

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](#_Toc11045936)

[1. Analizė 4](#_Toc11045937)

[1.1. Techninis pasiūlymas 4](#_Toc11045938)

[1.1.1. Sistemos apibrėžiams 4](#_Toc11045939)

[1.1.2. Sistemos aktualumas ir nauda 4](#_Toc11045940)

[1.2. Galimybių analizė 4](#_Toc11045941)

[2. projektas 5](#_Toc11045942)

[2.1. Reikalavimų specifikacija 5](#_Toc11045943)

[2.1.1. Sistemos funkcijos 5](#_Toc11045944)

[2.1.2. Apribojimai 5](#_Toc11045945)

[2.1.3. Vartotojo sąsajos specifikacija 5](#_Toc11045946)

[2.1.4. Realizacijai keliami nefunkciniai reikalavimai 5](#_Toc11045947)

[2.1.5. Technine specifikacija 5](#_Toc11045948)

[2.2. Projektavimo technologijos 5](#_Toc11045949)

[2.2.1. Projektavimo įrankiai, programavimo kalbos 5](#_Toc11045950)

[2.3. Sistemos projektas 6](#_Toc11045951)

[2.3.1. Įrenginiui naudoti įtaisai 6](#_Toc11045952)

[2.3.2. Numerių atpažinimo algoritmas 7](#_Toc11045953)

[3. Testavimas 8](#_Toc11045954)

[3.1. Testavimo pavyzdžiai ir rezultatai: 8](#_Toc11045955)

[3.1.1. Aplikacijos testavimas 8](#_Toc11045956)

[3.1.2. Įtaiso testavimas 9](#_Toc11045957)

[4. Dokumentacija naudotojui 10](#_Toc11045958)

[4.1. Programos diegimas 10](#_Toc11045959)

[4.2. Klaidos ir jų padarinių atstatymas 10](#_Toc11045960)

[4.3. Sistemos naudojimas 10](#_Toc11045961)

[4.4. Sistemos tipinės funkcijos 10](#_Toc11045962)

[4.5. Technines įrangos konfiguracija 10](#_Toc11045963)

[4.6. Sistemą sudarantys failai 10](#_Toc11045964)

[5. Projektavimo valdymas ir Eiga 11](#_Toc11045965)

[5.1. Sprintai 11](#_Toc11045966)

[5.2. Burndown grafikas 12](#_Toc11045967)

[6. Rezultatų apibendrinimas ir išvados 13](#_Toc11045968)

[7. Literatūra 14](#_Toc11045969)

# Įvadas

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, įrašytų į duomenų bazę bei prijungtų bendrai prie sistemos.

Projekta sugalvojomę 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 galėtu valdyti bei žiūrėti duomenis iš duomenų bazės
* Įgyvendinti programa su Raspberry Pi, kuri galėtu nufotogravuoti 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.

# Analizė

## Techninis pasiūlymas

### 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

### 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 aikštelė naudoti būtent šią sistemą.

## Galimybių analizė

Įgyvendinant šį projektą turėjome nuspręsti, ką naudoti vienas iš projekto narių turėjo techninę įranga iš kitų projektų. Taigi todėl pasiėmeme Raspberry Pi ir kūriame 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.

# projektas

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

## Reikalavimų specifikacija

### Sistemos funkcijos

* Įrenginys nuskaito/nufotografuoja automobilio numerį.
* Įrenginys patikrina ar yra automobilis duomenų bazėje, jeigu ne prideda, jeigu taip prideda prie stovejimo laikų paplidomą informacija.
* Vartotojas gali atsidaryti telefoną/kompiuterį ir patikrinti savo automobilio būsena bei jeigu išvažinėja apmokėti.
* Administratorius gali prieiti prie detalesnės informacijos apie visas įmones, padalius, aikšteles bei automobilius.
* Administratorius gali pažiūrėti statistiką apie kiekvieną aikštelę.

### 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 vieta užima automobilis aikštelėje

### Vartotojo sąsajos specifikacija

Įrenginio vartotojo sąsaja turi atvaizduoti galimų klaidų pranešimu, bei mašinos statusą, per unikalų sugeneruota koda sukurti duombazės query ir atvaizduoti informacija apie mašinos numerį, stovėjimo būseną, pradžios ir pabaigos laiką, ir bendrą suma kiek turi mokėti.

### Realizacijai keliami nefunkciniai reikalavimai

|  |  |
| --- | --- |
| **Reikalavimas** | **Kriterijus** |
| Programavimo kalba turi turėti griežtus kintamųjų tipus | Integruota programinės įrangos aplinka, tipų klaidų pranešimai kompiliavimo metu |
| Programvimo kalba turi būti objektišai orientuota | Kiekvienas periferinis įrenginys yra programuojamas kaip atskiras objektas |
| Įtaisas turi turėti kamera | Norint fotografuoti automobilių numerius gyvai reikalinga kamera. |
| Naudojamas įrenginys turi turėti prieigą prie interneto. | Turi būti priėimas prie interneto, kad vartotojas/adminstratorius duoti užklausas duomenų bazėj |

### Technine specifikacija

Norint kad įrenginys veiktu jis privalo galėti kompiliuoti ir paleisti Python kodą bei turi kamera kuri galėtu fotografuoti. Prieiga prie interneto.

## Projektavimo technologijos

Įrenginiui suprojektuoti nebuvo naudojamas specifinis architektūros projekatvimo modelis.

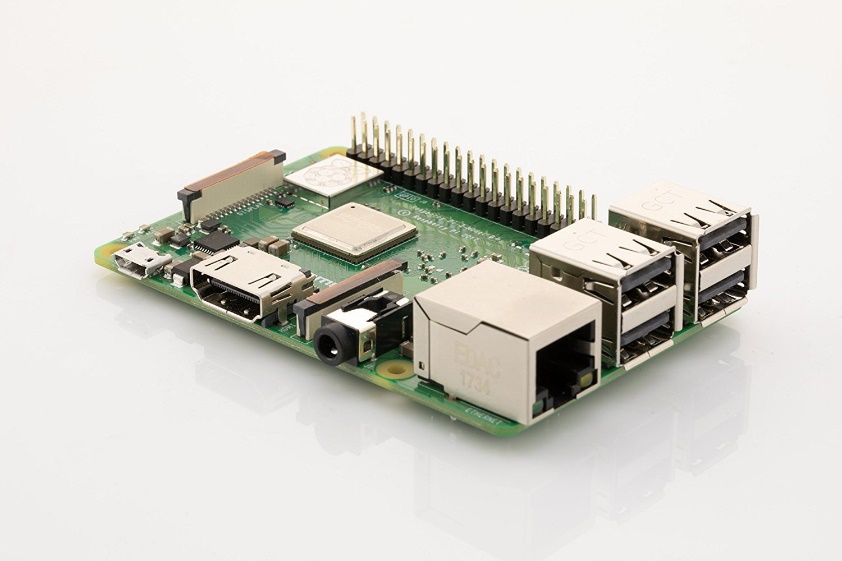
### Projektavimo įrankiai, programavimo kalbos

Sistemos kūrimui naudojome programą Visual Studio 2017. Su šia programa įgyvendinome 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.

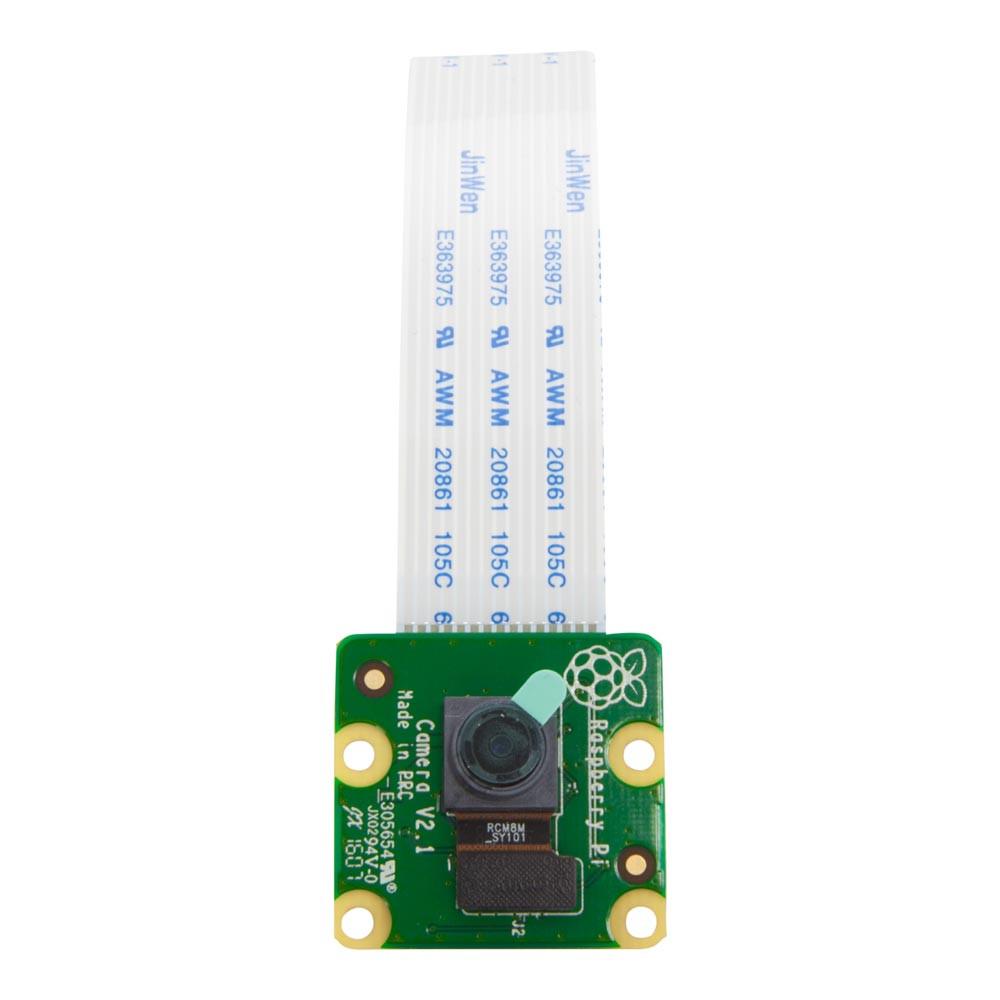
## Sistemos projektas

### Įrenginiui naudoti įtaisai

Raspberry Pi 3



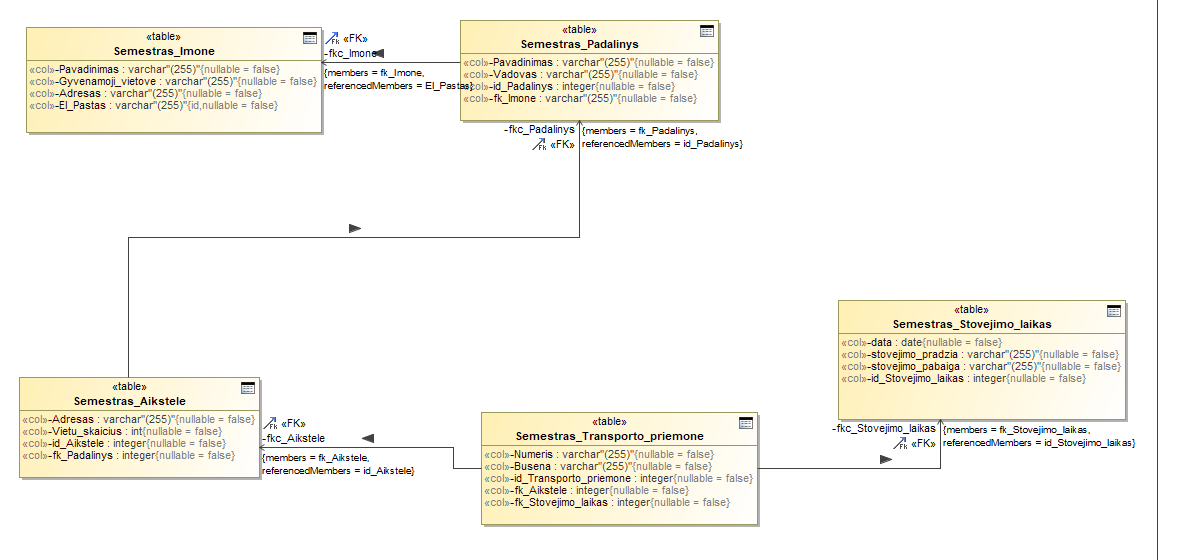
Raspberry Pi Camera Module V2



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

Klasių diagrama:

Semestras\_Imone – Talpina įmonės infromacija



Semestras\_Padalinys – Talpina informacija apie padalinį priklausantį įmonėj

Semestras\_Aikštelė – Skirta informacijai apie aikštelę kuri priklauso padaliniui

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

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

### Numerių atpažinimo algoritmas

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

# Testavimas

Įrenginys buvo testuotas naudojant rankinį testavimą

## Testavimo pavyzdžiai ir rezultatai:

### Aplikacijos testavimas

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

|  |  |
| --- | --- |
| **Testavimo nuotraukos** | **Paaiškinimas** |
| https://i.imgur.com/9ospkGi.png | Čia matote aplikacijoje suvestą neteisingą kodą. Programa neatvaizduoja jokios informacijos jeigu kodas yra blogas. |
| https://i.imgur.com/xNULiXt.png | Čia matote jeigu aplikacijoje mokėjimų formoje suvestas teisingas kodas. Programa atvaizduoja detalia informacija apie mašiną, stovėjimo laiką, bendrą kainą. |
| https://i.imgur.com/7braHEv.png | Čia matote užklausą datalesnės informacijos skirsnyje aplikacijoje, pirmame Dropdown menų pasirenkama Įmonė, atrame pasirenkamas padalinys. Kai tie pasirinkti aplikacija parodo ir gerai atvaizduoja visą esama informacija apie juos bei aikšteles. |
| https://i.imgur.com/plH2vPb.png | Čia matote statistikos dalį aplikacijos, kuri parodo kiek automobilių naudojosi paslaugomis per tam tikrai laiko tarpą. |
| https://i.imgur.com/GzpDcIO.png | Paskutiniame skirsnyje aplikacijos galite pasižiurėti informacija apie kiekvieną automobilį ir kokioje aikštelė jis stovi. |

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

### 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.

|  |  |
| --- | --- |
| **Nautraukos naudotos tęstavimui** | **Įtaiso atsakas** |
| **C:\Users\biweg\AppData\Local\Microsoft\Windows\INetCache\Content.Word\masinav1.jpg** | Numeriai sekmingai nuskaityti  Duomenys sekmingai prideti (Kuria nauja)  Jusu numeris : FDT792. Aciu, kad naudojates musu paslaugomis.  Likusiu vietu skaicius: 125  Numeriai sekmingai nuskaityti  Jusu kodas :YHXHCHAEMO , prasome apmoketi norint isvaziuoti  Sekmingai apmokejote uz stovejimo laika!  Jusu numeris : FDT792. Aciu, kad naudojates musu paslaugomis.  Likusiu vietu skaicius: 125  Duomenys sekmingai atnaujinti (Isvaziuoja)  Numeriai sekmingai nuskaityti  Duomenys sekmingai atnaujinti (Ivaziuoja)  Jusu numeris : FDT792. Aciu, kad naudojates musu paslaugomis.  Likusiu vietu skaicius: 124 |
|  | Numeriai sekmingai nuskaityti  Duomenys sekmingai prideti (Kuria nauja)  Jusu numeris : GNP214. Aciu, kad naudojates musu paslaugomis.  Likusiu vietu skaicius: 123  Numeriai sekmingai nuskaityti  Jusu kodas :SH1KMXJ7G9 , prasome apmoketi norint isvaziuoti  Neapmokejote, bandykite dar karta |
| C:\Users\biweg\Desktop\masinav3.jpg | Numeriai sekmingai nuskaityti  Duomenys sekmingai prideti (Kuria nauja)  Jusu numeris : 8. Aciu, kad naudojates musu paslaugomis.  Likusiu vietu skaicius: 122 |

# Dokumentacija naudotojui

## 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.

## Klaidos ir jų padarinių atstatymas

1. Įrašius neteisingą kodą į programos menių gali rodyti ne jūsų numerį, tai atsitinka nes blogai suvedėte kodą, patikrinkite kokį kodą sugeneravo Raspberry Pi ir teisingai jį nurašykite.
2. Jei įrašius koda 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.

## Sistemos naudojimas

1. Sėkmingai įdiegus sistema yra galimybe 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.

## 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 kokiu laiku buvo aplankyta tam tikra mašina ir ją stebėti laiko diagramoje.

## Technines įrangos konfiguracija

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.

## Sistemą sudarantys failai

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

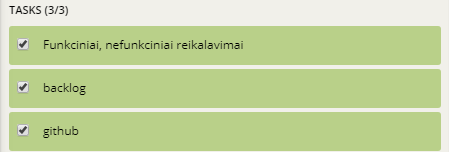
# Projektavimo valdymas ir Eiga

Procesui organizuoti taikėmė Agile metodiką. Mūsų komandą sudarė 3 nariai, jū sarašas pataikytas Įvade.

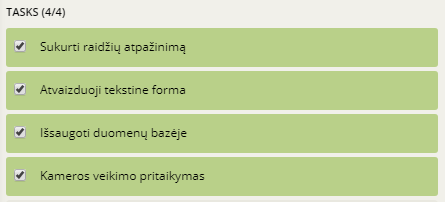
Projektą sudarė sprintai, kiekvieno sprint metu buvo suplanuotos atlikotos tam tikros funkcijos ir kiekvieno sprinto metu buvo stengtasi juos atlikti.

## Sprintai

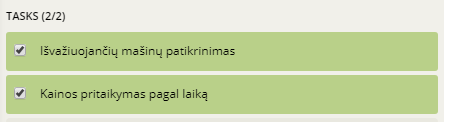
1 Sprinto metu atlikti darbai -



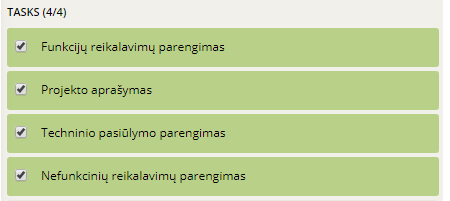
2 Sprinto metu atlikti darbai –



3 Sprinto metu atlikti darbai –



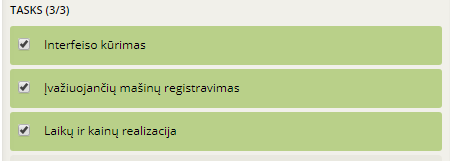
4 Sprinto metu atlikti darbai –



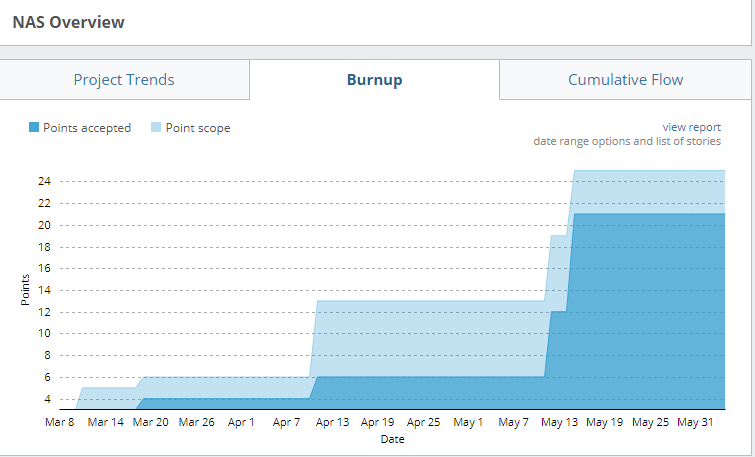
5 Sprinto metu atlikti darbai –



6 Sprinto metu atlikti darbai –



## Burndown grafikas



# Rezultatų apibendrinimas ir išvados

* Susipažinta su programinės įrangos kūrimo procesu Agile
* Susipažinta su technines 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

# Literatūra

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

Saugykla: [https://github.com](https://github.com/)

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