēju rediģēt savu profilu, ieskaitot vārda, uzvārda maiņu	Atveriet autorizācijas un reģistrācijas lapu. Ar peles kreiso pogu noklikšķiniet uz "Pieslēgties". Ievadiet savu e-pastu. Ievadiet savu paroli.	Jaunais vārds: [Ievadiet savu jauno vārdu] Jaunais uzvārds: [Ievadiet savu jauno uzvārdu]	Tiek parādīts ziņojums par veiksmīgu profila atjaunināšanu, un lietotāja dati, ieskaitot attēlu, tiek atjaunināti profila lapā.	PR. 20

			<p>un profila attēla augšupielādi.</p>	<p>Kreisās peles pogas klikšķis uz pogu "PIESLĒGTIES". Galvenās lapas headerī ar peles kreiso pogu noklikšķiniet uz "Mans profils".</p> <p>Profila lapā ar peles kreiso pogu noklikšķiniet uz "Rediģēt".</p> <p>Ievadiet savu jauno vārdu.</p> <p>Ievadiet savu jauno uzvārdu.</p> <p>Noklikšķiniet uz pogas "Augšupielādēt attēlu" un izvēlieties attēlu.</p> <p>Ar peles kreiso pogu noklikšķiniet uz pogas "Saglabāt".</p>	<p>Profila attēls: [Izvēlieties attēlu]</p> <p>Peles kreisā poga: [Kreisās peles pogas klikšķis]</p>		
TP.PROF.03	Konta papildināšana	Lietotājs ir pieslēdzies sistēmai un atrodas savā lietotāja profilā.	<p>Pārbaudīt lietotāja iespēju veikt papildinājumu savā kontā, izmantojot maksājumu sistēmu Stripe,</p>	<p>1)Nospiediet pogu "Papildināt kontu" savā profilā un ievadiet summu, kādu vēlaties pievienot.</p> <p>2)Nospiediet pogu "Apmaksāt".</p> <p>3)Tiks atvērta Stripe loga, kurā jāievada kartes dati.</p>	<p>Papildināmā summa: [Ievadiet summu]</p> <p>Kartes dati: [Ievadiet derīgus kartes datus]</p>	<p>Maksājums ir veiksmīgi veikts, un papildinātā summa ir pareizi atspoguļota lietotāja kontā.</p>	PR. 04

			un pārliecināties, ka summa tiek pareizi uzskaitīta lietotāja kontā.	4) Kad dati tiks ievadīti pareizi, maksājums tiks veikts. 5) Pārliecinieties, vai papildinātā summa tiek pareizi atspoguļota lietotāja kontā.			
TP.HOME.01	Parādīt dati lapā "Mana māja"	Lietotājam jābūt autorizētam un jāatrodas uz lapas "Mana māja".	Pārbaudīt, vai dati, kas tiek iegūti no datu bāzes, tiek pareizi attēloti uz lapas "Mana māja", izmantojot AG Grid vai līdzīgu saraksta vai režģa komponentu.	1) Pārliecinieties, vai esat autorizējies sistēmā un esat sasniedzis lapu "Mana māja". 2) Pārliecinieties, vai uz šīs lapas tiek attēloti dati no datu bāzes, izmantojot AG Grid 3) Pārliecinieties, vai dati tiek pareizi attēloti, tostarp visi nepieciešamie lauki un to vērtības.	Nav nepieciešami, ja dati tiek iegūti no datu bāzes automātiski.	Lapā "Mana māja" tiek attēloti dati no datu bāzes, un tie tiek pareizi attēloti saraksta, AG Grid, kas ļauj lietotājam viegli pārskatīt un pārvaldīt šos datus.	PR.12
TP.HOME.02	Mainīt piedāvājuma statusu uz slēpto, nospiežot pogu ar atzīmi	Lietotājam jābūt autorizētam un jāatrodas uz lapas "Mana māja".	Pārbaudīt, vai piedāvājuma statuss mainās uz slēpto, kad tiek	1) Atrodiet piedāvājumu, kam jāmaina statuss. 2) Pārliecinieties, ka piedāvājums sākotnēji ir redzams un statuss nav slēpts.	Piedāvājuma ID vai nosaukums, kuram jāmaina statuss: [Ievadiet	Piedāvājuma statuss ir veiksmīgi mainīts uz slēpto, kad tiek noklikšķināts uz pogas ar ikonu " ķeksītis ", neizmantojot pārlādi vai citas lapas pārlādes darbības.	PR.10

			noklikšķināts uz pogas ar atzīmi.	3) Noklikšķiniet uz pogas ar ikonu "ķeksītis", lai mainītu statusu. 4) Pārliecinieties, ka statuss tagad ir mainījies uz slēpto, neizmantojot pārlādi vai citas lapas pārlādes darbības.	piedāvājuma ID vai nosaukumu]		
TP.HOME.03	Rediģēt piedāvājumu lapā "Mana māja"	Lietotājam jābūt autorizētam un jāatrodas lapā "Mana māja".	Pārbaudīt, vai lietotājs var veiksmīgi rediģēt savu piedāvājumu, izmantojot iespējas lapā "Mans mājas".	1) Atveriet lapu " Mana māja ". 2) Atrodiet savu piedāvājumu, ko vēlaties rediģēt. 3) Noklikšķiniet uz pogas vai saites, kas paredzēta rediģēšanai. 4) Rediģējiet nepieciešamos laukus, piemēram, aprakstu, cenu vai citus datus. 5) Saglabāji izmaiņas, noklikšķinot uz pogas "Saglabāt" vai līdzīgu pogu. 6) Pārliecinieties, ka izmaiņas ir veiksmīgi saglabātas un pareizi atspoguļotas.	Piedāvājuma ID vai nosaukums, kuram jāmaina statuss: [Ievadiet piedāvājuma ID vai nosaukumu].	Piedāvājuma statuss ir veiksmīgi mainīts uz slēpto, kad tiek noklikšķināts uz pogas ar ikonu "ķeksītis " un veiktas izmaiņas modālajā logā	PR.13

TP.HOME.04	<p>Pievienot māju, nospiežot pogu ar ikonu pluss</p>	<p>Lietotājs jau ir autorizējies un atrodas lapā "Mana māja".</p>	<p>Pārbaudīt, vai, nospiežot pogu ar ikonu pluss, tiek atvērta lapa, kurā lietotājs var pievienot jaunu māju un ievadīt visus nepieciešamos datus.</p>	<p>1) Nospiežiet pogu ar ikonu pluss, lai atvērtu lapu, kurā pievienot jaunu māju. 2) Ievadiet mājas nosaukumu, adresi, adresi 2, mājas numuru, pilsētu, pasta indeksu un cenu. 3) Izvēlieties mājas ērtības, nospiežot pogas ar atbilstošajām ikonām. 4) Ievadiet mājas aprakstu un norādiet, cik cilvēkiem tā ir paredzēta. 5) Augšupielādējiet vismaz piecas mājas attēlus. 6) Izvēlieties mājas ģeolokāciju kartē, nospiežot kreiso peles pogu. 7) Nospiežiet pogu "Apstiprināt", lai dati tiktu nosūtīti uz datu bāzi un piedāvājums tiktu iesniegts administratora pārskatam.</p>	<p>Mājas nosaukums: [Ievadiet mājas nosaukumu] Adrese: [Ievadiet mājas adresi] Adrese 2: [Ievadiet papildu adresi, ja nepieciešams] Mājas numurs: [Ievadiet mājas numuru] Pilsēta: [Ievadiet pilsētu] Pasta indekss: [Ievadiet pasta indeksu] Cena: [Ievadiet mājas cenu] Ērtības: [Izvēlieties ērtības, izmantojot ikonas] Apraksts: [Ievadiet mājas aprakstu] Cilvēku skaits: [Norādiet, cik</p>	<p>Pēc pogas "Apstiprināt" nospiešanas dati tiek veiksmīgi nosūtīti uz datu bāzi, un mājas piedāvājums tiek iesniegts administratora pārskatam.</p>	TR.11
------------	--	---	--	---	--	---	-------

					cilvēkiem māja ir paredzēta]		
TP.USERLIS T.01	Parādīt visus lietotājus sarakstā	Lietotājam jābūt autorizētam kā administrators un jāatrodas lapā ar lietotāju sarakstu.	Pārbaudīt, vai lapā "Lietotāju saraksts" tiek parādīti visi pieejamie lietotāji tabulārā formātā.	1) Atveriet lapu "Lietotāju saraksts". 2) Pārlicinieties, ka visi pieejamie lietotāji tiek parādīti tabulārā formātā.	Visi lietotāji ir parādīti sarakstā tabulārā formātā.	Mašīnu attēla objekts sāks veikt rotāciju pulksteņrādītāja virzienam	PR.USERLIS T.01
TP.USERLIS T.02	Rediģēt lietotāju informāciju	Lietotājam jābūt autorizētam kā administrators un jāatrodas lapā ar lietotāju sarakstu.	Pārbaudīt, vai, kārtotošanas ikonai, tiek atvērts modālais logs, kurā rediģējama lietotāja informācija	1) Atrodiet lietotāju, kuru vēlaties rediģēt. 2) Nospiediet pogu ar kārtotošanas ikonu, lai atvērtu modālo logu ar šī lietotāja informāciju. 3) Rediģējiet nepieciešamos laukus. 4) Pārlicinieties, ka rediģētie dati ir pareizi atspoguļoti modālajā logā. 5) Apstipriniet izmaiņas un saglabājiet tās.	Izmaiņas lietotāja informācijā tiek veiksmīgi saglabātas.	Mašīnu attēla objekts sāks veikt rotāciju pretēji pulksteņrādītāja virzienam	PR.USERLIS T.02

TP.USERLIS T.03	Deaktivizēt lietotāju	Lietotājam jābūt autorizētam kā administrators un jāatrodas lapā ar lietotāju sarakstu.	Pārbaudīt, vai nospiežot pogu "Deaktivizēt", lietotāja statuss mainās uz "Deaktivizēts"	1) Atrodiet lietotāju, kuru vēlaties deaktivizēt. 2) Nospiediet pogu "Deaktivizēt". 3) Pārlicinieties, ka lietotāja statuss ir mainījies uz "Deaktivizēts".	Lietotāja statuss ir veiksmīgi mainīts uz "Deaktivizēts".	Attēla objekta platums būs palielināts līdz kamēr būs sasniegts tā limits	PR.03
TP.USERLIS T.04	Pievienot jaunu administratoru	Lietotājam jābūt autorizētam kā administrators un jāatrodas lapā "Pievienot administratoru".	Pārbaudīt, vai nospiežot pogu "Pievienot adminu", lai atvērtu modālo logu. 3) Ievadiet jaunā administratora vārdu, atvērts modālais logs, kurā lietotājs var ievadīt jaunā administratora informāciju.	1) Atveriet lapu "Lietotāju reģistrācija". 2) Nospiediet pogu "Pievienot adminu", lai atvērtu modālo logu. 3) Ievadiet jaunā administratora vārdu, uzvārdu, e-pastu un paroli. 4) Pārlicinieties, ka jaunā administratora informācija ir ievadīta pareizi. 5) Apstipriniet jaunā administratora pievienošanu, nospiežot pogu "Saglabāt".	Vārds: [Ievadiet vārdu] Uzvārds: [Ievadiet uzvārdu] E-pasts: [Ievadiet e-pastu] Parole: [Ievadiet paroli]	Jaunais administrators ir veiksmīgi pievienots pēc pogas "Saglabāt" nospiešanas.	PR.USERLIS T.01
TP.HOUSELI ST.01	Parādīt visus mājas piedāvājumus sarakstā	Administratoram jābūt autorizētam un jāatrodas lapā "Māju reģistrs".	Pārbaudīt, vai audīt, vai lapā "Māju reģistrs" tiek parādīti	1) Atveriet lapu "Māju reģistrs".		Visi māju piedāvājumi ir parādīti sarakstā tabulārā formātā.	PR.12

			visi pieejamie māju piedāvājumi tabulārā formātā.	2) Pārlicinieties, ka visi pieejamie māju piedāvājumi tiek parādīti tabulārā formātā.			
TP.HOUSELI ST.02	Skatīt mājas piedāvājumu detalizētu informāciju	Administratorm jābūt autorizētam un jāatrodas lapā "Māju reģistrs".	Pārbaudīt, vai, nospiežot ikonu ar aci, tiek atvērts detalizēts skats ar izvēlēto mājas piedāvājuma informāciju	1) Atrodiet māju piedāvājumu, uz kuru vēlaties skatīties detalizētu informāciju. 2) Nospiediet pogu ar ikonu "acis", lai atvērtu detalizētu skatu.	Nav nepieciešami, jo testējam tikai datu izvadi, nevis ievadi.	Detalizēta informācija par izvēlēto mājas piedāvājumu ir veiksmīgi parādīta.	PR.11
TP.HOUSELI ST.03	Rediģēt mājas piedāvājumu informāciju	Administratorm jābūt autorizētam un jāatrodas lapā "Māju reģistrs".	Pārbaudīt, vai, nospiežot ikonu ar "zimuļa" ikonu, tiek atvērts modālais logs, kurā var rediģēt izvēlēto mājas piedāvājuma informāciju.	1) Atrodiet māju piedāvājumu, kuru vēlaties rediģēt. 2) Nospiediet pogu ar kārtošanas "zimuļi", lai atvērtu modālo logu ar izvēlēto mājas piedāvājuma informāciju. 3) Rediģējiet nepieciešamos laukus.	Pilseta: Nosaukums Apraksts Bildes	Datubāzēs dati tiek rediģēti un atjaunināti attēloti.	PR.13

				4) Pārlicinieties, ka rediģētie dati ir pareizi atspoguļoti modālajā logā. 5) Apstipriniet izmaiņas un saglabāiet tās.			
--	--	--	--	---	--	--	--



6.3. Testēšanas žurnāls

Testēšanas ID	Datums	Testpiemēra ID	Testpiemēra nosaukums	Testētājs	Statuss	Kļūdas ziņojums	Kļūdas ziņojuma Nr.
Black Box							
TZ.REG.01	10.06.2024	TP.STA.01	Reģistreties sistemā	Dmitrijs Geraščenko	Veiksmīgs		
TZ.AUTH.0	10.06.2024	TP.STA.01	Pieslēgties sistemai	Dmitrijs Geraščenko	Veiksmīgs		
TZ.PROF.01	10.06.2024	TP.PROF.01	Ieeja lietotāja profilā	Dmitrijs Geraščenko	Veiksmīgs		
TZ.PROF.02	10.06.2024	TP.PROF.02	Rediģēt lietotāja profilu	Dmitrijs Geraščenko	Veiksmīgs		
TZ.PROF.03	10.06.2024	TP.PROF.03	Konta papildināšana	Dmitrijs Geraščenko	Veiksmīgs		

TZ.HOME.01	10.06.2024	TP.HOME.01	Parādīt dati lapā "Mana māja"	Dmitrijs Geraščenko	Veiksmīgs		
TZ.HOME.02	10.06.2024	TP.HOME.02	Mainīt piedāvājuma statusu uz slēpto, nospiežot pogu ar atzīmi	Dmitrijs Geraščenko	Veiksmīgs		
TZ.HOME.03	10.06.2024	TP.HOME.03	Rediģēt piedāvājumu lapā "Mana māja"	Dmitrijs Geraščenko	Veiksmīgs		
TZ.HOME.04	10.06.2024	TP.HOME.04	Pievienot māju, nospiežot pogu ar ikonu pluss	Dmitrijs Geraščenko	Veiksmīgs		
TZ.USERLIS T.01	10.06.2024	TP.USERLIS T.01	Parādīt visus lietotājus sarakstā	Dmitrijs Geraščenko	Veiksmīgs		
TZ.USERLIS T.02	10.06.2024	TP.USERLIS T.02	Rediģēt lietotāju informāciju	Dmitrijs Geraščenko	Veiksmīgs		
TZ.USERLIS T.03	10.06.2024	TP.USERLIS T.03	Deaktivizēt lietotāju	Dmitrijs Geraščenko	Veiksmīgs		
TZ.USERLIS T.04	10.06.2024	TP.USERLIS T.04	Pievienot jaunu administratoru	Dmitrijs Geraščenko	Veiksmīgs		
TZ.HOUSELI ST.01	10.06.2024	TP.HOUSELI ST.01	Parādīt visus mājas piedāvājumus sarakstā	Dmitrijs Geraščenko	Veiksmīgs		
TZ.HOUSELI ST.02	10.06.2024	TP.HOUSELI ST.02	Skatīt mājas piedāvājumu detalizētu informāciju	Dmitrijs Geraščenko	Veiksmīgs		

TZ.HOUSELI ST.03	10.06.2024	TP.HOUSELI ST.03	Rediģēt mājas piedāvājumu informāciju	Dmitrijs Geraščenko	Veiksmīgs		
---------------------	------------	---------------------	--	------------------------	-----------	--	--

7. Secinājumi

Izstrādē šī projekta laikā esmu ieguvis daudz jaunu prasmju un zināšanu programmēšanas jomā, it īpaši koncentrējoties uz tīmekļa izstrādi. Man bija iespēja dziļāk iepazīt un strādāt ar dažādām tehnoloģijām, un viena no tām bija React.js - moderna un elastīga tīmekļa bibliotēka.

Viena no galvenajām lietām, ko esmu iemācījies, ir darbs ar React.js un izstrāde dinamiskām un responsīvām tīmekļa lietojumprogrammām. React.js piedāvā plašas iespējas komponentu orientētai izstrādei, kas ļauj veidot atkārtoti izmantojamus un viegli uzturējamus kodu. Tāpat esmu iemācījies efektīvi izmantot React.js stāvokļa pārvaldību, kas ir būtiska daudzu modernu tīmekļa lietojumprogrammu izstrādē.

Turklāt esmu iepazinies ar NoSQL datu bāzu tehnoloģijām, kas ir svarīga sastāvdaļa modernā tīmekļa izstrādē. Šīs tehnoloģijas ļauj efektīvi glabāt un pārvaldīt strukturētus datus, nodrošinot elastīgumu un skalējamību. Man bija iespēja izmēģināt dažādas NoSQL datu bāzu platformas un pielietojumus, kas papildināja manas izstrādes zināšanas un prasmes.

Kopumā šī projekta izstrāde man sniedza iespēju iegūt praktisku pieredzi un attīstīt savas prasmes tīmekļa izstrādē. Es esmu pateicīgs par šo iespēju un esmu gatavs turpināt mācīties un pilnveidoties šajā jomā.

8. Lietoto terminu un saīsinājumu skaidrojumi

9. Literatūras un informācijas avotu saraksts

1. <https://react.dev/learn> 5.06.2024
2. <https://ant.design/components/overview/> 10.06.2024
3. <https://mui.com/material-ui/all-components/> 10.06.2024
4. <https://stripe.com/en-lv> 10.06.2024
5. <https://console.firebase.google.com/?hl=ru> 10.06.2024
6. <https://firebase.google.com/docs> 10.08.2024
7. <https://www.w3schools.com/react/> 1.07.2024

PIELIKUMS

```

31 // Definē soļus Stepper komponentei
32 const steps = ["Adrešes dati", "Ērtības", "Apraksts", "Bildes", "Lokācija"];
33
34 // Funkcija, kas atgriež saturu atkarībā no pašreizējā soli
35 function getStepContent(
36   step,
37   handleFormChange,
38   formData,
39   handleAprakstsChange,
40   aprakstsData,
41   handleAmenitiesChange,
42   selectedAmenities,
43   handleImagesChange,
44   handleLocationChange
45 ) {
46   switch (step) {
47     case 0:
48       return (
49         <AddressForm handleFormChange={handleFormChange} formData={formData} />
50       );
51     case 1:
52       return (
53         <Ertibas
54           onAmenitiesChange={handleAmenitiesChange}
55           selectedAmenities={selectedAmenities}
56         />
57       );
58     case 2:
59       return (
60         <Apraksts
61           handleAprakstsChange={handleAprakstsChange}
62           aprakstsData={aprakstsData}
63         />
64       );
65     case 3:
66       return <Bildes onImagesChange={handleImagesChange} />;
67     case 4:
68       return <Lokacija handleLocationChange={handleLocationChange} />;
69     default:
70       return (
71         <Result
72           status="404"
73           title="404"
74           subTitle="Atvainojiet, apmeklētā lapa neeksistē."
75           extra={<Button type="primary">Atpakaļ uz sākumlapu</Button>}
76         />

```

33.attels. Māju pievinošanas komponente

```

76     />
77   );
78 }
79 }
80
81 // Funkcija, kas augšupielādē attēlus uz krātuvi
82 const uploadImages = async (files) => {
83   const storage = getStorage();
84   const urls = await Promise.all(
85     files.map(async (file) => {
86       const storageRef = ref(storage, `images/${file.name}`);
87       const uploadTask = await uploadBytesResumable(storageRef, file);
88       const downloadURL = await getDownloadURL(uploadTask.ref);
89       return downloadURL;
90     })
91   );
92   return urls;
93 };
94
95 // Galvenā komponente
96 export default function Checkout() {
97   // Valsts mainīgie
98   const [activeStep, setActiveStep] = useState(0);
99   const [selectedAmenities, setSelectedAmenities] = useState([]);
100   const [messageApi, contextHolder] = message.useMessage();
101   const [formData, setFormData] = useState({
102     bookName: "",
103     address1: "",
104     cena: "",
105     address2: "",
106     city: "",
107     zip: "",
108   });
109   const [aprakstsData, setAprakstsData] = useState({
110     Apraksts1: "",
111     Nummurs: "",
112   });
113   const [imageFiles, setImageFiles] = useState([]);
114   const [imageUrls, setImageUrls] = useState([]);
115   const [locationData, setLocationData] = useState(null);
116
117   // Brīdinājuma funkcija
118   const warning = () => {
119     messageApi.open({
120       type: "warning",
121       content: "Lūdzu, aizpildiet visus laukus!",
122     });
123   };

```

34.attels. Māju pievinošanas komponente

```

122 | });
123 | };
124 |
125 | // Funkcija, kas pievieno mājas datus Firestore
126 | const adresFetch = async () => {
127 |   const auth = getAuth();
128 |   const uid = auth.currentUser ? auth.currentUser.uid : null;
129 |
130 |   if (!uid) {
131 |     console.error("Lietotājs nav autentificējies");
132 |     return;
133 |   }
134 |
135 |   try {
136 |     const uploadedImageUrls = await uploadImages(imageFiles);
137 |     const houseData = {
138 |       Name: formData.bookName,
139 |       Address1: formData.address1,
140 |       Address2: formData.address2,
141 |       Price: formData.cena,
142 |       City: formData.city,
143 |       Zip: formData.zip,
144 |       Status: "Aizsūtīts",
145 |       Description: aprakstsData.Apraksts1,
146 |       PeopleCount: aprakstsData.Nummurs,
147 |       Amenities: selectedAmenities,
148 |       Images: uploadedImageUrls,
149 |       Location: locationData,
150 |       createdAt: new Date(),
151 |       updatedAt: new Date(),
152 |       Owner: uid,
153 |     };
154 |
155 |     const userDocRef = collection(db, "Mājas");
156 |     await addDoc(userDocRef, houseData);
157 |
158 |     console.log("Dati veiksmīgi nosūtīti uz Firestore!");
159 |     setActiveStep(steps.length);
160 |   } catch (error) {
161 |     console.error("Kļūda sūtīt datus uz Firestore:", error);
162 |     throw error;
163 |   }
164 | };
165 |
166 | // Funkcija, kas apstrādā apraksta datu izmaiņas
167 | const handleAprakstsChange = (name, value) => {

```

35.attels. Māju pievienošanas komponente


```

166 | // Funkcija, kas apstrādā apraksta datu izmaiņas
167 | const handleAprakstsChange = (name, value) => {
168 |   setAprakstsData({
169 |     ...aprakstsData,
170 |     [name]: value,
171 |   });
172 | };
173 |
174 | // Funkcija, kas apstrādā formas datu izmaiņas
175 | const handleFormChange = (name, value) => {
176 |   setFormData({
177 |     ...formData,
178 |     [name]: value,
179 |   });
180 | };
181 |
182 | // Funkcija, kas apstrādā izmaiņas ērtībās
183 | const handleAmenitiesChange = (newAmenities) => {
184 |   setSelectedAmenities(newAmenities);
185 | };
186 |
187 | // Funkcija, kas apstrādā izmaiņas attēlos
188 | const handleImagesChange = (files) => {
189 |   setImageFiles(files);
190 | };
191 |
192 | // Funkcija, kas apstrādā izmaiņas lokācijā
193 | const handleLocationChange = (location) => {
194 |   const { lat, lng } = location;
195 |   const geoPoint = new GeoPoint(lat, lng);
196 |   setLocationData(geoPoint);
197 | };
198 |
199 | // Funkcija, kas validē soli
200 | const validateStep = (step) => {
201 |   switch (step) {
202 |     case 0:
203 |       return validateAddressForm();
204 |     case 1:
205 |       return validateErtibas();
206 |     case 2:
207 |       return validateApraksts();
208 |     case 3:
209 |       return validateImages();
210 |     case 4:
211 |       return validateLocation();
212 |     default:

```

36.attels. Māju pievinošanas komponente

```

213 |         return true;
214 |     }
215 | };
216 |
217 | // Funkcija, kas validē adreses formu
218 | const validateAddressForm = () => {
219 |     const { bookName, address1, address2, city, zip, cena } = formData;
220 |     return bookName && address1 && address2 && city && zip && cena;
221 | };
222 |
223 | // Funkcija, kas validē ērtības
224 | const validateErtibas = () => {
225 |     return selectedAmenities.length > 0;
226 | };
227 |
228 | // Funkcija, kas validē aprakstu
229 | const validateApraksts = () => {
230 |     const { Apraksts1, Nummurs } = aprakstsData;
231 |     return Apraksts1 && Nummurs;
232 | };
233 |
234 | // Funkcija, kas validē attēlus
235 | const validateImages = () => {
236 |     return imageFiles.length >= 3;
237 | };
238 |
239 | // Funkcija, kas validē lokāciju
240 | const validateLocation = () => {
241 |     return locationData !== null;
242 | };
243 |
244 | // Funkcija, kas pāreja uz nākamo soli
245 | const handleNext = () => {
246 |     if (validateStep(activeStep)) {
247 |         if (activeStep === steps.length - 1) {
248 |             adresFetch();
249 |         } else {
250 |             setActiveStep((prevActiveStep) => prevActiveStep + 1);
251 |         }
252 |     } else {
253 |         warning();
254 |     }
255 | };
256 |
257 | // Funkcija, kas pāreja atpakaļ uz iepriekšējo soli
258 | const handleBack = () => {

```

37.attels. Māju pievinošanas komponente

```

const express = require('express');
const cors = require('cors');
const stripe = require('stripe')('sk_test_510bI6xHCqLwHfBqfXm0TNjs4mVmHXPgCIApz5ChSscMSw4U1NvU9b4qLvGL4EwCLWj8XB9gQYTnwt2yYMrJMo3zJ00YTVq1fcP');

const app = express();
app.use(cors());
app.use(express.json());

const YOUR_DOMAIN = 'http://localhost:3001';

app.post('/create-checkout-session', async (req, res) => {
  try {
    const { amount, locale } = req.body;

    const session = await stripe.checkout.sessions.create({
      payment_method_types: ['card'],
      line_items: [
        {
          price_data: {
            currency: 'eur',
            product_data: {
              name: 'Custom Payment',
            },
            unit_amount: amount * 100,
          },
          quantity: 1,
        },
      ],
      mode: 'payment',
      success_url: `${YOUR_DOMAIN}?success=true`,
      cancel_url: `${YOUR_DOMAIN}?canceled=true`,
      locale: locale || 'lv',
    });

    res.json({ url: session.url });
  } catch (error) {
    console.error('Error creating checkout session:', error);
    res.status(500).json({ error: error.message });
  }
});

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

```

38.attels. Servera puses kods saistīts ar apmaksu