



KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS

**KOMANDA: NASA**

## **PROJEKTAS: NUMERIŲ ATPAŽINIMO SISTEMA**

Modulis: PR00B122 Semestro projektas  
Atsiskaitymo ataskaita

**Paskaitų dėstytojas:**  
lekt. DOBROVOLSKIS  
Algirdas

**Vertintojas:**  
lekt. DOBROVOLSKIS  
Algirdas

**Studentai:**  
Saulius Krugliakovas IFIN  
7/1  
Norkus Paulius IFIN 7/1  
Domantas Kelpšas IFIN 7/1

Kaunas, 2019

## TURINYS

Įvadas .....	3
1. Analizė .....	4
1.1. Techninis pasiūlymas .....	4
1.1.1. Sistemos apibrėžiams .....	4
1.1.2. Sistemos aktualumas ir nauda .....	4
1.2. Galimybių analizė .....	4
2. projektas .....	5
2.1. Reikalavimų specifikacija .....	5
2.1.1. Sistemos funkcijos .....	5
2.1.2. Apribojimai .....	5
2.1.3. Vartotojo sąsajos specifikacija .....	5
2.1.4. Realizacijai keliami nefunkciniai reikalavimai .....	5
2.1.5. Techninė specifikacija .....	5
2.2. Projektavimo technologijos .....	5
2.2.1. Projektavimo įrankiai, programavimo kalbos .....	5
2.3. Sistemos projektas .....	6
2.3.1. Įrenginiui naudoti įtaisai .....	6
2.3.2. Numerių atpažinimo algoritmas .....	7
3. Testavimas .....	8
3.1. Testavimo pavyzdžiai ir rezultatai: .....	8
3.1.1. Aplikacijos testavimas .....	8
3.1.2. Įtaiso testavimas .....	9
4. Dokumentacija naudotojui .....	12
4.1. Programos diegimas .....	12
4.2. Klaidos ir jų padarinių atstatymas .....	12
4.3. Sistemos naudojimas .....	12
4.4. Sistemos tipinės funkcijos .....	12
4.5. Techninės įrangos konfigūracija .....	12
4.6. Sistemą sudarantys failai .....	12
5. Projektavimo valdymas ir Eiga .....	13
5.1. Sprintai .....	13
5.2. Burndown grafikas .....	14
6. Rezultatų apibendrinimas ir išvados .....	15
7. Literatūra .....	16

## IVADAS

Šio programavimo inžinerijos projekto pagrindinis tikslas – įgyti bazinius įgūdžius ir žinias reikalingas sukurti programinės įrangos projektą taikant Agile metodiką bei ypatybes. Taikant šią metodiką realizuoti funkcionuojančią sistemą.

Nusprendėmė, kurti numerių atpažinimo sistemą, kuri taikant aikštelių sistemą būtų realizuota, kuri galėtų nufotografuoti mašinos numerį, įrašytą į duomenų bazę bei prijungtą bendrai prie sistemos.

Projekta sugalvojome išanalizavę dabartinius populiariausias problemas ir pamatėmė, kad viena iš jų yra aikštelių sistemos. Taip bandėmė sukurti savo įsivaizduojama aikštelės sistemą.

Darbo uždaviniai:

- Susipažinti su programinės įrangos kūrimo procesu;
- Susipažinti su techninės įrangos programavimo principais;
- Pasirinkti ir realizuoti nuotraukų palyginimo algoritmą;
- Įgyvendinti programą kompiuteryje, kuri galėtų valdyti bei žiūrėti duomenis iš duomenų bazės
- Įgyvendinti programa su Raspberry Pi, kuri galėtų nufotografuoti telefono numerį bei užtikrintu programos efektyvumą
- Apjungti sukuryą funkcionalumą į bendrą projektą.

Nuorodą į projekto kodą: <https://github.com/Katekizmas/NASA>

Projektavimo komandos nariai:

- Saulius Krugliakovas IFIN 7/1, [saulius.krugliakovas@edu.ktu](mailto:saulius.krugliakovas@edu.ktu) - atsakingas už kompiuterio programos kodą, duombazę bei ataskaitą.
- Domantas Kelpšas IFIN 7/1, [domantas.kelpsas@edu.ktu](mailto:domantas.kelpsas@edu.ktu) - atsakingas už kompiuterio programos kodą, jos veikimą bei vartotojo sąsają.
- Paulius Norkus IFIN 7/1, [paulius.norkus@edu.ktu](mailto:paulius.norkus@edu.ktu) - atsakingas už techninę projekto pusę.

## **1. ANALIZĖ**

### **1.1. Techninis pasiūlymas**

#### **1.1.1. Sistemos apibrėžiams**

Šio projekto metu kuriamas įrenginys bus atsakingas už:

- Nuoseklų ir be klaidų programos veikimą
- Kuo tikslesnį numerių nuskaitymą bei apdorojimą
- Vartotojo bei administratorių naudingą duomenų atvaizdavimą

#### **1.1.2. Sistemos aktualumas ir nauda**

Šiais laikais yra begale žmonių su mašinomis. Taip pat begalę aikštelių, tačiau kaip jas efektyviai kontroliuoti ir tvarkyti naudojant kuo efektyvesnius būdus. Žmonės šitoje srityje bando išsiaiškinti, kaip taip padaryti.

Pačios sistemos nauda jeigu ši sistema būtų toliau pratęsta kurti būtų galima kiekvieną aikštelę naudoti būtent šią sistemą.

### **1.2. Galimybių analizė**

Igyvendinant šį projektą turėjome nuspręsti, ką naudoti vienas iš projekto narių turėjo techninę įrangą iš kitų projektų. Taigi todėl pasiėmėme Raspberry Pi ir kuriame projekto pagrindą ant jo.

Tai vienas iš plačiausiai naudojamų techninių varikliukų, kurių funkcijos ir modeliavimas yra begalinis. Nors tai vienas iš geresnių pasirinkimų tačiau galima ir naudoti ir kitas sistemas kaip FEZ Raptor ir kt.

## 2. PROJEKTAS

Aprašoma sistemos projektavimo stadija, pateikiama detali specifikacija. Apibrežiama kuriamo produkto vizija.

### 2.1. Reikalavimų specifikacija

#### 2.1.1. Sistemos funkcijos

- Įrenginys nuskaityto/nufotografuoja automobilio numerį.
- Įrenginys patikrina ar yra automobilis duomenų bazėje, jeigu ne prideda, jeigu taip prideda prie stovėjimo laikų papildomą informaciją.
- Vartotojas gali atsidaryti telefoną/kompiuterį ir patikrinti savo automobilio būseną bei jeigu išvažinėja apmokėti.
- Administratorius gali prieiti prie detalesnės informacijos apie visas įmones, padalius, aikšteles bei automobilius.
- Administratorius gali pažiūrėti statistiką apie kiekvieną aikštelę.

#### 2.1.2. Apribojimai

- Įrenginys fotografuoja nuotraukas prasta kokybę
- Įrenginys su dabartine atpažinimo sistema nėra garantuotas, kad visada nuskanuos tą patį numerį nors nuotrauka tapati (nes algoritmas naudoja didžiausią šansą)
- Sistema negali žinoti kurią būtent vietą užima automobilis aikštelėje

#### 2.1.3. Vartotojo sąsajos specifikacija

Įrenginio vartotojo sąsaja turi atvaizduoti galimų klaidų pranešimų, bei mašinos statusą, per unikalų sugeneruotą kodą sukurti duombazės query ir atvaizduoti informaciją apie mašinos numerį, stovėjimo būseną, pradžios ir pabaigos laiką, ir bendrą sumą kiek turi mokėti.

#### 2.1.4. Realizacijai keliami nefunkciniai reikalavimai

Reikalavimas	Kriterijus
Programavimo kalba turi turėti griežtus kintamųjų tipus	Integruota programinės įrangos aplinka, tipų klaidų pranešimai kompiliavimo metu
Programavimo kalba turi būti objektiškai orientuota	Kiekvienas periferinis įrenginys yra programuojamas kaip atskiras objektas
Įtaisas turi turėti kamerą	Norint fotografuoti automobilių numerius gyvai reikalinga kamera.
Naudojamas įrenginys turi turėti prieigą prie interneto.	Turi būti prieinamas prie interneto, kad vartotojas/administratorius duoti užklausas duomenų bazėje

#### 2.1.5. Techninė specifikacija

Norint kad įrenginys veiktų jis privalo galėti kompiliuoti ir paleisti Python kodą bei turi kamerą kuri galėtų fotografuoti. Prieiga prie interneto.

### 2.2. Projektavimo technologijos

Įrenginiui suprojektuoti nebuvo naudojamas specifinis architektūros projektavimo modelis.

#### 2.2.1. Projektavimo įrankiai, programavimo kalbos

Sistemos kūrimui naudojome programą Visual Studio 2017. Su šia programa įgyvendiname visas savo užsibrėžtas užduotis. Programos dalis buvo sukurta naudojantis C# programavimu bei WinForms aplikacija. Įrenginyje buvo naudojama Python kalba. Numerio atpažinimo algoritmui buvo

naudojamas: ALPR ( <https://www.openalpr.com/> ). Projekto sistemos failų valdymui naudojome GIT platformą, visas kodas patalpintas GITHUB saugykloje.

## 2.3. Sistemos projektas

### 2.3.1. Įrenginiui naudoti įtaisai

Raspberry Pi 3

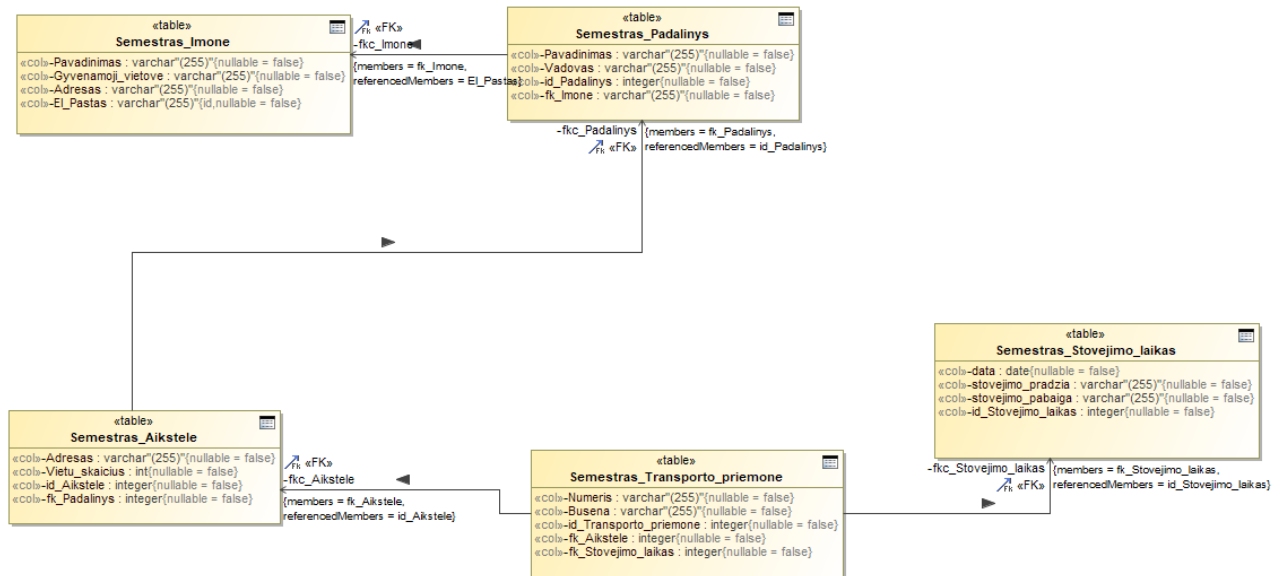


Raspberry Pi Camera Module V2



- Pagrindinis įtaisas Raspberry Pi3 skirtas kontroliuoti techninę projekto pusę ir atlieka visus skaičiavimus.
- Kamera, kuri fotografuoja numerius.

Klasių diagrama:



Semestras\_Imone – Talpina įmonės informacija

Semestras\_Padaliny – Talpina informacija apie padalinį priklausantį įmonei

Semestras\_Aikstele – Skirta informacijai apie aikštelę kuri priklauso padaliniui

Semestras\_Transporto\_priemone – Skirta talpini automobilio informacija bei jos būseną aikštelėje

Semestras\_Stovėjimo\_laikas – Kiekvienas automobilis gali stovėti tos pačios įmonės skirtinguose aikštelėse todėl yra pridėta ši klasė kuri talpina kiekvienos mašinos stovėjimo laikus.

### 2.3.2. Numerių atpažinimo algoritmas

Kadangi mūsų ištekliai bei laikas neleido sukurti individualaus ir unikalaus numerių atpažinimo algoritmo mes naudojama viena iš populiariausių algoritmu vadinamo: Open ALPR, kur daugiau informacijos apie tai galima rasti jų oficialiame tinklapyje: <https://www.openalpr.com/>

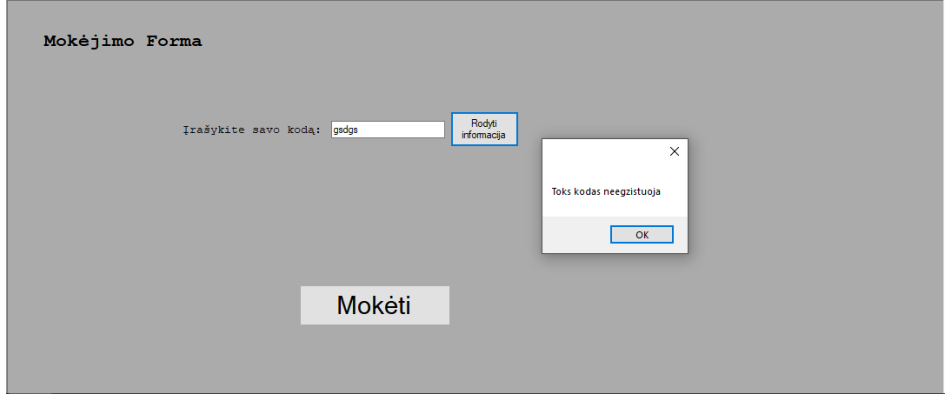
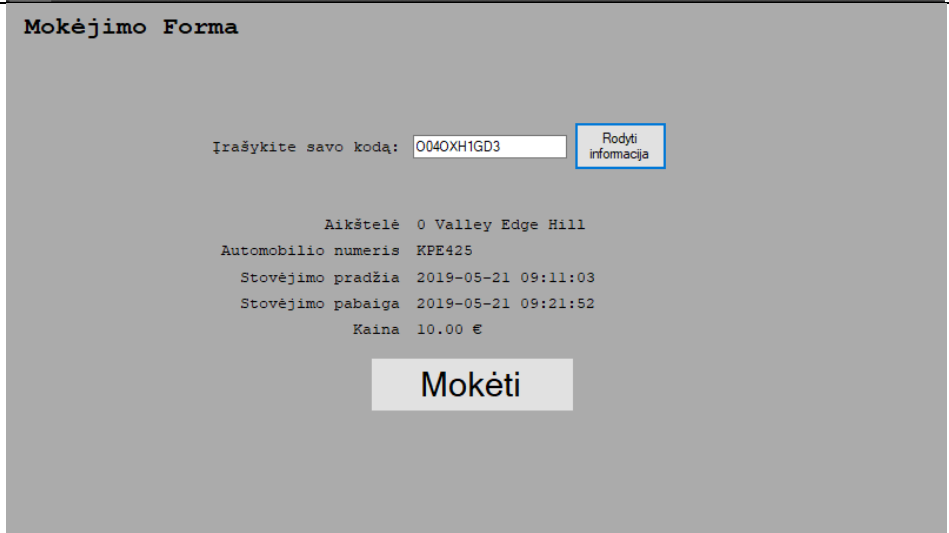
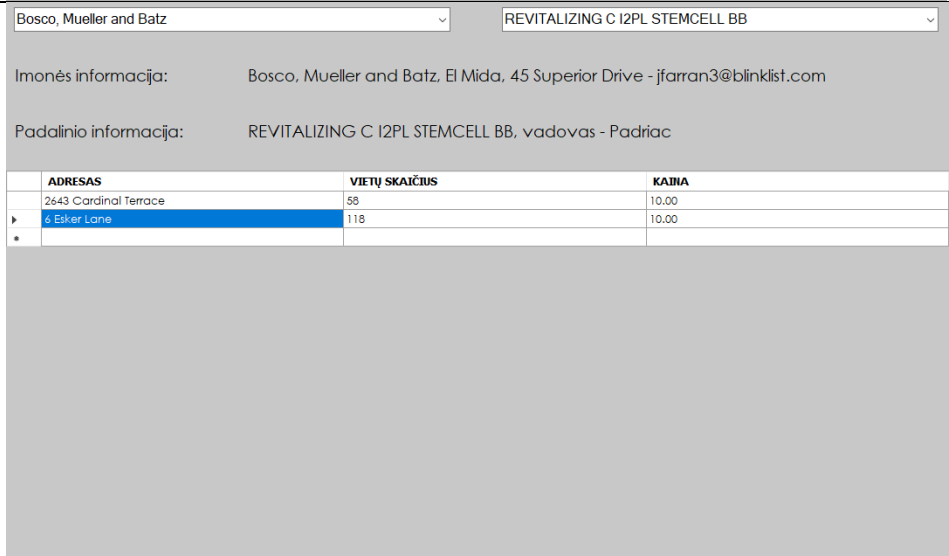
### 3. TESTAVIMAS

Įrenginys buvo testuotas naudojant rankinį testavimą

#### 3.1. Testavimo pavyzdžiai ir rezultatai:

##### 3.1.1. Aplikacijos testavimas

Buvo testuojama kaip programa tikrina visas klaidas ir ar teisingai apdoroja duomenis ir užklausia duombazę teisingai.

Testavimo nuotraukos	Paiškinimas
	Čia matote aplikacijoje suvestą neteisingą kodą. Programa neatvaizduoja jokios informacijos jeigu kodas yra blogas.
	Čia matote jeigu aplikacijoje mokėjimų formoje suvestas teisingas kodas. Programa atvaizduoja detalą informaciją apie mašiną, stovėjimo laiką, bendrą kainą.
	Čia matote užklausą datalesnės informacijos skirsnyje aplikacijoje, pirmame Dropdown menų pasirenkama Įmonė, atrame pasirenkamas padalinys. Kai tie pasirinkti aplikacija parodo ir gerai atvaizduoja visą esamą informaciją apie juos bei aikšteles.



1H24H7D1M1YALL

Date	Usage
2019-05-12	2
2019-05-15	10
2019-05-18	2
2019-05-21	3
2019-05-24	20
2019-05-27	40
2019-05-30	55
2019-06-02	5
2019-06-05	15
2019-06-08	20

Čia matote statistikos dalį aplikacijos, kuri parodo kiek automobilių naudojosi paslaugomis per tam tikrai laiko tarpą.

Automobilių info

NUMERIS
GGG111
GGG222
GGG333
BAD696
HH00E
DDS38
KPE425
LB0685
LB06854
ABC143
ABC413
ABC223
LB06851
LB06852
LB06853
B0685
LB0685
DDS38
JFD444
JFD444

Numeris: BAD696

Būseną: Stovi

Adresas: 0 Valley Edge Hill

Paskutiniame skirsnyje aplikacijos galite pasižiūrėti informacija apie kiekvieną automobilį ir kokioje aikštelė jis stovi.

Testavimo rezultatai – Vartotojo pusėje aplikacija suvedus teisingą kodą rodo gerą informaciją, bei susimokėjus duomenų bazėje atnaujinimi teisingi laukai. Administratorius pusėje teisingai parodo informaciją apie visas įmonės padalinius ir kita.

### 3.1.2. Įtaiso testavimas

Naudojant nuotraukas rastas iš interneto buvo paleista programa ir žiūrima ar teisingai ir kiek teisingai atpažysta numerį bei ar teisinga sąsaja su duomenų baze.

<b>Nautraukos naudotos testavimui</b>	<b>Įtaiso atsakas</b>
---------------------------------------	-----------------------



Numeriai sekmingai nuskaityti  
Duomenys sekmingai pridėti  
(Kuria nauja)

Jusu numeris : FDT792. Aciu,  
kad naudojates musu  
paslaugomis.

Likusių vietų skaičius: 125

Numeriai sekmingai nuskaityti  
Jusu kodas :YHXXHCHAEMO ,  
prasome apmokėti norint  
išvaziuoti

Sekmingai apmokejote už  
stovėjimo laiką!

Jusu numeris : FDT792. Aciu,  
kad naudojates musu  
paslaugomis.

Likusių vietų skaičius: 125

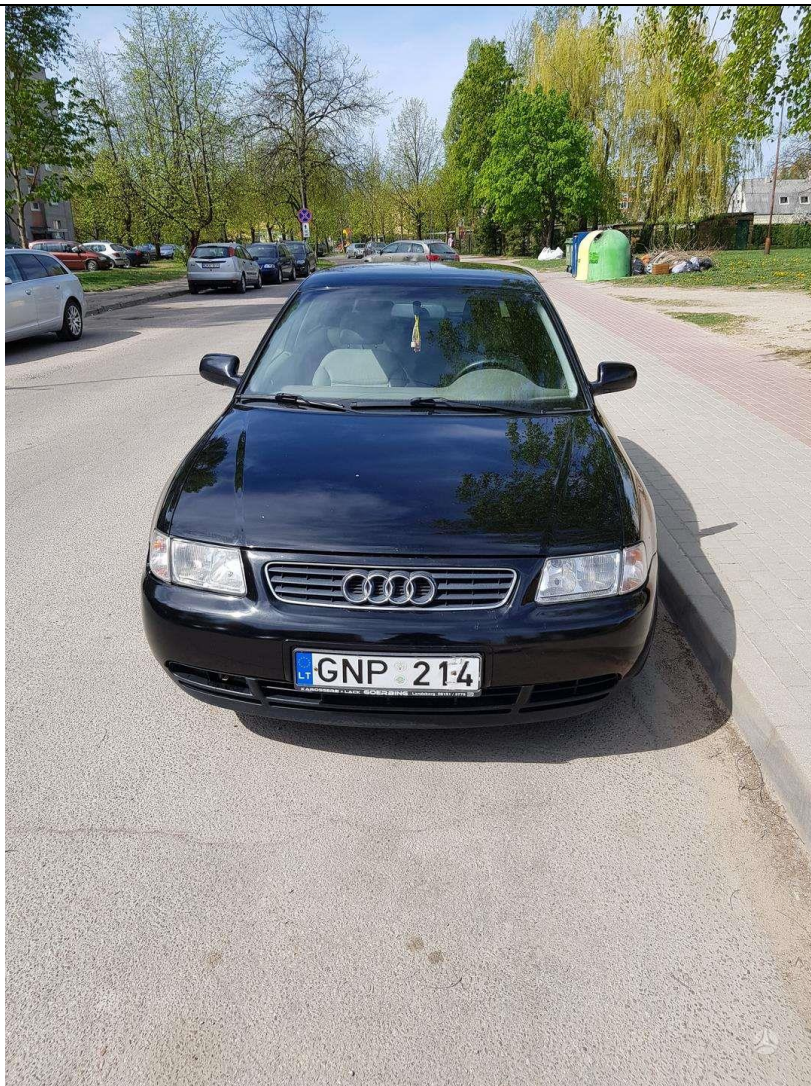
Duomenys sekmingai atnaujinti  
(Išvaziuoja)

Numeriai sekmingai nuskaityti  
Duomenys sekmingai atnaujinti  
(Išvaziuoja)

Jusu numeris : FDT792. Aciu,  
kad naudojates musu  
paslaugomis.

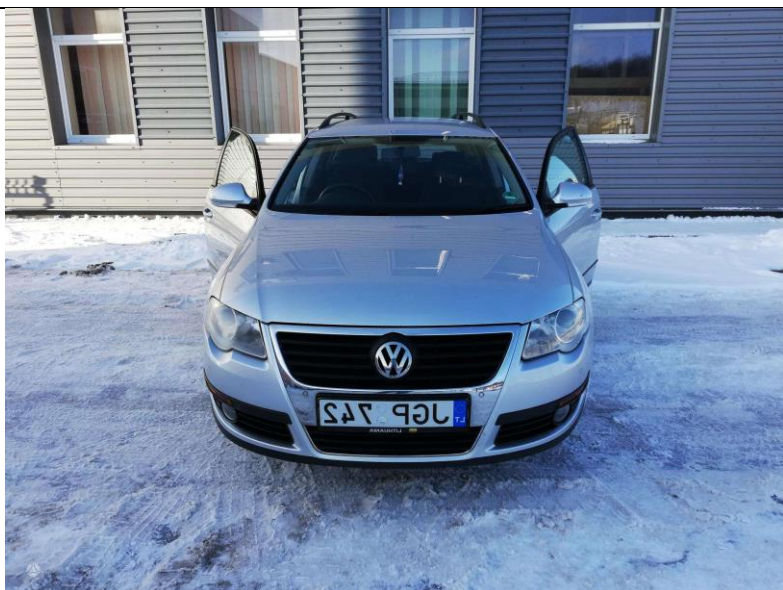
Likusių vietų skaičius: 124





Numeriai sekmingai nuskaityti  
Duomenys sekmingai pridėti  
(Kuria nauja)  
Jūsų numeris : GNP214. Aciu,  
kad naudojates musu  
paslaugomis.  
Likusiu vietu skaicius: 123

Numeriai sekmingai nuskaityti  
Jūsų kodas :SH1KMXJ7G9 ,  
prasome apmoketi norint  
isvaziuoti  
Neapmokejote, bandykite dar  
karta



Numeriai sekmingai nuskaityti  
Duomenys sekmingai pridėti  
(Kuria nauja)  
Jūsų numeris : 8. Aciu, kad  
naudojates musu paslaugomis.  
Likusiu vietu skaicius: 122

## **4. DOKUMENTACIJA NAUDOTOJUI**

### **4.1. Programos diegimas**

1. Sukompiliuoti programą bei paleisti naudojant Visual studio
2. Įjungti Raspberry Pi3, prijungti kamera bei būti pasiruošus Python kodą
3. Paleisti Python kodą, nukreipti kamera į nuotrauka/mašina ir laukti kol sistema nufotografuos
4. Jei įrenginys sėkmingai nufotografuoja ir parašo atitinkamą numerį jis yra paruoštas naudojimui.

### **4.2. Klaidos ir jų padarinių atstatymas**

1. Įrašius neteisingą kodą į programos meniu gali rodyti ne jūsų numerį, tai atsitinka nes blogai suvedėte kodą, patikrinkite kokį kodą sugeneravo Raspberry Pi ir teisingai jį nurašykite.
2. Jei įrašius kodą paspaudžiate „Sumokėti“ be rodo kad jūs „Vis dar nepasiruošę“, reiškia kad jus įrenginys nuskaitė blogai ir reikia bandyti dar kartą arba jūs dar nenufotografavote antrą kartą, kad pakeistų būseną.

### **4.3. Sistemos naudojimas**

1. Sėkmingai įdiegus sistema yra galimybė rinktis su techninę įranga arba skaityma iš failo arba fotografavimą, taip padaryti įmanoma jeigu ištrynete python kode tam tikras eilutes ir atkomentavote tą ką reikia.
2. Nukreipus kamerą į nuotrauką/mašina turite fotografuoti tol kol parašys kad jūsų numeris yra nuskanuotas
3. Aplikacija galima šiuo metu kaip administratoriui ir vartotojui vienu metu, galima peržiūrėti statistika bei detalę informaciją bei susimokėti paspaudus atitinkamą mygtuką.

### **4.4. Sistemos tipinės funkcijos**

Vartotojas gali fotografuoti mašinas ir įrašinėti duomenis į duomenų bazes.

Vartotojas gali „Sumokėti“ už savo stovėjimą ir toliau lankytis aikštelėse.

Administratorius gali tikrinti detalę informacija apie transporto priemones, aikšteles, įmones, padalius.

Administratorius gali pažiūrėti kiek ir koku laiku buvo aplankyta tam tikra mašina ir ją stebėti laiko diagramoje.

### **4.5. Techninės įrangos konfigūracija**

Raspberry Pi visi standartiniai nustatymai, reikalingas įdiegimas Pi cam norint naudoti kamera.

Kompiuteris norint paleisti programa turi atitikti minimalius Visual Studio reikalavimus, bei galėti kompiliuoti programos tekstą.

### **4.6. Sistemą sudarantys failai**

1. Duomenų bazėje esantys SQL failai
2. Aplikacijos sukurti programiniai failai per Visual Studio

## 5. PROJEKTAVIMO VALDYMAS IR EIGA

Procesui organizuoti taikėmė Agile metodiką. Mūsų komandą sudarė 3 nariai, jų sąrašas pateiktas įvade.

Projektą sudarė sprintai, kiekvieno sprint metu buvo suplanuotos atliktos tam tikros funkcijos ir kiekvieno sprinto metu buvo stengtasi juos atlikti.

### 5.1. Sprintai

1 Sprinto metu atlikti darbai -

**TASKS (3/3)**

☒ Funkciniai, nefunkciniai reikalavimai

☒ backlog

☒ github

2 Sprinto metu atlikti darbai –

**TASKS (4/4)**

☒ Sukurti raidžių atpažinimą

☒ Atvaizduoti tekstinę formą

☒ Išsaugoti duomenų bazėje

☒ Kameros veikimo pritaikymas

3 Sprinto metu atlikti darbai –

**TASKS (2/2)**

☒ Išvažiuojančių mašinų patikrinimas

☒ Kainos pritaikymas pagal laiką

4 Sprinto metu atlikti darbai –

**TASKS (4/4)**

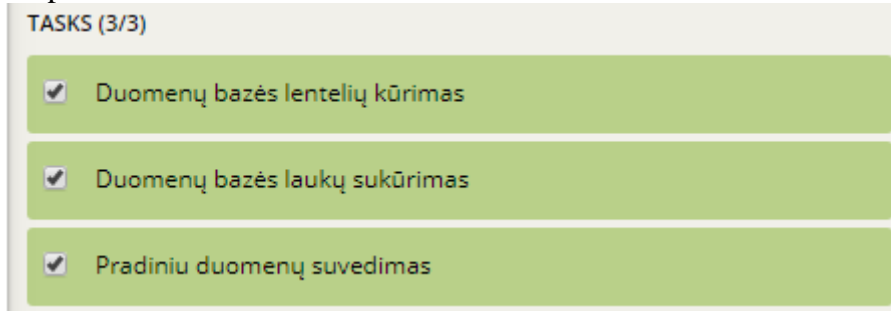
☒ Funkcijų reikalavimų parengimas

☒ Projekto aprašymas

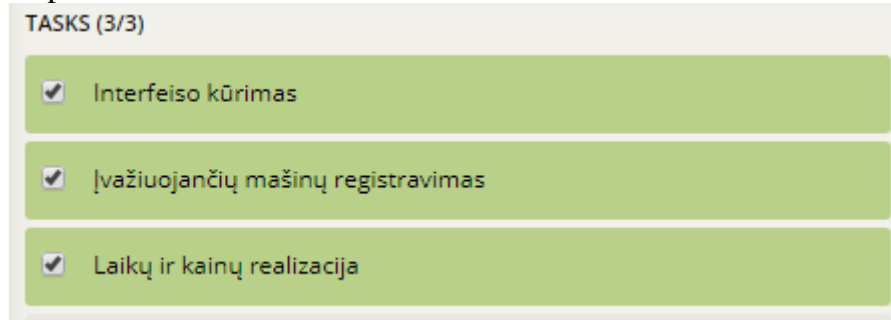
☒ Techninio pasiūlymo parengimas

☒ Nefunkcinių reikalavimų parengimas

5 Sprinto metu atlikti darbai –

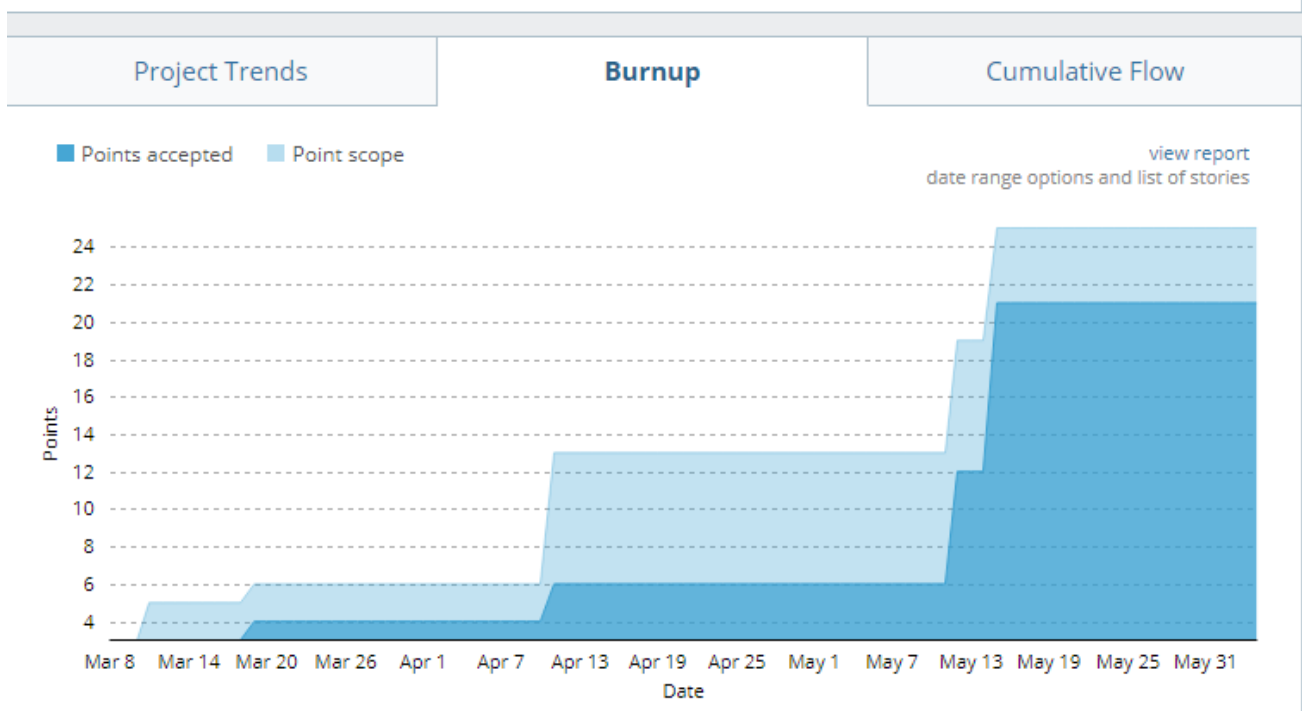


6 Sprinto metu atlikti darbai –



## 5.2. Burndown grafikas

### NAS Overview



## 6. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

- Susipažinta su programinės įrangos kūrimo procesu Agile
- Susipažinta su techninės įrangos programavimo principais
- Susipažinta su Raspberry Pi3 ir jo naudojimo principo
- Susipažinta plačiau su C# kalba bei Python
- Įvairių sistemų kurimo sudėtingumas
- Įrenginiui sukurti reikalaujama daug žmogiškųjų ir laiko išteklių
- Norint taisyklingiau atvaizduoti numerius reiktu geresnės kameros bei geresnės įrangos

## 7. LITERATŪRA

OPEN ALPR: <https://www.openalpr.com/>

Saugykla: <https://github.com>

Vaizdo mokymosi informacija: <https://www.youtube.com/>