Liepājas Valsts tehnikums

**Nekustamā īpašuma nomas sistēma**

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Profesionālā kvalifikācija ........................................................................

Grupas nosaukums .........................................................................

Projekta izstrādātājs ...............................................................

/vārds, uzvārds, paraksts/

Eksāmena datums 202\_\_. gada\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Liepāja 202\_\_

**Satura rādītājs**

[Ievads 2](#_Toc505381207)

[1. Uzdevuma formulējums 2](#_Toc419706154)

[2. Programmatūras prasību specifikācija 3](#_Toc1880975060)

[2.1. Produkta perspektīva 4](#_Toc921549650)

[2.2. Sistēmas funkcionālās prasības 4](#_Toc1687505765)

[2.3. Sistēmas nefunkcionālās prasības 4](#_Toc639056320)

[2.4. Gala lietotāja raksturiezīmes 4](#_Toc1321854737)

[3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums 4](#_Toc1941477648)

[3.1. Izvēlēto risinājuma līdzēkļu un valodu apraksts 5](#_Toc236302345)

[3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts 5](#_Toc1101316560)

[4. Sistēmas modelēšana un projektēšana 5](#_Toc434232131)

[4.1. Sistēmas struktūras modelis 6](#_Toc952236156)

[4.2. Klašu diagramma / ER diagramma (Obligāti, ja sistēmā ir klases un/vai tabulas) 6](#_Toc1485057272)

[4.3. Funkcionālais un dinamiskais sistēmas modelis 6](#_Toc983243283)

[4.4. Aktivitāšu diagramma (Activity) 6](#_Toc134961801)

[4.5. Lietojumgadījumu diagramma (Use Case) 6](#_Toc252965698)

[4.6. Sistēmas moduļu apraksts un algoritmu shēmas 6](#_Toc1753954499)

[5. Lietotāju ceļvedis 6](#_Toc1882650428)

[6. Testēšanas dokumentācija 7](#_Toc558039722)

[6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums 8](#_Toc1704878675)

[6.2. Testpiemēru kopa 8](#_Toc2054531866)

[6.3. Testēšanas žurnāl 8](#_Toc1361314418)

[7. Secinājumi 8](#_Toc3018307)

[8. Lietoto terminu un saīsinājumu skaidrojumi 9](#_Toc1703744626)

[9. Literatūras un informācijas avotu saraksts 10](#_Toc1757670403)

[PIELIKUMS 11](#_Toc446171730)

# 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 atvieglojumu 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 gan ar sociālo tīklu kontiem, gan ar e-pasta adresi un paroli, tādējādi piedāvājot dažādas reģistrācijas iespējas. 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 pārvaldība:

Reģistrācija un autentifikācija:

Lietotājiem ir jāievada personīgie dati reģistrācijas procesā, nodrošinot drošu un viegli saprotamu pieredzi.

Autentifikācija var ietvert divfaktoru autentifikāciju (2FA) vai citus papildu drošības slāņus.

Profila pārvaldība:

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

2. Nekustamā īpašuma pārvaldība:

Informācijas pievienošana un rediģēšana:

Īpašniekiem ir piekļuve intuitīvam interfeisam, lai pievienotu un rediģētu nekustamā īpašuma informāciju, tādu kā adrese, platība, īpašuma apraksts, īpašuma tipi utt.

Augšupielādē attēlus un dokumentus, kas piedāvā lietotājiem vizuālu priekšstatu par īpašumu.

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.

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.

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.

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.

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.

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.

9. E-pasta paziņojumi:

Personalizētie paziņojumi:

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

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.

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.

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 puse ir būtisks solis, lai nodrošinātu vietnes kvalitāti un nodrošinātu lietotājiem uzticamu un drošu pieredzi.

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.

15. 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.

16. 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.

17. 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.

18 Visu lietotaja pidavajumu 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.

19. Karte

Šī funkcija ir atbildīga par attēlošanu Google karti 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 Google karti, kurā lietotājs var norādīt atrašanās vietu, izmantojot marķieri un kartes noklikšķināšanu.

Visbeidzot, funkcija atgriež komponenti, kas iekļauta dokumentācijā kā piemērs tam, kā implementēt kartes funkcionalitāti lietotāju atrašanai.

20. 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.

21. 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 lietotājvārdu 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.

## 2.3. Sistēmas nefunkcionālās prasības

Veiktspēja:

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

Sistēmai jāatbalsta vismaz 500 vienlaicīgu lietotāju bez ievērojamas veiktspējas pasliktināšanās.

Pieejamība:

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

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

Drošība:

Sistēmā jāizmanto HTTPS protokols, lai aizsargātu sensitīvu datu pārraidi.

Lietotāju paroles jāglabā šifrētā veidā.

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

Pieejamība:

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

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

Kļūdu pārvaldība:

Sniegt informatīvus un lietotājam saprotamus kļūdu ziņojumus.

Izstrādāt mehānismus kļūdu žurnālu apkopošanai un uzraudzībai, lai ātri reaģētu uz problēmām.

Uzturēšanas izmaksas:

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

Lietderība:

Saskarnes dizainam jābūt intuitīvam un draudzīgam dažādām lietotāju kategorijām.

## 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 līdzēkļu 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 atkļūvošana 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 deklaratīva 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.

Firebase datu bāze:

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

Pamatojums: Firebase 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, atvieglina 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 oriēntētu 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

Šis sadaļas ievads sniedz vispārīgu priekšstatu par "Ēku īres sistēmas" darbības diagrammām. Šīs diagrammas ir vizuālas reprezentācijas, kas palīdzēs izprast sistēmas darbības loģiku, procesus un mijiedarbību starp dažādiem elementiem.

Klasifikācijas diagramma

Klasifikācijas diagramma (Class Diagram) ir strukturēta diagramma, kas parāda sistēmas klases un to attiecības. Tā sniedz informāciju par klases atribūtiem, metodēm un to savstarpējām saistībām.

ER- (Entity-Relationship) diagramma

ER-diagramma ir grafisks attēlojums, kas parāda datu entitātes un to attiecības sistēmā. Tās galvenais mērķis ir izprast datu struktūru un savstarpējo saistību starp datu entitātēm.

Lietojumgadījumu diagramma

Lietojumgadījumu diagramma (Use Case Diagram) ir grafisks attēlojums, kas parāda sistēmas funkcionalitāti no lietotāja perspektīvas. Tās mērķis ir identificēt un aprakstīt lietotāja prasības un to, kā sistēma reaģē uz tiem.

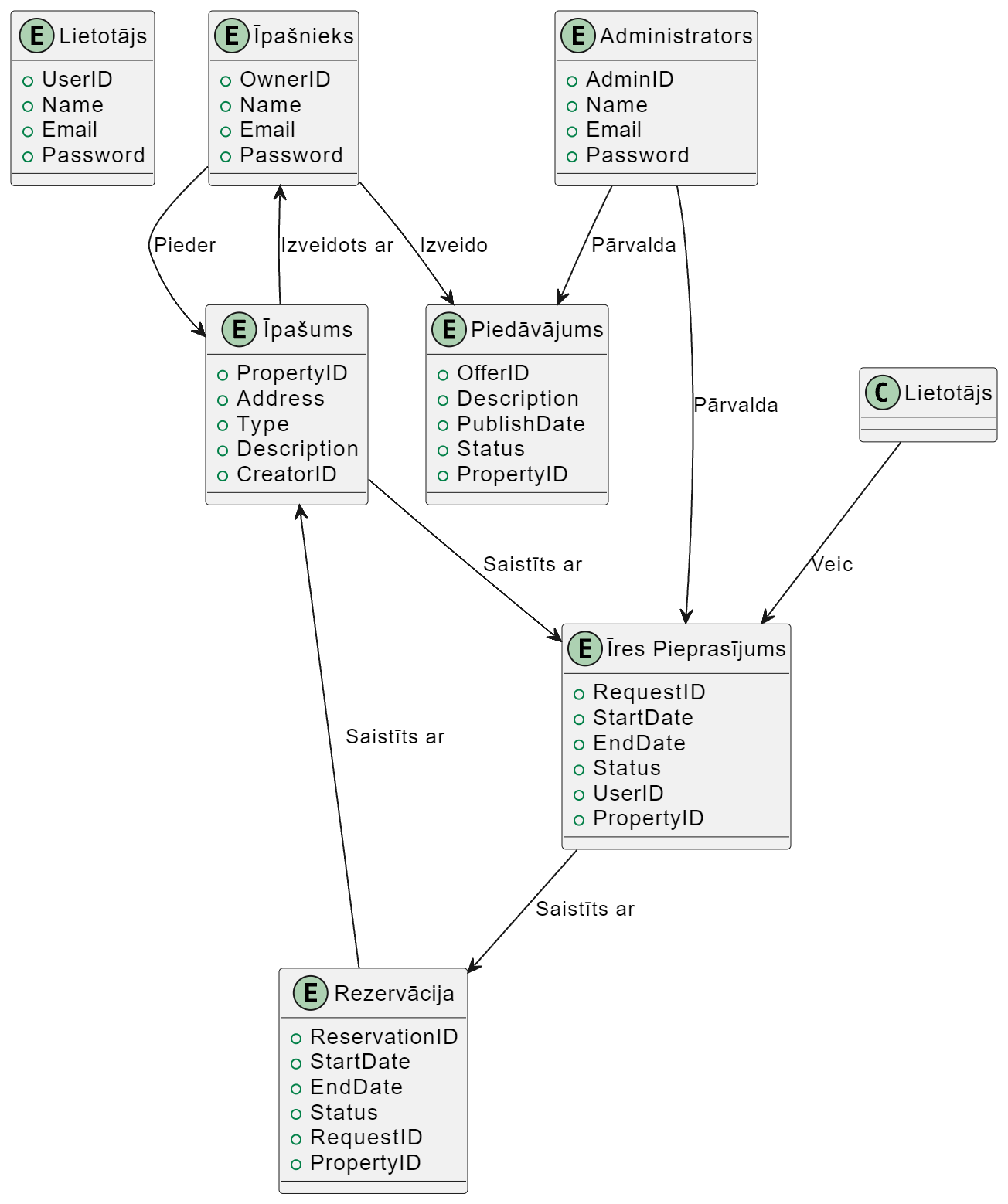
Secības diagramma

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

Šīs diagrammas kopā nodrošina padziļinātu izpratni par "Ēku īres sistēmas" struktūru, darbību un lietojumprogrammu, un tās ir noderīgas rīcības plānošanai un sistēmas izstrādei.

## 4.1. Sistēmas struktūras modelis

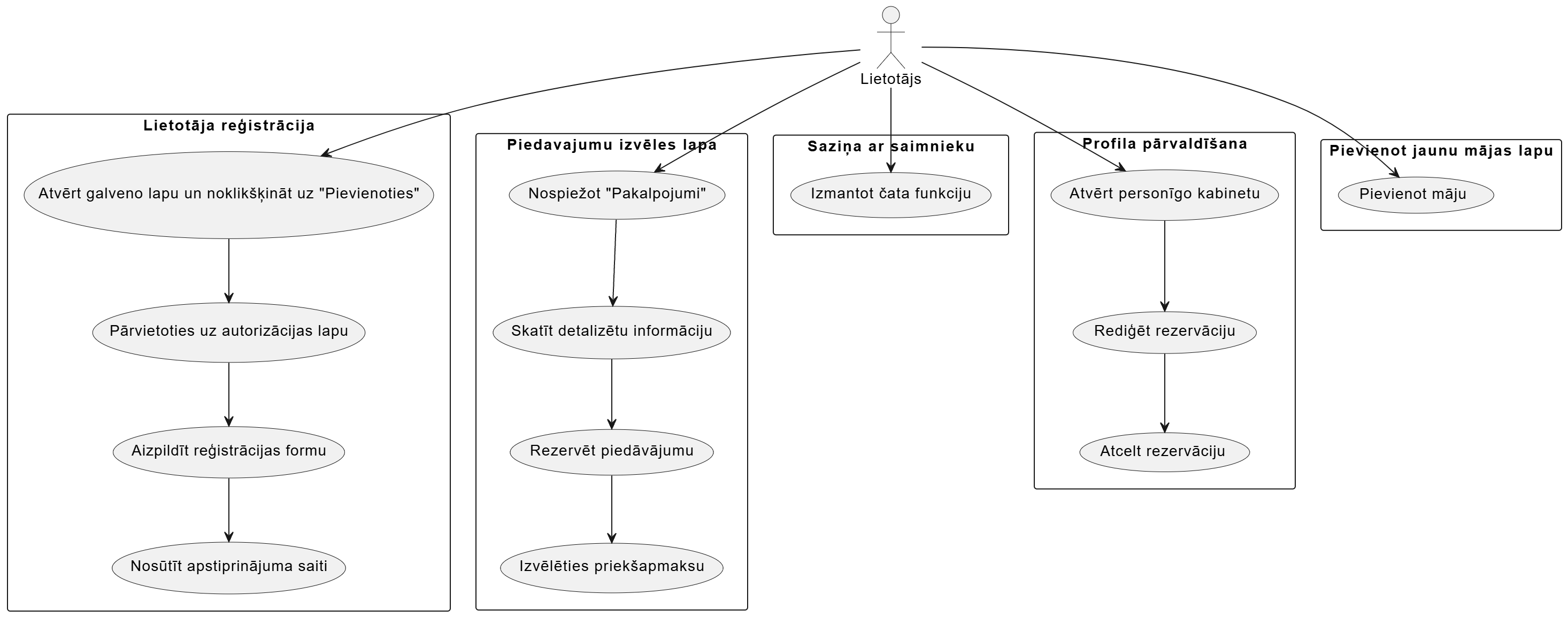
## 4.2. Klašu diagramma / ER diagramma (Obligāti, ja sistēmā ir klases un/vai tabulas)



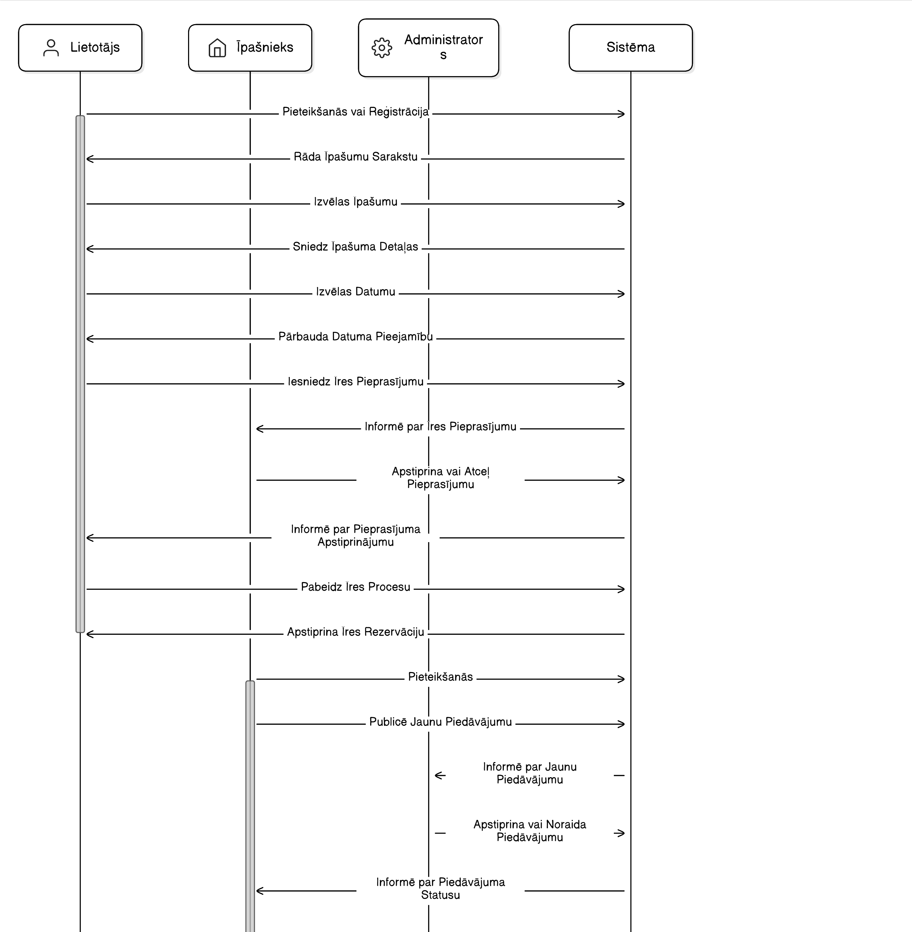
## 4.3. Funkcionālais un dinamiskais sistēmas modelis

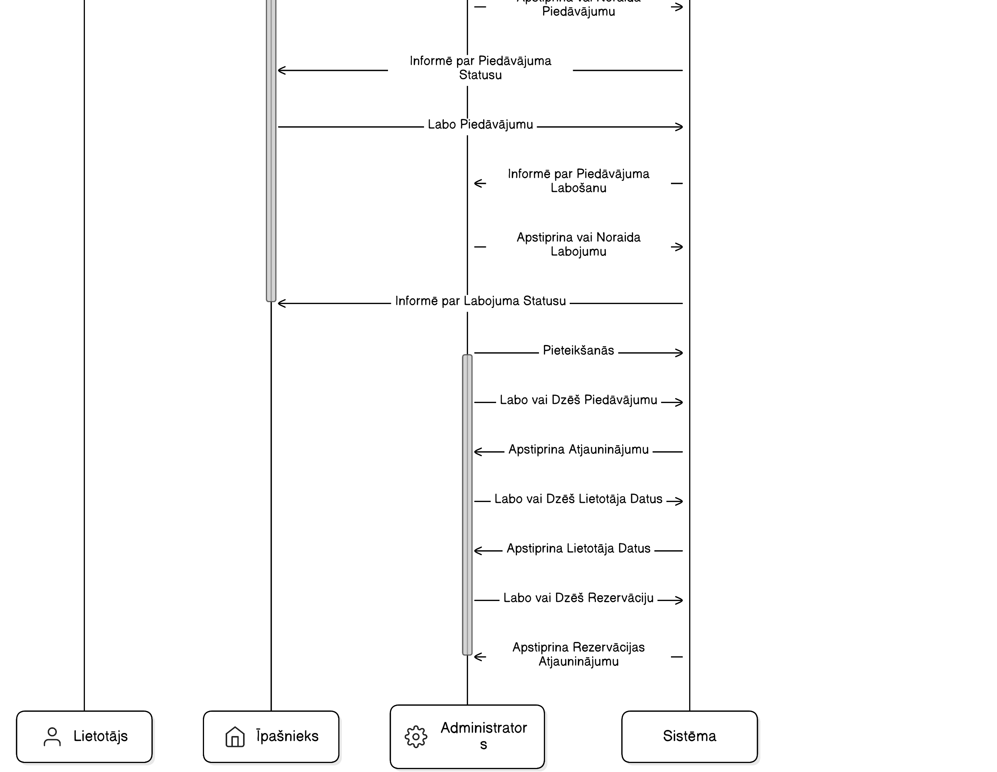
## 4.4. Aktivitāšu diagramma (Activity)

## 4.5. Lietojumgadījumu diagramma (Use Case)



4.6. secības diagramma





# 5. Lietotāju ceļvedis

# Sākums un autorizācija:

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

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

# Reģistrācija:

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

# 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.

# Galvenās pakalpojumu izvēles lapas:

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

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

# Rezervācija:

# Kad esi atradis interesējošo piedāvājumu, nospied "Skatīt vairāk", lai iegūtu detalizētu informāciju.

# Ja esi gatavs rezervēt, nospied "Rezervēt" un sekot instrukcijām, lai veiktu rezervāciju.

# Varēsi izvēlēties, vai veikt priekšapmaksu, lai rezervācija būtu apstiprināta.

# Saziņa ar saimnieku:

# Ja tev ir jautājumi vai nepieciešama papildus informācija par piedāvājumu, vari izmantot čata funkciju, kas pieejama uz piedāvājuma lapas.

# Profila pārvaldīšana:

# Nospiežot "Profils" galvenajā izvēlnē, atver savu personīgo kabinetu, kurā redzēsi visus savus rezervētos piedāvājumus.

# Vari rediģēt rezervāciju, mainīt laiku vai datumu vai arī atcelt rezervāciju.

# Pievienot jaunu mājas lapu:

# Nospiežot "Mans mājas" galvenajā izvēlnē, atver sadaļu, kur var pievienot jaunas mājas, kuras vēlies izīrēt.

# Nospiežot "Pievienot māju", tiek pārslēgts uz lapu, kur būs aizpildāma informācija par māju, ieskaitot adresi, ērtības, pieejamo vietu, pievienot bildes un norādīt cenas.

# 6. Testēšanas dokumentācija

## 6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums

## 6.2. Testpiemēru kopa

## 6.3. Testēšanas žurnāl

# 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

# PIELIKUMS