Pirkimo sąlygų

1 priedas

**RINKLIAVOS KONTROLĖS PROGRAMĖLĖS SU IŠEITIES KODU SUKŪRIMO   
TECHNINĖ SPECIFIKACIJA**

# **Bendroji informacija**

* 1. **Perkančioji organizacija:** SĮ „Susisiekimo paslaugos“ (toliau – PO).
  2. **Pirkimo objektas:** Rinkliavos kontrolės programėlės sukūrimas su išeities kodu (toliau – paslaugos).
  3. Projekto įgyvendinimas turi būti atliktas per **3 mėnesius** nuo sutarties pasirašymo.
  4. Pirkimo objektui taikomi „žalieji“ reikalavimai – vadovaujantis Aplinkos apsaugos kriterijų taikymo, vykdant žaliuosius pirkimus, tvarkos aprašo, patvirtinto Lietuvos Respublikos aplinkos ministro 2011 m. birželio 28 d. įsakymu Nr. D1-508 (aktualia redakcija), 4.4.3. p., perkama tik nematerialaus pobūdžio (intelektinė) ar kitokia paslauga, nesusijusi su materialaus objekto sukūrimu, kurios teikimo metu nėra numatomas reikšmingas neigiamas poveikis aplinkai, nesukuriamas taršos šaltinis ir negeneruojamos atliekos.

# **Sąvokos**

* 1. **API (Application Programming Interface)** - sąsaja, leidžianti programai bendrauti su kitomis programomis ar paslaugomis. Dažniausiai naudojami standartizuoti protokolai, pvz., HTTPS.
  2. **APK (Android Package Kit)** - Failo formatas, naudojamas Android operacinėje sistemoje programėlėms diegti.
  3. **Autentifikacija** - Procesas, kurio metu vartotojo tapatybė patvirtinama naudojant prisijungimo duomenis (vartotojo vardą ir slaptažodį).
  4. **Kontrolės zona** - Parkavimo kontrolės programėlė, naudojama nustatyti, kur ir kaip galioja parkavimo leidimai.
  5. **Leidimas** - Leidimas transporto priemonei stovėti tam tikroje vietoje be papildomų mokesčių, pavyzdžiui, gyventojo leidimas, rezervuota vieta ir kt.
  6. **Naudotojo patirtis (UX)** - Naudotojo patirties aspektai, susiję su programėlės naudojimo patogumu, efektyvumu ir pasitenkinimu.
  7. **Naudotojo sąsaja (UI**) - Programėlės grafinis dizainas, kuris leidžia vartotojui sąveikauti su programa. UI turi būti intuityvi ir lengvai naudojama.
  8. **OAuth2** - Autentifikacijos ir autorizacijos protokolas, naudojamas užtikrinti, kad vartotojai ir programos galėtų saugiai pasiekti API resursus, neatskleisdami savo prisijungimo duomenų.
  9. **Saugumo testas** - kurio metu siekiama identifikuoti ir pašalinti programėlės saugumo spragas.
  10. **PFDB (Parkavimo fiksavimo duomenų bazė)** – duomenų bazė, kurioje saugomi parkavimo pažeidimų ir leidimų duomenys. Ši duomenų bazė naudojama tikrinant leidimų galiojimą ir apdorojant su parkavimu susijusias operacijas.
  11. **Prieigos kontrolė** - Procesas, kuris riboja prieigą prie tam tikrų programėlės dalių, remiantis vartotojo rolėmis ir leidimais. Pvz., administratoriaus teisės gali skirtis nuo paprasto vartotojo.

# **Pirkimo objekto bendrieji techniniai ir funkcionalumo reikalavimai**

Ši techninė dokumentacija skirta apibrėžti reikalavimus programėlei, kuri turi būti sukurta, siekiant užtikrinti aukščiausią kokybę, saugumą ir funkcionalumą. Programėlė turi būti skirta Android operacinei sistemai, o jos paskirtis – atlikti automobilių parkavimo patikrinimus naudojant kamerą, rankinį numerio įvedimą ir vartotojo autentifikaciją per API.

**Tikslas** -Pagrindinis šios programėlės tikslas – suteikti kontrolieriams galimybę greitai ir patogiai atlikti automobilių parkavimo patikrinimus. Programėlė turi pasižymėti patikimumu, saugumu ir intuityviu vartotojo sąsajos dizainu. Visi duomenys privalo būti perduodami per saugius API kanalus, naudojant pažangiausius šifravimo metodus.

* 1. **Pagrindinis langas ir meniu**
     1. Zonos pasirinkimas
        1. Programėlė turi leisti pasirinkti zoną iš sąrašo arba nustatyti automatiškai pagal GPS koordinates
        2. Zonų sąrašas gaunamas iš API
        3. Turi būti galimybė lengvai surasti/filtruoti zonas
        4. GPS koordinačių tikslumas turi atitikti reikalavimus zonos nustatymui kuriuos turi būti galima keisti nustatymuose.
     2. Transporto priemonių tikrinimas
        1. Programėlė turi leisti atlikti tikrinimą dviem būdais:
           1. Automatinis numerio nuskaitymas naudojant įrenginio kamerą
           2. Rankinis numerio įvedimas
        2. Tikrinimo rezultatas turi būti aiškiai matomas ir išsaugomas Programėlėje
        3. Programėlė turi kaupti tikrinimų istoriją
     3. Pažeidimų valdymas
        1. Programėlė turi leisti registruoti pažeidimus patikrinimams
        2. Pažeidimų sąrašas turi būti lengvai peržiūrimas ir valdomas
        3. Turi būti galimybė:
           1. Pasirinkti kelis pažeidimus vienu metu
           2. Masiškai pridėti ženklus prie pasirinktų pažeidimų
           3. Matyti pažeidimo vietą ir laiką
           4. Filtruoti/rūšiuoti pažeidimus
     4. Duomenų išsaugojimas
        1. Tikrinimų ir pažeidimų duomenys turi būti išsaugomi iki dienos pabaigos (00:00)
        2. Programėlė turi užtikrinti duomenų vientisumą ir neprarandamumą iki nustatyto laiko
        3. Po nustatyto laiko duomenys turi būti automatiškai išvalomi tik iš programėlės
     5. Apsauga nuo dubliuotų pažeidimų fiksavimo
        1. Programėlė turi automatiškai patikrinti, ar tam pačiam valstybiniam numeriui jau nėra užfiksuoto pažeidimo per pastarąsias 24 valandas
        2. Aptikus potencialų dublikatą, sistema turi parodyti įspėjimą su ankstesnio pažeidimo duomenimis
        3. Kontrolierius turi turėti galimybę patvirtinti naują pažeidimą arba atšaukti veiksmą
        4. Programėlė taip pat turi turėti apsauga nuo dubliuojančio pažeidimo fiksavimo. Pvz. dvigubas paspaudimas saugoti.
  2. **Numerių atpažinimo reikalavimai**
     1. Kontrolierius padaro transporto priemonės valstybinio numerio nuotrauką naudojant įrenginio kamerą, ir tik tada vykdo automatinį numerių nuskaitymo algoritmą (LPR), kuriame naudojama OCR (Optical Character Recognition) technologija valstybiniams numeriams atpažinti.
     2. Programėlė turi gebėti atpažinti numerius nepaisant kameros laikymo kampo, dulkių ar kitų nežymių kliūčių, išskyrus atvejus, kai numeriai yra visiškai nematomi. Be to, automobilio valstybinis numeris turi būti nuskenuojamas ir tamsiu paros metu.
     3. Turi būti teisingai nuskaityti nestandartiniai numeriai, įskaitant vardinius, elektromobilių, motociklų bei kitų valstybių numerius. Sistema turi gebėti atpažinti ir patikrinti ne tik lietuviškus numerius. Blogai nuskaitytų numerių nuotraukos būtų kaupiamos mokymui, kad pagerintų OCR modelį, kartu išsaugant ir teisingai įvestą numerį.
     4. OCR Modulis turėtų galimybę mokytis iš 3.3.4 punkte nurodytų patikslintų nuotraukų.
     5. Turi būti galima konfigūruoti patikimumo nuskaitymo procentą (pagal nutylėjimą 90 procentų).
     6. Kontrolierius gauna aiškų teigiamą patvirtinimą per:
        1. VIBRACIJĄ: Vienas trumpas vibracijos signalas (0.5 sekundės)
        2. GARSĄ: Vienas trumpas „ding“ tipo garsinis signalas
        3. EKRANĄ:
           1. Ekranas nušvinta ŽALIA spalva
           2. Rodomas didelis žalias varnelės (✓) ženklas
           3. Apačioje pateikiama tekstinė informacija:
           4. „GALIOJANTIS STOVĖJIMAS“
           5. Galiojimo laikas: „nuo 15:30“
           6. Jei yra leidimas rodoma informacija: pvz., „GALIOJA | MLG-123456 | Zona: Geltona ir žalia zonos“ arba „GALIOJA | GL1-42977 | Ter.: 25“
           7. Jei transporto priemonei šiandien jau buvo išrašytas pažeidimas: pvz.: „2025-02-14 09:10 | ABC123 | S. Konarskio gatvė 37 | Žalioji“
           8. Jei transporto priemonė šiandien buvo fiksuota rezervuotoje vietoje: pvz.: „Statymo vieta: UŽIMTOS REZERVUOTOS, TIKRINTI LEIDIMĄ, Zona: Visos zonos, Įmonė: „Uab Testas“ Galioja nuo: 2025-02-14 12:23:52 Galioja iki: 2025-02-14 23:59:59 Pastabos:“
           9. Mygtukas „Fiksuoti KET pažeidimą“, leidžiantis kontrolieriui užfiksuoti kitus Kelių eismo taisyklių pažeidimus (pvz., stovėjimas ant pėsčiųjų perėjos, ant vejos, ne pagal ženklus ir pan.).
           10. Mygtukas „Fiksuotis rezervuotos vietos pažeidimą“, leidžiantis kontrolieriui užfiksuoti rezervuotos vietos pažeidimą.
     7. Kontrolierius gauna aiškų neigiamą patvirtinimą per:
        1. VIBRACIJĄ: Trys trumpi vibracijos signalai (0.2 sek. intervalais)
        2. GARSĄ: Trys trumpi „pyp“ tipo garsiniai signalai
        3. EKRANĄ:
           1. Ekranas nušvinta RAUDONA spalva
           2. Rodomas didelis raudonas (X) ženklas
           3. Apačioje pateikiama tekstinė informacija:
           4. „NEGALIOJANTIS STOVĖJIMAS“
           5. Priežasties pasirinkimas iš API pateikto sąrašo.
           6. Jei transporto priemonei šiandien jau buvo išrašytas pažeidimas: pvz.: „2025-02-14 09:10 | ABC123 | S. Konarskio gatvė 37 | Žalioji“
           7. Mygtukas „Fiksuoti RINKLIAVOS pažeidimą“ už stovėjimo apmokėjimo pažeidimą.
           8. Mygtukas „Fiksuoti KET pažeidimą“, leidžiantis kontrolieriui užfiksuoti kitus Kelių eismo taisyklių pažeidimus (pvz., stovėjimas ant pėsčiųjų perėjos, ant vejos, ne pagal ženklus ir pan.).
           9. Mygtukas „Fiksuotis rezervuotos vietos pažeidimą“, leidžiantis kontrolieriui užfiksuoti rezervuotos vietos pažeidimą.
     8. Numerio atpažinimo greitis turi būti ne ilgesnis nei 2 sekundės nuo kameros užfiksavimo momento.
     9. Pažeidimų nuotraukos turi būti išsaugomos su pavadinimu formatu: „YYYY-MM-DD\_HHMMSS\_VALSTNR“ (pvz., „2025-02-14\_143022\_ABC123“).
     10. Turi būti galimybė rankiniu būdu koreguoti automatiškai atpažintus numerius prieš patvirtinant.
     11. Paskyrus pažeidimą grįžtama į skanavimo langą ir tęsiama sesija.
     12. Nuotraukose privalo būti išsaugotos GPS koordinatės EXIF metaduomenyse
  3. **Rankinis numerių tikrinimas**
     1. Vartotojas turi galimybę įvesti valstybinius numerius rankiniu būdu.
     2. Tikrinant numerius yra atskiras langas fiksuoti KET pažeidimą, kurį pasirinkus vartotojas nukeliamas į pažeidimo langą.
     3. Programėlė turi automatiškai patikrinti, ar numerio formatas atitinka leistinas taisykles, neleistų įvesti netinkamų simbolių, tokių kaip tarpai, simboliai ir pan. Būtų galima įvesti tik raides ir skaičius be tarpų. Visos raidės turi būti paverčiamos į didžiąsias.
     4. Įvedus transporto priemonės numerius, vartotojas nukreipiamas į kitą langą, kuriame bus atvaizduojama informacija apie užsakytą ar neužsakytą parkavimą.
     5. Šiame lange nurodoma informacija apie naudotojo stovėjimo būseną: transporto priemonės valstybinis numeris, Užsakymo būsena, Mokamo stovėjimo pradžia, Rinkliavos zona, Kontrolės zona.
     6. Ir taip pat būtų atvaizduojami šiame lange suteikti leidimai transporto priemonei. Leidimų būsenos: Pagal automatų gedimus, Pagal rezervuotas vietas, Pagal gyventojų leidimus
     7. Gavus atsakymą, leidžiama pasirinkti, skirti pažeidimą ar praleisti pasirinkus priežastį kodėl praleidžiama.
     8. Sistema turi tikrinti "Juodąjį sąrašą" (Black List) ir aiškiai indikuoti, jei transporto priemonė yra įtraukta į šį sąrašą.
  4. **Pažeidimo atspausdinimo funkcija**
     1. Naudotojas gali atspausdinti pranešimą apie pažeidimą.
     2. Gavus atsakymą, jog stovėjimas neužsakytas atidaromas pažeidimo formavimo langas.
     3. Atvaizduojami šie duomenys: tikrinimo data, laikas, adresas, pažeidimo pobūdis, valstybinis numeris.
     4. Pažeidimo pobūdis – leidžiama pasirinkti iš sąrašo pažeidimų, kurie parsisiunčiami per API.
     5. Kiekvienam pažeidimo tipui turi būti numatytoji parinktis:
        1. Rinkliava – siūloma „16.1“ kaip pažeidimo tipas.
        2. KET – ženklas 333 (stovėti draudžiama).
        3. Rezervuota vieta – siūloma „16.5“ kaip pažeidimo tipas.
        4. Administratorius turi turėti galimybę pakeisti numatytas reikšmes nustatymuose.
        5. Prie kiekvieno pažeidimo tipo rodomi tik tam tipui priklausomi pažeidimai.
     6. Nurodoma adreso laukelis, tai vartotojui yra siūlomas adresas iš žemėlapio integracijos. Bet adresą turi būti galimą pataisyti. Papildomų adresu sąrašas turi būti atsisiunčiamas per API.
     7. Turi būti atskiri laukai, kurie leis pridėti nuotraukas (iki 10 vnt.) ir komentarus.
     8. Suformuojama bauda išsiunčiama į sistemą.
     9. Po to kai pažeidimas nusiunčiamas į sistemą spausdinamas pažeidimo lapelis. Turi būti galima nustatymuose išjungti automatini spausdinimą.
     10. Pažeidimas patenką į sąrašą, kuriame matosi pažeidimo data laikas, pažeidimo pobūdis, automobilio numeris, adresas. Su galimybe pakartoti pažeidimo spausdinimą.
     11. Programėlėje turi būti galimybė matyti visus išrašytus pažeidimus ir pasirinkus varnelėmis pažeidimus pridėti reikiamus papildomus kelio ženklus, kad nereiktų kiekvieno pažeidimo atsidarinėti ir atskirai pridėti tuos pačius ženklus.
     12. Visos prie pažeidimo padarytos nuotraukos turi būti automatiškai optimizuotos - nuotraukos dydis neturi viršyti 1MB, o maksimali raiška turi būti 2064x1290 pikselių. Programėlė privalo išlaikyti originalią fotografavimo orientaciją (horizontalią/vertikalią), išlaikyti nuotraukos proporcijas ir neiškraipyti vaizdo kokybės. Optimizuotose nuotraukose neturi matytis akivaizdžių vaizdo suspaudimo artefaktų ar spalvų iškraipymų.
     13. Kiekviena nuotrauka turi būti papildyta informacine eilute:
         1. Vieta: apatinėje nuotraukos dalyje
         2. Išvaizda: juodas fonas, baltas Arial šriftas (lengvai įskaitomas dydis po optimizavimo)
         3. Turinys: „DATA LAIKAS | AUTOMOBILIO NUMERIS | GPS KOORDINATĖS“
         4. Formatas: „2025-02-14 14:30:22 | ABC123 | GPS: 54.687157, 25.279652“
         5. Įterpimas: informacinė eilutė tampa neatskiriama nuotraukos dalimi prieš optimizavimą
     14. Visos nuotraukos privalo išsaugoti GPS koordinates EXIF metaduomenyse, nepriklausomai nuo optimizavimo

1. **Technologiniai Reikalavimai**
   1. **Platforma**
      1. Programėlė turi būti visiškai suderinama su Android operacine sistema.
   2. **Technologijos**
      1. Programavimo kalba: Kotlin arba Java.
      2. Duomenų saugojimas: vietinės duomenų bazės naudojimas svarbiems duomenims saugoti.
   3. **API** 
      1. Programėlė naudos REST API.
      2. API komunikacija: HTTPS užklausos, saugumas užtikrinamas naudojant OAuth2.
      3. Visi duomenys turi būti perduodami tik šifruotais HTTPS kanalais.
      4. API apsaugotas naudojant prieigos ženklus (access tokens), kurių galiojimo laikas turi būti ribotas.

# **Saugumo Reikalavimai**

* 1. **Autentifikacija ir autorizacija**
     1. Prisijungimui naudojamas vartotojo vardas ir slaptažodis.
     2. Vartotojo slaptažodžiai turi atitikti griežtus reikalavimus (minimalus ilgis, simbolių įvairovė ir pan.).
     3. Naudojami unikalūs prieigos ženklai (access tokens), kurių galiojimo laikas yra ribotas.
     4. Prisijungimo metu turi būti naudojamas API ir generuojamas prieigos ženklas (angl. access token) JWT.
     5. Programėlė neturi reikalauti pakartotinio prisijungimo išjungus ekraną ar išėjus į foną.
     6. Turi būti realizuotas biometrinis atrakinimas (pirštų atspaudai arba veido atpažinimas), leidžiantis greitai atnaujinti sesiją.
  2. **Duomenų apsauga**
     1. Visi API užklausų duomenys turi būti perduodami per HTTPS.
     2. Visi vartotojo duomenys, saugomi įrenginyje, turi būti šifruoti naudojant AES-256 šifravimą.
  3. **Prieigos kontrolė**
     1. Naudotojų teisių valdymas, leidžiantis riboti funkcionalumą pagal vartotojo rolę.
     2. Apsaugos mechanizmai, užkertantys kelią SQL injekcijoms ir XSS atakoms.

# **Naudotojo Sąsajos (UI/UX) Reikalavimai**

* 1. **Naudotojo patirtis (UX)**
     1. Programėlė turi būti pilnai suderinama su išmaniaisiais telefonais ir planšetiniais kompiuteriais, užtikrinant sklandų veikimą bei optimizuotą naudotojo patirtį abiejuose įrenginiuose.
     2. Programėlės kalba – Lietuvių. Vartotojo sąsaja turi būti intuityvi, aiški ir lengvai naudojama.
     3. Sąsaja turi būti pritaikyta darbui lauko sąlygomis.
     4. Svarbiausi veiksmai turi būti lengvai pasiekiami.
     5. UX sprendimai turi būti suderinti su tiekėju prieš galutinį įgyvendinimą.
     6. Turi būti parengti ir suderinti UX prototipai pagrindinėms funkcijoms.
     7. Esamo UX dizaino pavyzdžiai:
  2. **Naudotojo sąsajos (UI) reikalavimai**
     1. Sąsaja turi būti pritaikyta įvairiems įrenginių ekranų dydžiams
     2. Programėlė turi būti pritaikyta lietimui jautriems ekranams.

# **Testavimas ir kokybės užtikrinimas**

* 1. Programėlė turi būti ištestuota visais galimais naudojimo atvejais.
  2. Turi būti atlikti saugumo testai ir veikimo našumo testai.

# **Programėlės pateikimas turi apimti:**

* 1. Sukompiliuotą APK failą (Android programėlės paketą)
  2. Projekto kodo komitai (commits) turi būti atliekami reguliariai ir nuosekliai viso projekto vystymosi metu, atspindint realią funkcionalumo vystymo eigą.
  3. Draudžiama komitinti didelius kodo kiekius retais intervalais ir/arba tik projekto pradžioje/pabaigoje.
  4. Rekomenduojama:
     1. Komitinti baigus įgyvendinti vieną loginę funkcionalumo dalį
     2. Vengti labai didelių komitų, kurie apima daug skirtingų funkcionalumo pakeitimų
     3. Išlaikyti reguliarų komitų dažnumą viso projekto metu

# **Dokumentacija**

* 1. Vartotojo dokumentacija:
     1. Išsami naudotojo instrukcija, skirta galutiniam vartotojui
  2. Techninė dokumentacija:
     1. API specifikacijos
     2. Funkcionalumo aprašymai
     3. Testavimo ataskaitos
  3. Programuotojo dokumentacija:
     1. Sistemos diegimo instrukcija:
        1. Detalūs diegimo žingsniai
        2. Programavimo aplinkos paruošimo nurodymai
        3. Priklausomybių ir bibliotekų sąrašas
     2. Šios instrukcijos turi būti pakankamai detalios, kad kiti programuotojai, vadovaudamiesi jomis, galėtų sėkmingai:
        1. Įsidiegti ir paleisti, sukompiliuoti programėlę
        2. Paruošti programavimo aplinką
        3. Tęsti programos tobulinimą ir vystymą
        4. Atlikti programėlės testavimą skirtinguose įrenginiuose.

1 priedas

**API aprašas**

Visos užklausos, išskyrus prisijungimą, reikalauja autentifikacijos per prieigos tokeną (access token), kuris gaunamas per **/api/token** POST užklausą (būtinas HTTPS).

**Pagrindinės API funkcijos parkavimo kontrolės programėlei:**

1. **Vartotojo autentifikacija:**
   * **POST /api/token**: Leidžia vartotojams prisijungti prie programėlės ir gauti prieigos tokeną. Palaikomi įvairūs prisijungimo būdai (pvz., vartotojo vardas/slaptažodis, mobiliojo operatoriaus autentifikacija).
2. **Parkavimo galiojimo tikrinimas:**
   * **POST /api/ParkingInspection**: Leidžia programėlei patikrinti, ar konkretaus automobilio parkavimas tam tikroje zonoje yra galiojantis. Tai gali būti naudojama realiu laiku tikrinant parkavimo statusą.
3. **Parkavimo pažeidimų valdymas:**
   * **POST /api/ParkingInspection/Fine**: Leidžia programėlės naudotojams (pvz., kontrolieriams) registruoti naujus parkavimo pažeidimus.
   * **POST /api/ParkingInspection/RegisterFineAndGetNotice**: Leidžia iškart užregistruoti pažeidimą ir sugeneruoti pranešimą.
   * **GET /api/ParkingInspection/FineNotice**: Leidžia peržiūrėti arba atspausdinti jau užregistruotą pažeidimo pranešimą pagal jo ID.
   * **GET /api/ParkingInspection/GetFines**: Leidžia peržiūrėti anksčiau užregistruotų pažeidimų sąrašą pagal parkavimo zoną.
4. **Konfigūracijos duomenų gavimas:**
   * **GET /api/ParkingInspection/config**: Leidžia programėlei gauti dinamiškai kintančius konfigūracijos duomenis, tokius kaip parkavimo zonos, galiojančios gatvės, pažeidimų tipai, automobilių markės ir modeliai. Tai užtikrina, kad programėlė visada naudoja naujausią informaciją.