

Lengvųjų automobilių nuomos sistemos modelio projektavimas ir realizavimas

Donatas Daugirdas

Šiaulių valstybinė kolegija, lekt.

Šiauliai State University of Applied Sciences, Lithuania, Lect.

Domas Vileikis

Šiaulių valstybinė kolegija, stud.

Šiauliai State University of Applied Sciences, Lithuania, Stud.

Anotacija

Straipsnyje analizuojama lengvųjų automobilių nuomos rezervavimo sistema, suprojektuota pagal užsakovo poreikio sistemos kūrimui analizę. Atliktas panašių analogų lyginamasis vertinimas naudojant ISO-9126 programų inžinerijos produkto vertinimo standarto metodiką. Atlikta programavimo įrankių teorinė analizė, parengtas reliacinių duomenų bazių valdymo sistemos modelis. Šio modelio pagrindu sukurta ir įdiegta taikomoji programinė įranga, atliktas sukurtos sistemos testavimas.

Reikšminiai žodžiai: transportas, logistika, informacinės ir ryšių technologijos.

Projection and implementation of a model for car rental system

Summary

The article analyses the system for car rental reservation, designed according to the analysis of the customer's need for the development of the system. A comparative evaluation of similar analogues was performed using the methodology of the ISO-9126 software engineering product evaluation standard. Theoretical analysis of programming tools is performed, a model of relational database management system is prepared. On the basis of this model, application software was developed and installed, and the developed system was tested.

Keywords: transport, logistics, information and communication technologies.

Įvadas

Temos aktualumas. Rezervavimo sistema turi didelę įtaką verslui. Skaitmenizuotos rezervacijos sistemos didina darbo efektyvumą ir sumažina nereikalingas išlaidas. Zavarika (2016) straipsnyje [1] minima, jog rezervacijų skaitmenizavimas buvo pradėtas XX amžiaus šeštojo dešimtmečio pradžioje. Skaitmenizuoti imta dėl didėjančios paklausos ir krūvio, tai padėjo verslui atsikratyti papildomo darbo ir vykdyti veiklą daug greičiau. Verslo skaitmenizavimas pritraukia naujų klientų. Internetinė rezervavimo sistema – tai programinė įranga, kurią galima naudoti savo verslui ar rezervavimo paslaugai valdyti. Rezervavimo sistema leidžia įvairių rūšių paslaugų įmonėms priimti užsakymus internetu ir taip sutaupyti daug laiko [2]. Įmonėms labai svarbu, jog jų įmonės informacija būtų lengvai pasiekiamą internete. Šiuolaikinės technologijos padeda įmonėms lengvai valdyti jų pajamas ir užsakymus naudojant rezervacijos sistemas. Rezervacijos sistemų nauda: 1) padidina užsakymų kiekį; 2) taupo laiką; 3) padidina darbo efektyvumą; 4) suteikia pranašumą prieš konkurentus.

Žengiant technologijomis į priekį, dauguma verslų naudoja informacines technologijas. Ne išimtis – ir automobilių verslai. Siekdami padidinti darbo efektyvumą, būti daugiau pastebėtiems ir pritraukti naujų klientų, automobilių nuomos verslai savo veiklas perkelia į skaitmeninę rinką. Kompiuterinių technologijų poreikis yra labai svarbus versle tam, kad būtų galima greičiau ir efektyviau vykdyti sandorius ir rinkti duomenis.

Automobilių nuomos sistemos – tai tokios sistemos, kurios leidžia klientui greitai išsirinkti tinkamą automobilį ir jį rezervuoti elektroniniu būdu, o nuomininkui gauti tikslią informaciją apie automobilius ir jų būseną realiu laiku. Esant COVID-19 pandemijai, šios sistemos poreikis išaugo dar labiau, kadangi sistemos klientui nereikia vykti pas nuomininką tam, kad gautų reikiamą informaciją

apie automobilius: jis visą informaciją gali pasiekti internetinėje erdvėje ir, radęs tinkamą variantą, jis gali jį rezervuoti elektroniniu būdu. Taip išvengiama nuomininko nebūtinio kontakto su nuomotoju.

Tyrimo problema. Lietuvos rinkoje automobilių rezervacijos sistemos nėra naujiena, bet, atlikus analizę, galima teigti, jog kiekviena nuomos rezervavimo sistema turi savų pliusų ir savų minusų, todėl panaudojus tinklalapių analizę, bus sukurta automobilių nuomos sistema, kuri turės pilną suskaitmenintą procesą tarp kliento ir nuomininko.

Tyrimo objektas – rezervacijos sistema.

Tyrimo tikslas – suprojektuoti ir realizuoti lengvųjų automobilių nuomos rezervacijos ir jos administravimo sistemą.

Tyrimo uždaviniai: 1) atlikti užsakovo poreikių analizę; 2) išanalizuoti panašius analogus rinkoje; 3) atlikti sistemos programavimo įrankių teorinę analizę; 4) parengti duomenų bazės modelį; 5) pateikti panaudos atvejų diagramą; 6) atlikti sistemos testavimą.

Tyrimo metodai: mokslinių ir dokumentinių šaltinių analizė, lyginamoji analizė, sisteminimas ir apibendrinimas, kokybinis interviu, sistemos projektavimas, programavimas ir diegimas.

Užsakovo poreikių analizė

Norint vykdyti nuomos veiklą, reikia sukurti nuomos sistemą, į kurią įeina: tinklalapio dizainas, programinis sistemos kodas. Siekiant išsiaiškinti užsakovo poreikius kuriamai automobilių nuomos valdymo sistemai, buvo pasirinktas interviu metodas. Interviu pokalbis vyko 2021-10-02, 2022-02-11, 2022-04-28, kuriame dalyvavo įmonės generalinis direktorius, vadybininkas bei apklausos tyrėjas (šio straipsnio antrasis autorius). Interviu metu buvo išsiaiškinti dizaino reikalavimai, vartotojui reikalingos funkcijos bei administravimo sistemos reikalavimai: 1) sistemoje informacija turi būti lengvai prieinama; 2) sistema turi būti pritaikyta mobiliems įrenginiams; 3) administravimo pulte administratorius turi galėti koreguoti automobilių informaciją, stebėti jų techninę charakteristiką bei stebėti automobilio užimtumą realiu laiku; 4) sistemos dizaino paletėje spalvų gama turi būti šviesi.

Šiuo metu užsakovo tinklalapyje galima rasti tik kontaktinius duomenis ir informaciją apie jų vykdomą veiklą. Tinklalapis nebuvo optimizuotas Google paieškos varikliui, o tai lemia, jog tinklalapio lankomumo rodikliai yra žemi. Todėl buvo nuspręsta sukurti naują tinklalapį su galimybe išsinuomoti automobilį, kai sistema bus optimizuota Google paieškos varikliui.

Panašių analogų analizė

Atlikta trijų įmonių, kurios užsiima automobilių rezervacijomis ir jų nuoma, tinklalapių analizė. Remiantis Google paieškos sistema ir jos pateikiamais reitingais, parinkti 3 tinklalapiai: 2 lietuviški ir 1 užsienio. Tinklalapių analizei naudota dešimties balų vertinimo sistema. Kriterijams sudaryti buvo pasirinktas ISO-9126 standartas [3], kuris nusako, jog yra penkios pagrindinės modelio kokybės charakteristikos: funkcionalumas, našumas, perkeliamumas, naudojamumas, patikimumas. Pagal šias 5 pagrindines modelio kokybės charakteristikas sudaryti vertinimo kriterijai (žr. 1 lentelę).

1 lentelė

Tinklalapių lyginamoji analizė

Kriterijus	Tinklalapis	myavis.lt	autobanga.lt	momondo.com
Funkcionalumas				
Tinklalapis veikia visuose įrenginiuose		+	+	+
Dizainas prisitaiko prie įvairių įrenginių		+	+	+
Tinklalapio užkrovimo greitis < 5 sek.		+	-	-
Našumas				
Tinkami URL kanalai		+	+	+
Ar apsaugotas tinklalapis nuo struktūrinių duomenų klaidų ir pan.		+	+	-
Perkeliamumas				
Tinklalapio navigacijoje – tik prasmingos nuorodos		+	+	-
Navigacijoje esančios nuorodos – veikiančios		+	+	+
Naudojamumas				
Dizaino spalvų paletė suderinta		+	-	-

Palyginimo funkcija	+	-	+
Rezervacijos galimybė	-	-	+
Išsami filtravimo galimybė	-	-	+
Užimtumo grafikas (kalendorius)	-	-	-
Patikimumas			
TLS sertifikatas	+	+	+
Nuorodos, esančios tinklalapyje, patikimos	+	+	-
Nėra reklamų iš trečiųjų šalių tinklalapių	+	-	-
Surinktas balas / 15 galimų balų	13 / 15	8 / 15	8 / 15

Išvada. Išanalizavus 3 tinklalapius, geriausiu tapo tinklalapis **myavis.lt**, kuris surinko daugiausiai balų: 13 iš 15 galimų. Šis tinklalapis turi daugiausia funkcijų, reikalingų automobilio rezervacijos sistemai. Pastebėta, jog nei vienas iš visų trijų analizuojamų tinklalapių neturėjo pateikto užimtumo grafiko (kalendoriaus), kuriame būtų galima matyti, kuriuo laiku automobilis yra laisvas. Automobilio rezervacijos galimybė buvo rasta tik viename iš visų analizuojamų tinklalapių (momondo.com), kiti tinklalapiai turėjo tik užklauskos pateikimo formas. Filtravimo galimybė yra visuose tinklalapiuose, bet ji nėra išsami: mažai kriterijų. Spalvų paletė visuose trijuose tinklalapiuose dera tarp sistemos elementų. Nepasitaikė struktūrinės duomenų klaidos pranešimų. Remiantis atlikta analize bei aptarus su užsakovu rezultatus, buvo nuspręsta, jog sistema turi turėti šias galimybes:

- Rezervacijos galimybė – sistemos naudotojas įgis galimybę rezervuoti automobilį nuotoliniu būdu. Atlikus apmokėjimą bankiniu pavedimu, automobilis automatiškai bus rezervuotas;
- Užimtumo grafikas (kalendorius): sistemoje prie kiekvieno automobilio bus matomas kalendorius, kuriame bus galima matyti, kuriomis datomis automobilis yra rezervuotas ir kuriomis – laisvas;
- Sistemoje nebus reklamų iš trečiųjų šalių su nepatikimomis nuorodomis, pvz., svetainės nuorodų URL, neatitinkantys skelbimo galutinio URL domeno;
- Tinkama filtravimo galimybė – bus sukurta filtravimo galimybė.

Sistemos programavimo įrankių teorinė analizė

Kiekvienas tinklalapis kaip sistema yra programuojamas, naudojant skirtingas programavimo kalbas. Tinklalapių kodas yra išskiriamas į 2 dalis: 1) *Front-end* – išorinis programavimas, apima viską, ką vartotojas mato, įskaitant dizainą, jo karpymą ir kai kurias programavimo kalbas; 2) *Back-end* susideda iš trijų pagrindinių dalių – serverių, aplikacijos ir duomenų bazės. Back-end programuotojų parašytas kodas naršyklėje pateikia informaciją, laikomą duomenų bazėje. Norint suprogramuoti funkcionalų tinklalapį, reikia žinoti ir panaudoti tam tikras programavimo kalbas. Front-end'ui (svetainės išorei) yra naudojama HTML kompiuterinė žymėjimo kalba, o tinklalapio „griaučiams“ (Back-end) yra galimi skirtingi pasirinkimai. Pagal šių dienų statistiką dabar pasaulyje yra veikiančių 1,7 milijardo tinklalapių [4]. Remiantis statistika, 78,2 % visų tinklalapių naudoja programavimo kalbą PHP [5]. Ši programavimo kalba yra labiausiai paplitusi tarp tinklalapių programavimo kalbų dėl savo paprastumo ir greitos sąveikos su MySQL duomenų baze. Programavimo kalbų analizei buvo pasirinktos 3 populiariausios programavimo kalbos. Tiksliai analizei atlikti buvo sukurta vertinimo sistema. Rezultatai matomi 2 lentelėje.

2 lentelė

Programavimo kalbų analizė

Kriterijus	PHP	NODEJS	Ruby
Programavimo sintaksė	Lengva	Vidutinė	Lengva
Atviro kodo	Taip	Taip	Taip
Kaina	Nemokama	Nemokama	Nemokama
Tipas	Interpretuotas	Interpretuotas	Kompilijuojamas
Naudotojai	Daugiau nei 5 milijonai	2 milijonai	1 milijonas
Išlaikymas	Pigus	Vidutinis	Brangus
Išleidimo data	1995	2009	1996

PHP programavimo kalba buvo išleista 1995 metais [10]. Ši programavimo kalba yra atviro kodo, todėl ji yra nemokama. PHP programavimo kalba yra labiausiai naudojama tinklalapių kūrimui [5], todėl naudotojų skaičius yra ženkliai didesnis, nei kitų programavimo kalbų. Didelis naudotojų

skaičius parodo, jog ši programavimo kalba yra lengvai rašoma ir turi suprantamą sintaksę. Todėl, ateityje iškilus problemoms, sprendimo būdus galima rasti greitai dėl didelio naudotojų kiekio. Šios programavimo kalbos sintaksė yra lengva palyginus su „ASP.NET“ ar „Ruby“ programavimo kalbomis. PHP kalba yra pakankamai optimizuota ir nereikalauja didelių serverio resursų. Programuojamas kodas šioje kalboje nėra kompiliuojamas.

NODEJS programavimo kalba yra atvirojo kodo. Ši programavimo kalba skirta serverių programavimui arba kitaip – „Back-end“. Šios programavimo kalbos sintaksė yra šiek tiek sudėtingesnė, palyginus su PHP programavimo kalba. NODEJS reikalauja gana didelių serverio parametrų, norint ją naudoti sklandžiai. Visa tai reiškia, jog šios programavimo kalbos išlaikymas yra brangesnis, nei PHP kalbos.

„Ruby“ programavimo kalba yra atvirojo kodo. Ši programavimo kalba yra naudojama programuoti „Web“ aplikacijoms. „Ruby“ kalba yra panaši su programavimo kalba „Python“, todėl ši programavimo kalba gali būti naudojama ne tik „Web“ aplikacijoms. Šios programavimo kalbos bendruomenė nėra labai didelė, palyginus su PHP ar NODEJS. Skirtingai nuo kitų dviejų analizuojamų programavimo kalbų, „Ruby“ programavimo kalba yra kompiliuojama ir reikalauja didelių serverio resursų, o tai reiškia suprogramuotos sistemos brangesnį išlaikymą.

Išvada. Atlikus programavimo kalbų analizę, buvo pasirinkta PHP programavimo kalba dėl pigaus išlaikymo, didelio naudotojų skaičiaus, greitos sąveikos su duomenų baze ir dėl savo paprastos sintaksės.

Bibliotekos pasirinkimas

Pasitarus su užsakovu, nuspręsta rinktis geriausią 2021 metų biblioteką – „Laravel“ [6] dėl šių priežasčių: sauga; tinkama MVC struktūra; greitaveika; pigus išlaikymas; didelis naudotojų skaičius.

Tinklalapio dizaino kūrimo pasirinkimas

Tinklalapio dizainas bus kuriamas CSS karkaso pagalba. Naudojant CSS karkasą, yra ir pliusų, ir minusų [7].

CSS karkaso plusai: karkasai padeda pagilinti CSS žinias; turi jau parašytą kodą, kurio nereikia perrašinėti kelis kartus; palaiko visas naršyklės; tinklalapį suskaido į „tinklelį“, kuris padeda svetainės adaptyvumui skirtingiems įrenginiams; turi savo dokumentaciją. *CSS karkaso minusai:* pakeičia HTML rašymo įpročius; gali turėti nereikalingų papildomų klasių, todėl tinklalapis gali krautis lėčiau.

Projektavimui pasirinktas vienas populiariausių „CSS“ karkasų [8] – „Tailwind CSS“.

„Tailwind CSS“ karkasas pasirinktas dėl šių priežasčių [9]: „Laravel“ karkaso palaikymas; optimizuotas kodas; pašalintas nereikalingas kodas; turi savo konfigūracijos failus; turi jau sukoduotus komponentus.

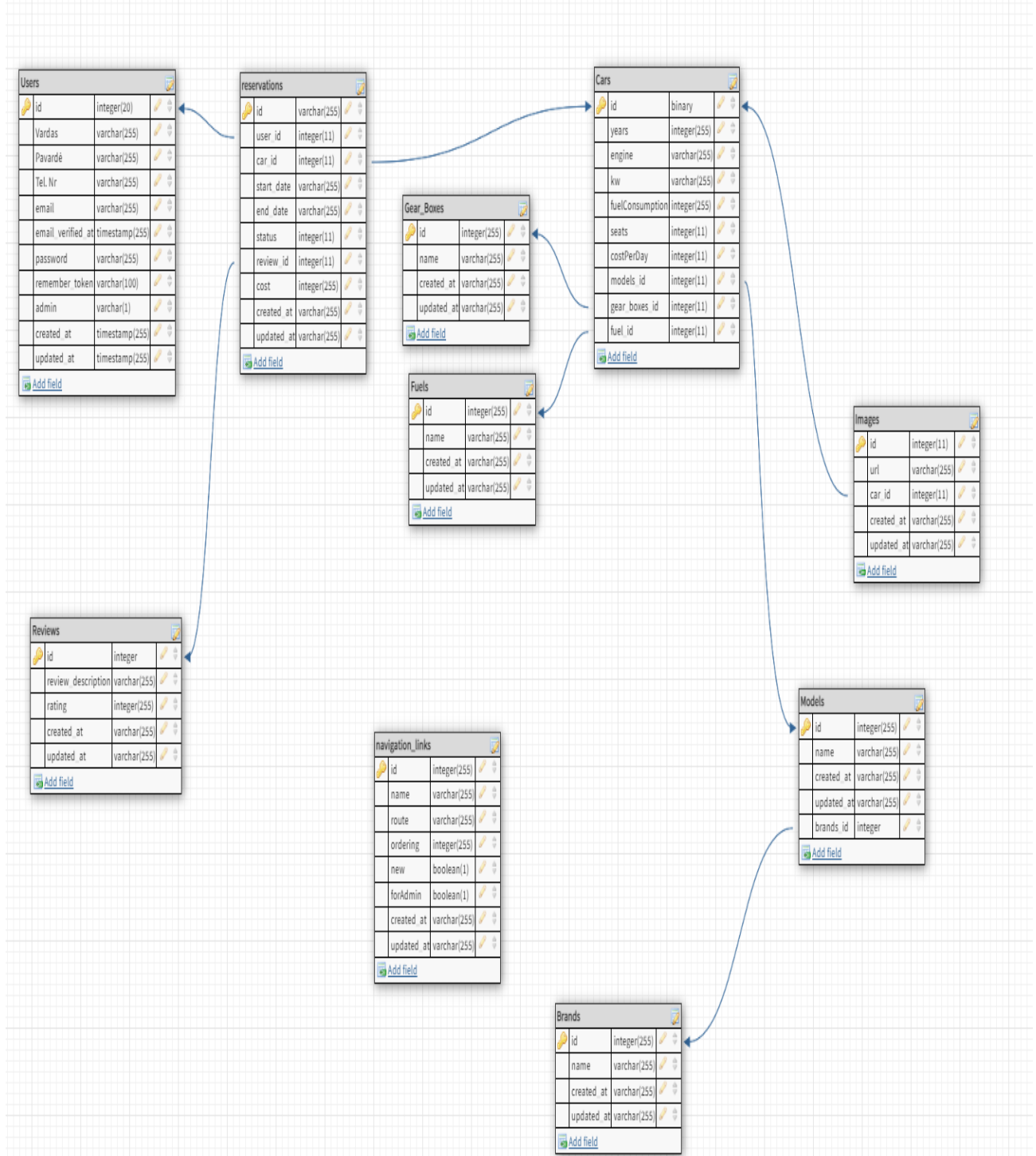
Duomenų bazės modelio parengimas

Didelės ir mažos įmonės bei organizacijos reguliariai generuoja vidutinius arba netgi didžiulius duomenų kiekius, todėl juos reikia nuolat atnaujinti ir tvarkyti. Retrospektyvinė duomenų analizė labai padeda tobulinti verslo strategijas ir rinkodaros tendencijas, todėl duomenys ilgai nepasensta, juos verta išsaugoti. Tačiau laikui bėgant šie duomenys gali augti ir tapti nevaldomais, jei bus tvarkomi įprastai, pavyzdžiui failų sistema. Todėl pasaulinėje rinkoje atsirado poreikis kurti duomenų bazių valdymo sistemas [11].

Duomenų bazė yra programinė įranga, skirta duomenims saugoti. Duomenų bazių valdymo sistemos užtikrina darbo automatizavimą pagal tikslų duomenų organizavimo modelį. Kuriant loginį modelį pasirenkama viena iš trijų modeliavimo prieigų: hierarchinis, tinklinis modelis arba reliaciniai tinklai. *Hierarchinis modelis* turi į medį panašią struktūrą ir atspindi vertikalų apatinės pakopos pavaldumo ryšį su viršutine pakopa. Taip lengviau pasiekti reikiamą informaciją tada, kai visos užklauskos turi į medį panašią struktūrą. *Tinklo modelis* yra sudėtingas ir nuo hierarchinio modelio skiriasi horizontalių ryšių buvimu. *Reliacinis modelis* pateikiamas lentelių rinkinio pavidalu. Lentelėse formuojamos operacijos. Modelio pranašumas yra santykinis paprastumas, šiuolaikinių jį palaikančių priemonių kompaktiškumas, duomenų struktūros aiškumas, veikimo greitis priklauso nuo

duomenų bazės dydžio. Reliaciniai modeliai dabar yra plačiai paplitę, juose visi komponentai yra sujungti tarpusavyje apibrėžtu ryšiu [12].

Automobilių nuomos sistemos duomenų krovimui ir saugojimui buvo pasirinkta MySQL duomenų bazė. MySQL – viena iš reliacinių duomenų bazių valdymo sistemų, palaikanti daugelį naudotojų, dirbanti SQL kalbos pagrindu. Saugojimo varikliai yra žinomi kaip „lentelių tvarkytojai“, iš esmės yra duomenų bazės dalys, kurios interpretuoja ir valdo su duomenų bazių lentelių SQL užklausomis susijusias operacijas. Naujausiose MySQL versijose saugojimo variklius galima projektuoti ir valdyti naudojant „prijungiamą“ architektūrą [13]. Yra įvairių saugojimo variklių, tačiau dažniausiai naudojami du: InnoDB ir MyISAM. InnoDB yra standartinis MySQL serverio duomenų talpinimo variklis. Pasirinktas InnoDB talpos variklis. Duomenų bazės koduotė buvo pasirinkta „utf8_general_ci“ (žr. 1 pav.).

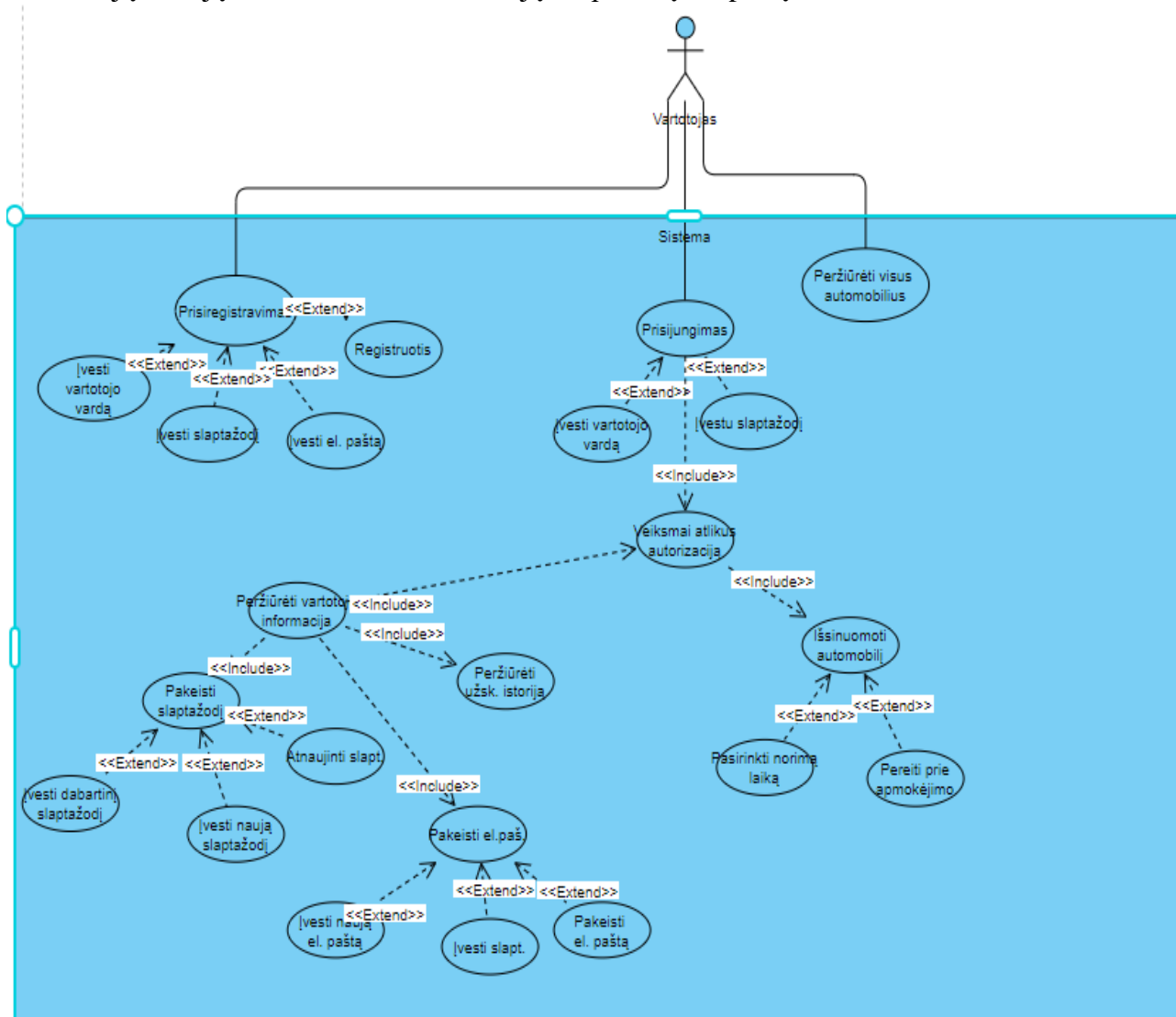


1 pav. Duomenų bazės schema (sukurta autorių)

Automobilių nuomos sistemos panaudos atvejų diagramos modeliavimas

UML – modeliavimo ir specifikacijų kūrimo kalba, skirta specifikuoti, atvaizduoti ir konstruoti į objektus orientuotų programų dokumentus.

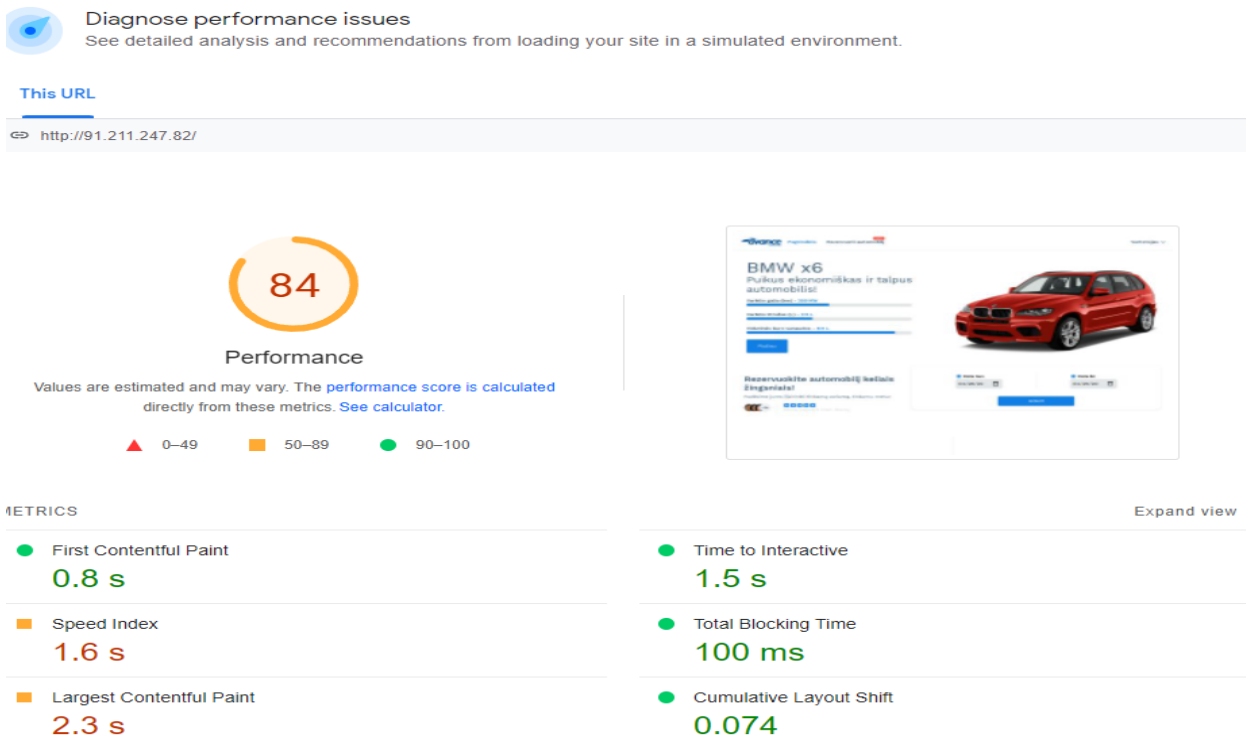
Suprojektuota automobilių nuomos sistemos panaudos atvejų diagrama (žr. 2 pav.). Iš šios diagramos galime matyti vartotojo veiksmus sistemoje. Vartotojas, norėdamas užsiregistruoti sistemoje, turi įvesti savo vardą, pavardę, telefono numerį ir el. pašto adresą. Prisijungimui reikalingi tik du laukai: vartotojo el. paštas ir slaptažodis. Norint peržiūrėti visus siūlomus automobilius, vartotojui nereikia prisijungti prie sistemos, bet jei vartotojas nori atlikti rezervacijos veiksmą ir rezervuoti automobilį tam tikromis datomis, tuomet vartotojas privalo prisijungti prie sistemos. Norint rezervuoti automobilį, reikia pasirinkti rezervacijos laiką kurį sudaro: rezervacijos pradžios ir rezervacijos pabaigos datos. Rezervacijos apmokėjimas yra galimas 2 būdais: grynaisiais pinigais ir elektroniniu atsiskaitymu – elektronine bankininkyste. Prisijungęs vartotojas gali peržiūrėti rezervacijų istoriją bei keisti savo informaciją: slaptažodį, el. paštą.



2 pav. Sistemos panaudos atvejų diagrama (sukurta autorių)

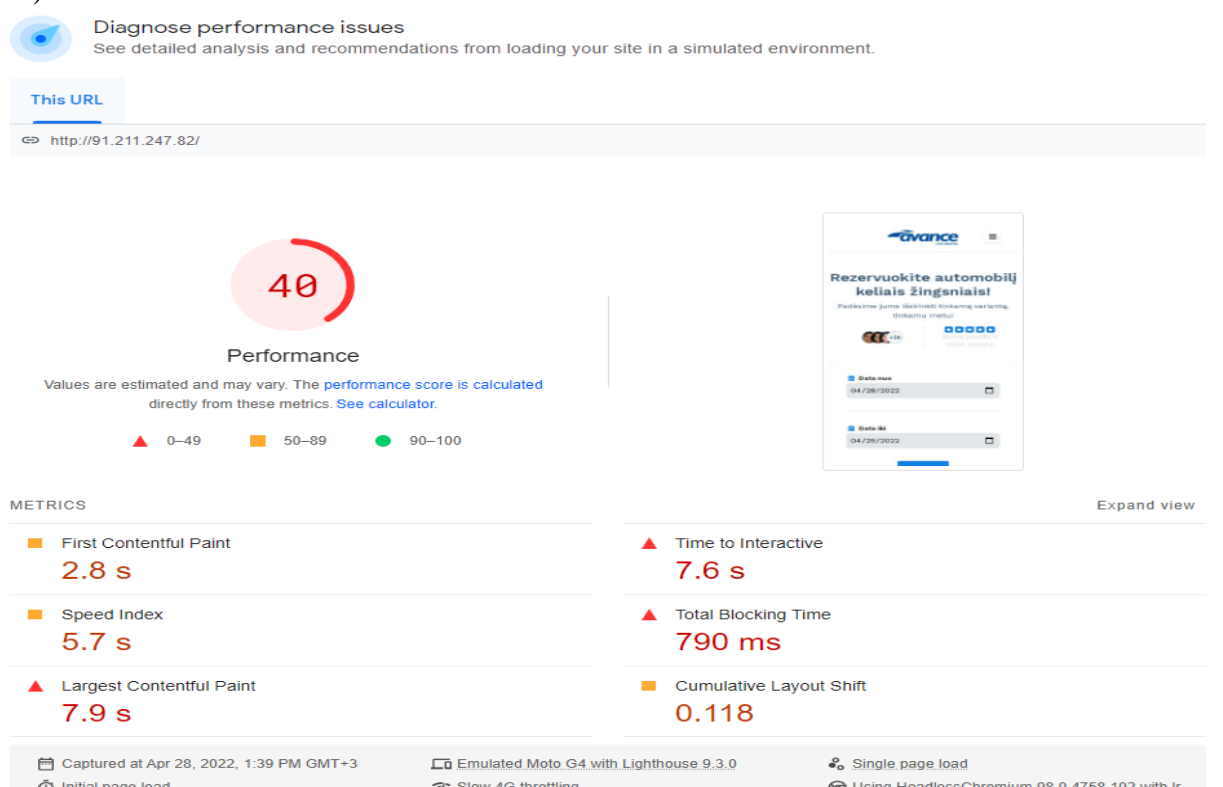
Sukurtos sistemos testavimas

Sistema sukurta ir patalpinta lokaliame serveryje testavimo-programavimo darbams bei nutolusiame serveryje. Sistemos testavimui buvo naudojamas „Google Page Speed Insights“ įrankis. Šis įrankis nustato (matuoja) tinklalapio atsivertimo greitį kompiuteriams, planšetėms ir mobiliems įrenginiams. Rezultatas šiame įrankyje pateikiamas nuo 0 iki 100 balų skalėje. Atlikus pagrindinio puslapio greičio tikrinimo testą kompiuteriui, buvo gautas rezultatas – 84 balai (žr. 3 pav.).



3 pav. tinklalapio atsivėrimo greičio rezultatas kompiuteryje (sudaryta autorių)

Atlikus pagrindinio puslapio greičio tikrinimo testą telefonui, gautas rezultatas – 40 balų (žr. 4 pav.).



4 pav. tinklalapio atsivėrimo greičio rezultatas telefone (sudaryta autorių)

Lyginant kompiuterio ir telefono atsivėrimo greičių rezultatus, buvo išsiaiškinta, jog kompiuteryje tinklalapio krovimo greitis yra patenkinamas, bet telefone balas nėra didelis. Norint padidinti šiuos skaičius ir pagreitinti tinklalapį, reikėtų optimizuoti JavaScript. Tinklalapio Front-end'ui testuoti buvo rašomi Unit testai, kurių pagalba buvo galima stebėti duomenų gavimą ir pasikeitimą (žr. 5 pav.).


```
Snapshot Summary
> 1 snapshot updated from 1 test suite.

Test Suites: 2 passed, 2 total
Tests:       5 passed, 5 total
Snapshots:  1 updated, 1 total
Time:        1.021s
```

5 pav. Unit testai (sudaryta autorių)

Šių testų pagalba galima išbandyti kiekvieną tinklalapio komponentą atskirai. Testų pagalba galima nustatyti, kokia turi būti komponento reikšmė prieš testui paleidžiant ir kokia reikšmė turi būti testui atėjus į pabaigą. Jeigu informacija atitinka, tuomet į konsolę grąžinamas pranešimas – „passed“.

Išvados

Išanalizuotas įmonės poreikis kuriamai sistemai. Aprašyti veiklos funkcijų uždaviniai, išanalizuoti užsakovo keliami reikalavimai, iškelti strateginiai tikslai. Atlikta teorinė analizė ir pasirinktos reikalingos priemonės, įrankiai projektui įgyvendinti.

Atlikta panašių sistemų analizė, buvo analizuojamos trys sistemos, susijusios su automobilių nuoma. Kriterijams sudaryti buvo pasirinktas ISO-9126 šeimos standartas. Palyginimo metu buvo išsiaiškinta, kokios funkcijos būdingos tokioms sistemoms ir kurios funkcijos yra rečiau naudojamos, tačiau reikalingos, norint užtikrinti patogesnę ir greitesnę vartotojo darbą. Analizės metu nustatyta, kad geriausiu tapo tinklalapis myavis.lt kuris surinko daugiausiai balų (13 iš 15). Šios analizės pagrindu nustatyti kuriamos sistemos reikalavimai ir galimybės.

Palyginus galimas kūrimo technologijas, buvo išrinktos geriausiai tinkamos technologijos šios sistemos kūrimui. Vidiniai sistemos daliai ir išorinei buvo pasirinkta naudotis *PHP* programavimo kalba, pasinaudojant Tailwind CSS karkasu. Pasitarus su užsakovu, nuspręsta rinktis geriausią 2021 metų biblioteką – „Laravel“. Duomenų saugojimui buvo pasirinkta naudotis *MySQL* duomenų baze, talpos variklis pasirinktas „InnoDB“.

Suprojektuota sistema: parengtos panaudos atvejų diagramos bei duomenų bazių diagrama.

Sukurta, realizuota automobilių registravimo sistema su interneto prieiga, sukurtos vartotojų ir administratorių funkcijos. Sistema patalpinta nutolusiame serveryje bei lokaliame.

Siekiant užtikrinti tinkamą sistemos veikimą, joje buvo testuotos svarbiausios funkcijos, sistemos suderinamumas su įvairiomis naršyklėmis, rezoliucijomis.

Literatūra

1. Zavarika, G. (2016). The evolution of computer reservation systems in tourism. In: *Globalization of scientific and educational space. Innovations of transport. Problems, experience, prospects*, p. 47-49. <https://core.ac.uk/download/pdf/84593901.pdf>
2. Gungon, R. (2020). *What is an Online Reservation System and why do you need it?* Omnify. <https://www.getomnify.com/blog/one-reservation-system>
3. Buenaflor, L. (2017). *ISO 9126 Software Quality Characteristics*. <https://medium.com/@leanarbuenaflor/iso-9126-software-quality-characteristics-a25a26e7d046>
4. Website Setup (2021) *How Many Websites Are There?* <https://websitesetup.org/news/how-many-websites-are-there/>
5. W3techs (2022). *Historical yearly trends in the usage statistics of server-side programming languages for websites*. Web Technology Surveys. https://w3techs.com/technologies/history_overview/programming_language/ms/y
6. Salisbury, S. (2022). *Why is Laravel the Best PHP Framework in 2021?* KOMODO. <https://www.komododigital.co.uk/insights/why-laravel-is-the-best-php-framework-in-2021>
7. Coyier, Ch. (2008). *What Are The Benefits of Using a CSS Framework?* CSS-TRICKS. <https://css-tricks.com/what-are-the-benefits-of-using-a-css-framework>

8. Bose, Ch. (2021). *Top 5 CSS Frameworks for Developers and Designers*. Browser Stack. <https://www.browserstack.com/guide/top-css-frameworks>
9. Young, N. (2021). The Pros and Cons of Tailwind CSS. <https://www.webdesignerdepot.com/2021/09/the-pros-and-cons-of-tailwind-css/>
10. Moutaouakkil, A., Mbarki, S. (2020). Generating a PHP meta model using X text framework. *Procedia Computer Science*, 170, 838-844.
11. Sharma, A., Karamchandani, A., Dave, D., Patel, A., Doshi, N. (2022). *Database Management Systems – An Efficient, Effective, and Augmented Approach for Organizations*. Springer Nature Switzerland AG.
12. Rashidov, A. R. O. (2022). The difference between the concepts of database and database management system. *Archive of Conferences*, 33-34. <https://www.conferencepublication.com/index.php/aoc/article/view/1845>
13. Matsunobu, Y., Siying D., Lee, H. (2020). My Rocks: LSM-tree database storage engine serving Facebook's Social Graph. *Proceedings of the VLDB Endowment*, 13(12), 3217-3230.