****

**Kauno technologijos universitetas**

Informatikos fakultetas

**Realaus laiko stalo futbolo sekimo sistema**

Baigiamasis bakalauro studijų projektas

|  |
| --- |
|  |
| **Matas Kulkovas**  Projekto autorius |
|  |
| **dokt.** **Karolis Ryselis**  Vadovas |
|  |

**Kaunas, 2019**

****

**Kauno technologijos universitetas**

Informatikos fakultetas

**Realaus laiko stalo futbolo sekimo sistema**

Baigiamasis bakalauro studijų projektas

Programų sistemos (612I30002)

|  |  |
| --- | --- |
|  |  |
| **Matas Kulkovas**  Projekto autorius | (parašas) (data) |
|  |  |
| **dokt.** **Karolis Ryselis**  Vadovas | (parašas) (data) |
|  |  |
| **Pareigų sutrumpinimas Vardenis Pavardenis**  Recenzentas / Recenzentė | (parašas) (data) |
|  |  |

**Kaunas, 2019**

****

**Kauno technologijos universitetas**

Informatikos fakultetas

Matas Kulkovas

**Realaus laiko stalo futbolo sekimo sistema**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Mato Kulkovo, baigiamasis projektas tema „Realaus laiko stalo futbolo sekimo sistema“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| (vardą ir pavardę įrašyti ranka) |  | (parašas) |

Kulkovas, Matas. Realaus laiko stalo futbolo sekimo sistema. Bakalauro studijų baigiamasis projektas / dokt. Karolis Ryselis; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Programų sistemos.

Reikšminiai žodžiai: Stalo futbolas, vaizdo analizė, realus laikas, API.

Kaunas, 2019. XX p.

Santrauka

Darbe pristatoma realaus laiko stalo futbolo sekimo sistema. Užsakovo įmonėje norima pridėti automatizuotą duomenų surinkimą apie stalo futbolo rungtynes. Pelnyti įvarčiai, sekant kamuoliuko poziciją, bei žaidėjų pozicijos, turėtų būti pranešami ir vaizduojami realiu laiku. Taip pat sistema turėtų padėti įmonėje rengti stalo futbolo turnyru. Šios sistemos tikslas – sistema, kuri naudodama vaizdo analizę sektų stalo futbolo rungtynių eigą, surinktus duomenis išsaugotų ir pateiktų juos, bei padėtų valdyti stalo futbolo turnyrus ir įtrauktų žaidėjus juose dalyvauti.

Įvade apžvelgiamas darbo aktualumas užsakančioje įmonėje, iškeliami tikslai ir uždaviniai. Problematikos ir aktualumo tema plėtojama analizės dalyje, kur apžvelgiamas bendras sistemos veiklos tikslas, pagrįstumas. Šioje dalyje taip pat nagrinėjami prototipai, rinkoje esantys konkurentai bei sistemos apimtis ir jos realizavimo galimybės.

Projekto dalyje pateikiama sistemos reikalavimų specifikacija: jos funkciniai ir nefunkciniai reikalavimai, apribojimai, reikalavimai techninei įrangai. Vėliau apžvelgiama metodika bei technologijos, naudotos projekto sudarymo metu. Galiausiai pateikiamas sistemos projektas bei jo analizė.

Testavimo dalyje apžvelgiama testavimo metodika bei jo planas. Toliau detaliai nagrinėjama kiekvieno testavimo tipo specifika, taikytos technologijos bei testavimo rezultatai.

Darbe taip pat pateikiama sistemos dokumentacija. Joje aprašoma, kaip reikia naudotis sistemos funkcijomis, jos diegimo bei priežiūros instrukcija. Darbo pabaigoje aptariami sistemos kūrimo rezultatai bei išvados, prie kurių buvo prieita kūrimo metu,

Kulkovas, Matas. Real time foosball tracking system. Bachelor's  Final Degree Project / supervisor assoc. prof. practitioner Karolis Ryselis; Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer Sciences, Software Systems.

Keywords: Foosball, video analysis, real-time, API.

Kaunas, 2019. Number of pages.

Summary

A real time foosball tracking system is presented in the thesis. The client company requires an automated solution for collecting data, using video recording equipment. Scored goals, detected by tracking position of the ball, and positions of players, should be reported and displayed in real time. The system should also aid in creating and managing foosball tournaments in the company. The goal of the system, is to track and report the state of the games and to collect data about them and also to help manage the tournaments in the company.

The paper begins with the introduction where the thesis’ relevance for the client company and main problems and task are reviewed. The topic is further investigated in the analysis section, where the system under creation is examined and analyzed. Furthermore, the prototypes and main competitors are reviewed and scope and implementation possibilities are considered.

System’s modelling section begins with the requirements’ analysis: functional and non-functional requirements, restrictions and technical requirements are reviewed. Then the paper looks in detail into modelling methods and technologies that were used for modelling. Finally, system’s project and it’s analysis are presented

Testing section includes testing method and it’s plan. Further on, every type of performed, it’s tools and results are looked more in-depth.

System’s documentation is included in the paper as well. It contains a manual for users, administrators and the deployment and maintenance guides for the system.

The paper is ended with results and conclusions, that occurred while the system was in development.

Turinys

[Lentelių sąrašas 7](#_Toc3806737)

[Paveikslų sąrašas 8](#_Toc3806738)

[Santrumpų ir terminų sąrašas 9](#_Toc3806739)

[Įvadas 10](#_Toc3806740)

[1. Analizė 11](#_Toc3806741)

[1.1. Techninis pasiūlymas 11](#_Toc3806742)

[1.1.1. Sistemos apibrėžimas 11](#_Toc3806743)

[1.1.2. Bendras veiklos tikslas 11](#_Toc3806744)

[1.1.3. Sistemos pagrįstumas 11](#_Toc3806745)

[1.1.4. Konkurencija rinkoje 11](#_Toc3806746)

[1.1.5. Prototipai ir pagalbinė informacija 11](#_Toc3806747)

[1.1.6. Ištekliai, reikalingi sistemai sukurti 13](#_Toc3806748)

[1.2. Galimybių analizė 14](#_Toc3806749)

[1.2.1. Techninės galimybės 14](#_Toc3806750)

[1.2.2. Vartotojų pasiruošimo analizė 14](#_Toc3806751)

[2. Projektas 16](#_Toc3806752)

[2.1. Reikalavimų specifikacija 16](#_Toc3806753)

[2.1.1. Komercinė specifikacija 16](#_Toc3806754)

[2.1.2. Sistemos funkcijos 16](#_Toc3806755)

[2.1.3. Vartotojo sąsajos specifikacija 17](#_Toc3806756)

[2.1.4. Realizacijai keliami reikalavimai 17](#_Toc3806757)

[2.1.5. Techninė specifikacija 17](#_Toc3806758)

[2.2. Projektavimo metodai 18](#_Toc3806759)

[2.2.1. Projektavimo valdymas ir eiga 18](#_Toc3806760)

[2.2.2. Projektavimo technologija 18](#_Toc3806761)

[2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos 18](#_Toc3806762)

[2.3. Sistemos projektas 19](#_Toc3806763)

[2.3.1. Statinis sistemos vaizdas 19](#_Toc3806764)

[2.3.2. Dinaminis sistemos vaizdas 22](#_Toc3806765)

[3. Testavimas 29](#_Toc3806766)

[3.1. Testavimo planas 29](#_Toc3806767)

[3.2. Testavimo kriterijai 29](#_Toc3806768)

[3.3. Komponentų testavimas 29](#_Toc3806769)

[3.4. Integracinis testavimas **Error! Bookmark not defined.**](#_Toc3806770)

[3.5. Vartotojo sąsajos testavimas 32](#_Toc3806771)

[4. Dokumentacija naudotojui 34](#_Toc3806772)

[4.1. Apibendrintas sistemos galimybių aprašymas 34](#_Toc3806773)

[4.2. Vartotojo vadovas 34](#_Toc3806774)

[4.3. Diegimo vadovas 41](#_Toc3806775)

[4.4. Administravimo vadovas 41](#_Toc3806776)

[Rezultatai ir išvados 43](#_Toc3806777)

[Literatūros sąrašas 44](#_Toc3806778)

[Priedai 45](#_Toc3806779)

[1 priedas. Priedo pavadinimas 45](#_Toc3806780)

Lentelių sąrašas

[1.1 lentelė. **Sistemų palyginimo lentelės pavyzdys** 13](#_Toc8855272)

[**2 lentelė.** Pagrindiniai baigiamojo projekto stiliai ir jų aprašymai 45](#_Toc8855273)

Paveikslų sąrašas

[**1 pav.** Sistemos panaudojimo atvejų diagrama 16](#_Toc8855277)

Santrumpų ir terminų sąrašas

**Santrumpos:**

Doc. – docentas;

Lekt. – lektorius;

Prof. – profesorius.

**Terminai:**

**Saityno analitika** – lorem ipsum dolor sit amet, eam ex decore persequeris, sit at illud lobortis atomorum. Sed dolorem quaerendum ne, prompta instructior ne pri. Et mel partiendo suscipiantur, docendi abhorreant ea sit. Recteque imperdiet eum te.

**Tinklaraštis** – lorem ipsum dolor sit amet, eam ex decore persequeris, sit at illud lobortis atomorum. Sed dolorem quaerendum ne, prompta instructior ne pri. Et mel partiendo suscipiantur, docendi abhorreant ea sit. Recteque imperdiet eum te.

Beje, darbe rekomenduojame pateikti tik svarbesnes ir mažiau žinomas santrumpas bei terminus (tarkime tokių santrumpų kaip HTML, PC, IT paaiškinti nereikia)

Įvadas

Įmonėje nuspręsta pradėti duomenų, apie stalo futbolo rungtynes, automatizavimą. Automatizuotas įvarčių, kamuoliuko bei stalo figurėlių pozicijos sekimas bei saugojimas duomenų saugykloje leistų tiksliau įvertinti žaidėjų gebėjimus ir rezultatų istoriją. Įvertinta, jog patogiausia įvairius duomenis apie rungtynes surinkti bus vaizdo kameros pagalba. Rungtynių rezultatus ir kitus duomenis peržiūrėti patogiausia per saityno sąsają, kuri bus prieinama įmonėms darbuotojams.

Darbo tikslas – pateikti duomenis apie stalo futbolo rungtynes, sukuriant duomenų surinkimo, analizavimo ir pateikimo sistemą. Šio tikslo įgyvendinimui iškelti uždaviniai:

1. Atlikti reikalavimų analizę.
2. Ištirti rinkoje esančias panašias sistemas.
3. Sudaryti sistemos projektą.
4. Realizuoti bei ištestuoti sistemą.
5. Dokumentuoti sukurtą sistemą.

Darbas susideda iš 4 dalių: analizės, projekto, testavimo bei dokumentacijos. Analizės skyriuje apžvelgiamas projekto tikslas, pagrįstumas, konkurentų apžvalga, įvertinamos realizavimo ištekliai ir galimybės. Projekto skyriuje pristatomi sistemos reikalavimai, naudojami projektavimo metodai bei pats sistemos projektas.

Projekto apimtis vertinta projektą išskaidžius į 3 dalis: vaizdo kameros duomenų surinkimą ir apdorojimą bei perdavimą į saugyklą, saugyklos serveris su savo API, bei saityno puslapis. Projektas įvertintas virš 500 valandų. Atsižvelgus į tai, sistema bus kuriama vieno žmogaus. Sistema bus testuojami lokalioje ir produkcinėje aplinkoje.

# Analizė

## Techninis pasiūlymas

### Sistemos apibrėžimas

Stalo futbolo realaus laiko sistema – tai surenkanti duomenis, juos analizuojanti ir išsauganti, bei stalo futbolo žaidimą papildanti sistema. Stalo futbolo realaus laiko sekimo sistemą sudaro trys dalys – vaizdo sekimo posistemė, serverio posistemė bei serverio klientas. Vaizdo sekimo posistemė, su kameros pagalba, seka stalo futbolo stalą, įrašo bei praneša serveriui apie kamuoliuko poziciją, įvarčius bei žaidėjų pozicijas. Serveris saugo žaidimų bei žaidėjų duomenis.

### Bendras veiklos tikslas

Padėti įmonėje rinkti informaciją apie stalo futbolo rungtynes. Įtraukti daugiau žaidėjų dalyvauti, palengvinant rezultatų skaičiavimą bei dalyvavimą turnyruose. Susipažinti su vaizdo analizės algoritmais bei juos taikyti.

### Sistemos pagrįstumas

Įmonėjė stalo futbolo žaidėjams dažnai kyla daugybė klausimų, pavyzdžiui, koks vidutinis smūgio greitis? Kokia žaidėjų pozicija yra statistiškai geriausia besiginančiam žaidėjui? Į tokius klausimus tiksliai atsakyti žmogui yra sudėtinga, todėl buvo nuspręsta, jog reikia sistemos, kuri galėtų pateikti tokius duomenis, kurie yra konkretūs ir pateikia informacijos apie žaidimo eigą, žaidėjų sugebėjimus, žaidimo stilius ir t.t.

Taip pat rezultatų išsaugojimas šiuo metu vyksta tik turnyrinėse dvikovose, įsirašant rezultatą rankiniu būdu. Tai kartais sukelia diskomfortą bei neturnyrinių dvikovų rezultatų praradimą. Todėl norint išspręsti šias problemas, rezultatų saugojimas turėtų galioti visoms dvikovoms bei būti automatizuotas. Rezultatus taip pat turėtų būti galima nesunkiai peržiūrėti visiems įmonėms darbuotojams, ne tik programuotojams.

Dėl šių priežąsčių, bei įvertinus rinkoje esančius panašius produktus, iš kurių nei vienas pilnai neatitiko reikalavimų, buvo nuspręsta, jog reikia sukurti stalo futbolo realaus laiko sekimo sistemą, kurią sudarys vaizdo analizės posistemė, serveris su duomenų baze, kurioje bus saugomi rezultatai, bei internetis puslapis, naudojantis serverio API.

### Konkurencija rinkoje

Rasti produktų, kurie vykdytų vaizdo analizę ir atliktų žaidimų organizavimo ir rezaltatų saugojimo funkciją – sudėtinga. Egzistuoja nemažai aplikacijų, kurios skirtos rezultatų saugojimui ir pateikimui, turnyrų organizavimui, tačiau jos nerenka duomenų apie žaidimą, rezultatus reikia suvedinėti pačiam. Yra parašyta nemažai straipsnių apie prototipines aplikacijas skirtas stalo futbolo vaizdo sekimui, tačiau jose retai yra dalinamasi programiniu kodu. Taip pat egzsituoja bendro naudojimo pobūdžio video analizės įrankis Amazon „Rekognition“ (1). Tačiau dėl 0.10$ įkainio už analizuojamo vaizdo minutę, šis pasirinkimas buvo atmestas.

Rezultatų sekimui yra nemažai sistemų, tiek mokamų tiek nemokamų. Papraščiausią funkcionalumą išpildanti aplikacija buvo pasirinkta „Kicktrack“. Tai web sąsajoje veikiantis produktas, leidžiantis susikurti savo įmonės grupę, suvedinėti rezultatus ir juos sekti. Tačiau ne visos siūlo turnyrų kūrimą.

Paieškojus GitHub atvirojo kodo sistemoje, buvo aptiktas naudotojo „mtszkw“ projektas „foosball“ (*<https://www.github.com/mtszkw/foosball>***).** Šio analizės įrankio teikiami duomenys yra pakankamai išsamūs planuojamam projektui – aptinkama kamuoliuko pozicija, figurėlių pozicija ant žaidėjų juostos. Tačiau įrankis analizuoja tik įrašytą video medžiagą. Jei modifikavus jo kodą, kad analizuotų vaizdo srautą gyvai, jis veikia mažiau nei 10 kadrų per sekundę ant dedikuoto kompiuterio. Taip pat programinis kodas parašytas su C++ 17. Nors tai yra vaizdo analizės kalbų standartas, užsakovas plėtojimo tikslais pageidavo programinio kodo Python 3 programavimo kalba.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lyginimo kriterijai** | **Kicktrack** | **Kickertool** | **github.com/mtszkw/foosball** | **Kuriama sistema** |
| Žaidėjų išsaugojimas | Realizuota | Realizuota | Nerealizuota | Realizuota |
| Pavienės rungtynės | Realizuota | Nerealizuota | Nerealizuota | Realizuota |
| Turnyrų kūrimas | Nerealizuota | Realizuota | Nerealizuota | Realizuota |
| Žaidėjų pasiekimai | Realizuota | Nerealizuota | Nerealizuota | Nerealizuota |
| Figurėlių aptikimas vaizde | Nerealizuota | Nerealizuota | Realizuota, bet tik vaizdo įrašuose | Realizuota su vaizdo įrašais ir vaizdo transliacijoje apie 30 FPS |
| Kamuoliuko pozicijos aptikimas vaizde | Nerealizuota | Nerealizuota | Realizuota, bet tik vaizdo įrašuose | Realizuota su vaizdo įrašais ir vaizdo transliacijoje apie 30 FPS |
| Duomenų perdavimas iš vaizdo analizės per tinklą | Nerealizuotas | Nerealizuotas | Nerealizuotas | Realizuota su serveriu, į kurį galima prisijungti WebSocket protokolu. |
| Licencijos kaina | Free – nemokama  Stadium – 7$ / mėn  Arena – 29$ / mėn | Nemokama | Nemokama | Nemokama |
| Vaizdo analizės programavimo kalba | Nėra | Nėra | C++ 17 su OpenCV 3.4.1 biblioteka | Python 3.7 su OpenCV 3.4.4 biblioteka |
| Vaizdo iškreipimo panaikinimas | Nėra | Nėra | Yra | Gali būti įtraukta sekančioje versijoje |

Iš konkurentų analizės lentelės matyti, jog kitose sistemose yra realizuota dauguma atskirų sistemos reikalavimų. Saityno įrankiai – Kicktrack, Kickertool turi patogias saityno programas, tačiau rungtynių rezultatus turi sekti patys žaidėjai, taip pat trūksta kai kurio turnyrų funkcionalumo, kuris reikalingas užsakovo įmonėje. Github.com/mtszkw/foosball įrankis, kuris aptinką kamuoliuką ir žaidėjus vaizdo medžiagoje, neveikia realiu laiku, o tik su vaizdo įrašais, taip pat, neturi duomenų perdavimo per tinklą funkcionalumo. Visa tai leidžia manyti, jog kuriama sistema pasižymi itin plačiu funkcionalumu ir geriausiai tenkina užsakovo reikalavimus.

1.1 lentelė. Sistemų palyginimo lentelės pavyzdys

### Prototipai ir pagalbinė informacija

Serveris bei naudotojo sąsaja projektuoti nenaudojant prototipų, tačiau buvo naudojamos kelios bibliotekos. Serverio dalyje buvo naudojama Entity Framework Core 2.2.0 biblioteka operacijoms su duomenų baze. EF Core 2.2.0 padeda išvengti tekstinio užklausų rašymo, užklausas paverčia į operacijas su objektais, taip leidžiant neprisirišti prie konkrečios duomenų bazės bei vykdo visas duomenų bazės užklausas naudodama repozitorijos projektavimo šabloną (angl. Reposity Design Pattern).

Vaizdo analizės posistemėje buvo remtasi kuriant Indianos Universiteto studentų darbo ataskaita [1]. Programos kodas nebuvo pateiktas, tačiau buvo apytiksliai aprašyti vaizdo analizės metodai, kurie pasirodė tinkami ir kurių dauguma buvo remtasi programuojant žaidėjų sekimą. Kamuoliuko sekimui nuo prototipo buvo pakeistas paskutinis algoritmo žingsnis – vietoj šablonų lyginimo (angl. Template matching) buvo pasirinktas kontūro sekimo metodas (angl. Contour following), kuris eksperimentiškai pateikė didesnės greitaveikos ir panašaus tikslumo rezultatus. Taip pat vaizdo analizei buvo naudoja OpenCV 3.4.4. Ji suteikia metodus palengvinančius darbą su vaizdo srauto skaitymu, keletą algoritmų, kurie naudojami analizuoti paveikslėliams.

### Ištekliai, reikalingi sistemai sukurti

Sistemos kurimą sudaro trys didesnės dalys:

1. Vaizdo analizės dalis
2. Naudotojo sąsaja sujungianti vaizdo analizę ir duomenų saugojimo serverį
3. Duomenų saugojimo servisas

Vaizdo analizės dalis yra atsakinga už vaizdo srauto iš kameros apdorojimą ir kamuoliuko bei žaidėjų pozicijos gavimą bei perdavimą prisijungusiam klientui. Kliento funkciją atliks naudotojo sąsaja, kuri surinkusi duomenis bei taip pat grafiškai pateikusi juos naudotojui, perduos duomenų saugojimo servisui.

Buvo įvertinta, jog sistemai sukurti reikės apie 400 darbo valandų. Be automatinių testų serverio dalis turėtų sudaryti apie 1500 C# kodo eilučių, vaizdo analizės posistemėje apie 1000 Python kodo eilučių. Naudotojo sąsajos JavaScript bei HTML ir CSS kodo eilučių skaičius nebuvo prognozuojamas. Apžvelgus reikalingus išteklius su klientu, buvo nuspręsta, jog sistemai sukurti užtenka vieno žmogaus, tačiau bus mažai laiko praleidžiama prie naudotojo sąsajos dizaino.

## Galimybių analizė

### Techninės galimybės

Kodui saugoti naudojama git versijų valdymo sistema. Naujausios produkcinės versijos bei atnaujinimų atšakos talpinamos į GitHub saugyklą.

Sistema bus realizuota naudojant kelis karkasus. Kliento sąsajos daliai bus naudojama React.js biblioteka, kuri šiuo metu yra viena populiariausių ir plačiausiai naudojamų tiek įmonėje, tiek visoje saityno projektų rinkoje. Serveriui kurti bus naudojamas ASP.NET Core 2.2 atvirojo kodo karkasas, kuris taip pat yra plačiai naudojamas įmonėje. ASP.NET Core 2.2 projektai lengvai integruojasi su Microsoft Azure debesų kompiuterijos paslaugomis, tačiau esant poreikiui galima perkelti kaip Docker konteinerį į bet kokį kitą debesų kompiuterijos tiekėją.

Stalo sekimui bus naudojama Logitech C922 web kamera. Ji pasirinkta dėl sąlyginai nedidelių kaštų, greičio bei vaizdo kokybės – iki 60 kadrų per sekunde 720p režimu. Vaizdas apdorojamas kompiuterije su Python 3.7 ir OpenCV 3.4.4. Žemam atsilikimo laikui užtikrinti, kamera prie vaizdo apdorojimo prietaiso, dažniausiai – tam dedikuoto asmeninio kompiuterio arba Raspberry Pi 3 B+, turėtų būti jungiama USB 3.0 jungtimi.

Įvertinus projekto reikalavimus, galima teikti, jog visos techninės galimybės yra išpildytos.

### Vartotojų pasiruošimo analizė

Produktas kuriamas naudojimui bei plėtojimui ateityje. Produkto naudojimui, kai įrašomos vienos rungtynės, bus pakankama mokėti naudotis interneto naršykle bei naudotojo sąsajoje pasirinkti žaidėjus. Už turnyrų moderavimą atsakingas asmuo turės praleisti apie 5 min. susipažindamas su turnyro struktūromis ir sąsajos veikimu.

Tolesnį sąsajos ir serverio plėtojimą turėtų gebėti atlikti dauguma įmonėje dirbančių programuotojų. Norint konfiguruoti vaizdo sekimo posistemę, naudotojas turėtų būti susipažinęs su bazinėmis OpenCV funkcijomis, taip pat mokėti programuoti Python 3 kalba.

## Vaizdo analizės algoritmai

### Bendras analizės veikimas

Vaizdo analizė bus skaidoma į du atskirus algoritmus ir vykdoma tokiais etapais:

1. Kamuoliuko pozicijos sekimas. Žingsniai taikomi kiekvienam kadrui.
   1. Iš kadro iškerpamas 200x200 pikselių pločio langas, kurio centre yra paskutinė žinoma kamuoliuko pozicija, arba Kalmano filtro prognozuojama sekanti pozicija, jei kamuoliukas nėra randamas. Pirmą kadrą iškirpimas nėra vykdomas, vykdoma paieška visame kadre.
   2. Naudojamam langui pritaikoma nustatytos kamuoliuko HSV spalvų paletės kaukė.
   3. Lange pritaikoma erozijos operacija panaikinti likusiam triukšmui.
   4. Naudojamas OpenCV kontūrų paieškos algoritmas.
   5. Jei kamuoliukas rastas, einama prie sekančio žingsnio jei ne – atliekama paieška visame kadre, jei nerastas visame – sekančiam paieškos langui bus naudojama Kalmano filtro prognozė.
   6. Atnaujinamas Kalmano filtras su praeitame žingsnyje gauta kamuoliuko pozicija.
2. Žaidėjų pozicijos sekimas. 2.2 žingsnis vykdomas kiekvieną kadrą.
   1. Pirmame vaizdo srauto kadre aptinkamos žaidėjų lazdos.
      1. Kadrui taikomas „Canny“ edge detection algoritmas, kad gauti kadre esančias kraštines.
      2. Kraštinėms taikomas „Hough transformation“ algoritmas, atrenkatinis kadre esančias tiesias linijas. Iš linijų atrenkamos tik vertikalios arba horizontalios (priklausomai nuo kameros pozicijos) linijos.
      3. Iš likusių linijų, paieškomos 8 didžiausios kraštinių grupių viršūnės. Viršūnių padėtys – tai žaidėjų lazdų koordinatės.
   2. Kiekviename kadre aplink žaidėjų lazdas perskaičiuojamas skirtumas tarp pikselių RGB reikšmių ir ieškomų žaidėjų RGB spalvos. Pagal iš anksto sukonfiguruotą žaidėjų išsidėstyma, atrenkamos kelios aukščiausios reikšmės.

### Kalmano filtro veikimas

Vaizdo analizėje kamuoliuko sekimui patikslinti naudojamas Kalmano filtras. Kalmano filtras – Tai rekursinis filtras, kuris nuspėja tiesionio dinaminio proceso būseną – kamuoliuko poziciją ir greitį X bei Y ašyse – pagal matavimus, kurie gaunami su statistiniu triukšmu. Todėl judėjimui Kalmano filtro būsenos matrica atrodo:

=

# Projektas

## Reikalavimų specifikacija

### Komercinė specifikacija

Tai yra atvirojo kodo projektas. Jo užsakovė – įmonė MB „Tvirtas baitas“. Projektą vykdo įmonės darbuotojas, jo naudotojai – visi įmonės darbuotojai, žaidžiantys stalo futbolą bei kiti sistemos atvirąjį kodą naudojantys žmonės.

Projekto biudžetas nenumatytas.

Numatoma projekto išleidimo data – 2019 m. Gegužės 15 d. Iki šios datos turi būti paleista produkcinė saityno sistema, bei suprogramuota kameros posistemė.

### Sistemos funkcijos

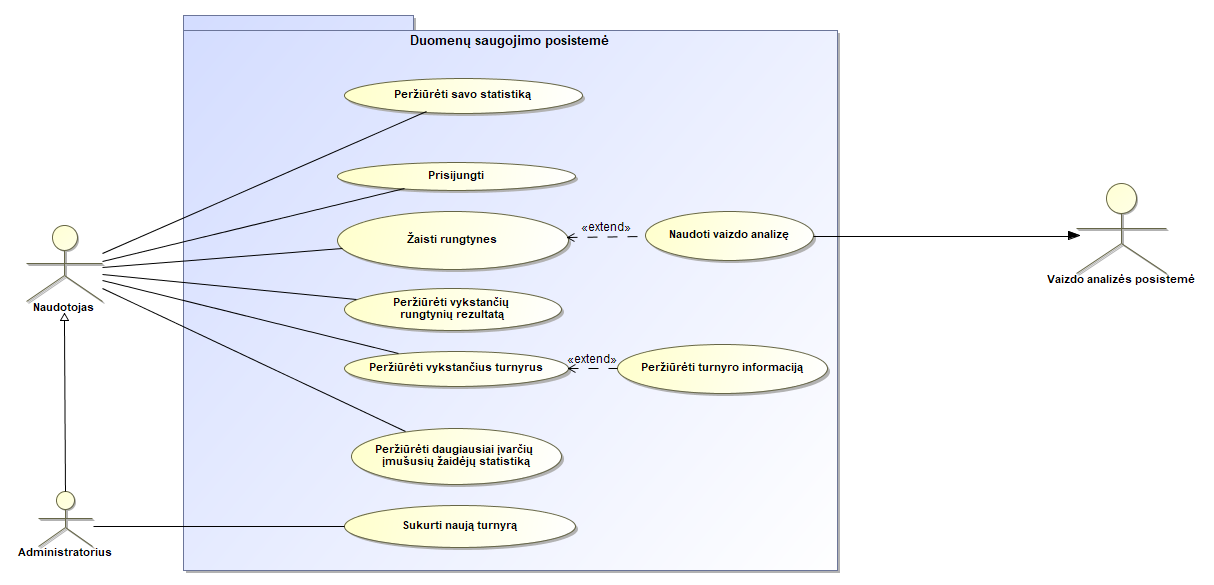
Sistemos naudojimo atvejai pateikti UML diagrama. Stalo futbolo sistemos paskirtis – patogus rezultatų saugojimas ir peržiūrėjimas. Tam pasiekti kiekvienas naudotojas turi savo paskyrą, prie kurios gali prisijungti naudodamasis savo įmonės paskyra, kurią galima pasiekti, autorizuojantis per Google UserInfo paslaugą.

Autentifikuotas naudotojas gali žaisti rungtynes. Taip pat jose jis gali pasirinkti – naudoti vaizdo analizę, ar ne.

Autentifikuoti naudotojai taip pat gali peržiūrėti vykstančių rungtynių rezultatą, peržiūrėti vykstančius turnyrus, kurių puslapyje taip pat gali pasirinkti peržiūrėti detalesnę informaciją apie turnyrą.

Apie autentifikuotus naudotojus yra kaupiama statistika, kurią jie gali peržiūrėti. Taip pat statistikos puslapyje jie gali pasirinkti peržiūrėti daugiausiai įvarčių pelniusius žaidėjus.

Sistemoje paskirti administratoriai turi įgaliojimą sukurti turnyrą.



**1 pav.** Sistemos panaudojimo atvejų diagrama

### Vartotojo sąsajos specifikacija

Vartotojo sąsajai nebuvo sukurta specifikacija, tačiau buvo iškelti keli reikalavimai:

1. Vartotojo sąsaja turi prisitaikyti prie vartotojo ekrano dydžio.
2. Vartotojo sąsajoje visuomet turi matytis grupės (šiuo atveju – įmonės), kurios žaidimų sistemoje yra esama, logotipas.
3. Vartotojo sąsaja turėtu mokėti naudotis ir ne programuotojai, be papildomų apmokymų.
4. Kurti turnyrus vartotojo sąsajoje administratorius turėtų galėti vieną kart perskaitęs naudotojo vadovą.
5. Vartotojo sąsajai turėtų būti naudojami Material Design nurodymai ir elementai.

### Realizacijai keliami reikalavimai

Realizacijai keliami nefunkciniai reikalavimai:

1. Realizacijai:
   1. Vaizdo apdorojimas turi vykti ne mažesniu nei 25 kadrų per sekunde greičiu.
   2. Kamuoliukas įprastose rungtynėse turėtų būti aptinkamas bent 90% kadrų, kuriuose jis aiškiai matosi.
   3. Duomenų saugojimo servisas neturėtų saugoti jokios būsenos.
   4. Vaizdo apdorojimo serveris turėtų asinchroniškai analizuoti vaizdo kadrus, bei siųsti ir gauti žinutes WebSocket kanalu.
2. Sistemos priežiūrai:
   1. Kodo pokyčių istorijai turi būti naudojama versijų kontrolės sistema GIT.
3. Saugumui:
   1. Žaidimų kūrimas prieinamas tik autentikuotiems vartotojams.
   2. Turnyrų kūrimas prieinamas tik autentikuotiems vartotojams su administratoriaus privilegijomis.
   3. Kliento – serverio bendravimo apsauga turi būti paremta JWT naudojimu.
   4. Prisijungimas prie vaizdo analizės serverio galimas tik žinant jame saugomą slaptažodį.

### Techninė specifikacija

Vaizdo analizės posistemė bus leidžiama lokaliame įrenginyje. Vazdas gaunamas iš Logitech C922 vaizdo kameros, tačiau tinka bet kuri kamera fiksuojanti 30 kadrų per sekundę 720p rezoliucija. Vaizdo analizei bus naudojamas kompiuteris su Intel i7-6700HQ procesoriumi.

Duomenų saugojimo posistemė bus laikoma Azure App Services duomenų servise, planuojama jog srautas nebus didelis, todėl pasirinktas pats pigiausias planas. Kartu su klientu buvo nuspręsta, jog šiai posistemei atnaujinimai bus reti ir mažos rizikos, todėl kuriama tik produkcinė aplinka.

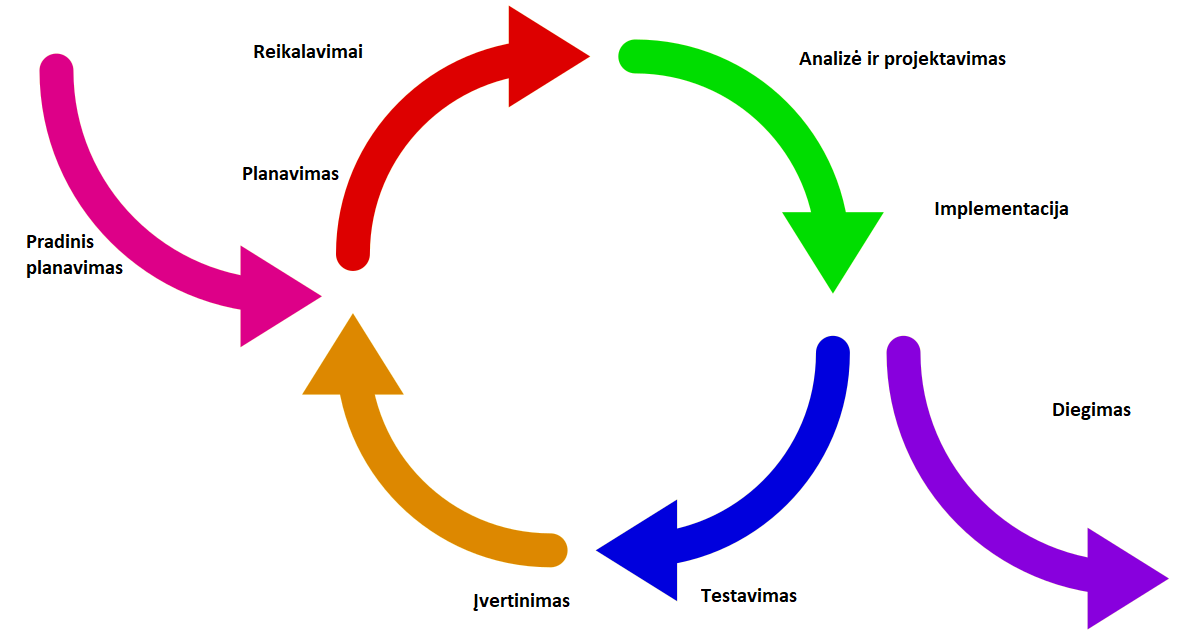
Detalesnė sistemos techninė specifikacija:

1. Vaizdo analizės posistemė.
   1. Procesorius Intel i7-6700HQ. 4 branduolių veikiančių 2.60 GHz dažniu.
   2. 32GB operatyviosios atminties veikiančios 2133MHz dažniu.
2. Duomenų saugojimo posistemė.
   1. *Azure App Services* instancija su automatiniu apkrovos balansavimu.
   2. *SQL Databases* duomenų bazė su 2 GB saugojimo vietos ir 5 DTU (Database Transaction Unit). DTU naudojami Azure išreikšti duomenų bazės CPU, operatyviosios atminties našumą. Pagal sqlperformance.com tyrimą, 5 DTU yra maždaug vienas CPU branduolys, naudojamas 5% našumu.

## Projektavimo metodai

### Projektavimo valdymas ir eiga

Projektavimas buvo vykdomas remiantis iteraciniu metodu. Taikant šį modelį buvo renkami reikalavimai posistemėms ar stambesniems komponentams. Pagal reikalavimus ruošiami projektai, kuriais remiantis posistemės ir komponentai buvo realizuoti. Buvo atliktos trys keturių savaičių iteracijos.



### Projektavimo technologija

Sistemos projektas buvo kuriamas naudojant UML 2.5 standarto grafinius elementus. Šalia diagramų pateiktas neformalus aprašas, paaiškinantis projektavimo idėjas. Diagramos kurtos naudojant MagicDraw 19.0 projektavimo įrankį.

### Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos

Sistemos duomenų saugojimo posistemė buvo kuriama C# kalba, .naudojant NET Core 2.2 karkasą, todėl ji gali būti diegiama tiek Windows, tiek Linux ar macOS operacinėse sistemose. Sistemos diegimui yra sukurtas Docker konteinerio atvaizdas. Posistemė buvo kuriama ir diegiama naudojant Visual Studio 2017 integruotą kūrimo aplinką su Reshaper 2017 įskiepiu. Posistemei testuoti naudojamas xUnit 2.4.0 testavimo karkasas.

Duomenų bazei pasirinktas Microsoft SQL Server 2017 palaikomas Windows operacinėse sistemose.

Naudotojo sąsaja buvo programuojama JavaScript kalba, naudojantis React.js biblioteka. Sąsajos projekto programavimui pagreitinti projektas pradėtas naudojantis create-react-app įrankiu. Lokaliam sąsajos testavimui buvo naudojamas Node.js serveris.

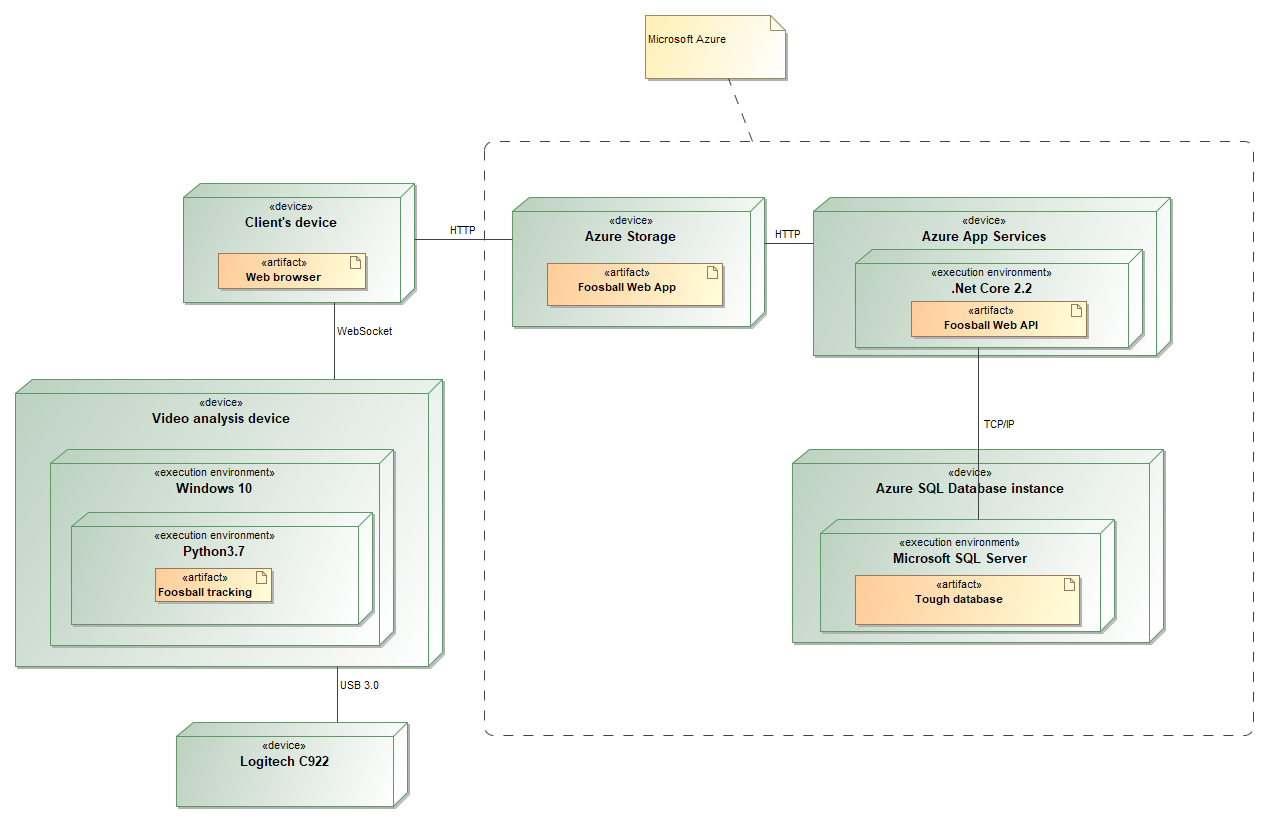
Vaizdo analizės sąsaja buvo programuojama su Python 3.7 programavimo kalba, naudojantis OpenCV 3.4.4 bibliotekos versija.

Naudotojo sąsaja bei vaizdo analizės posistemė buvo kuriama naudojant Visual Studio Code teksto redaktorių.

## Sistemos projektas

### Statinis sistemos vaizdas

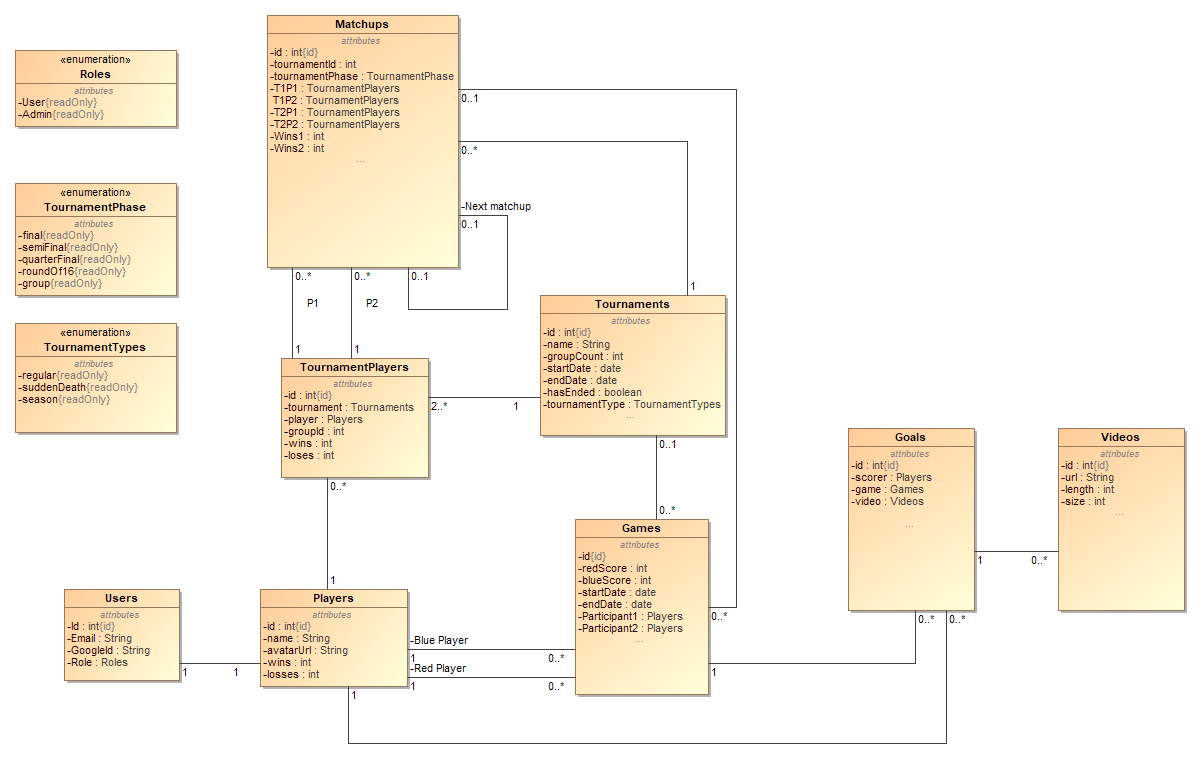
Sistema yra tik vidinio naudojimo, todėl buvo kuriama tik produkcinė aplinka. Jos išdėstymas pavaizduotas x paveikslėlyje.



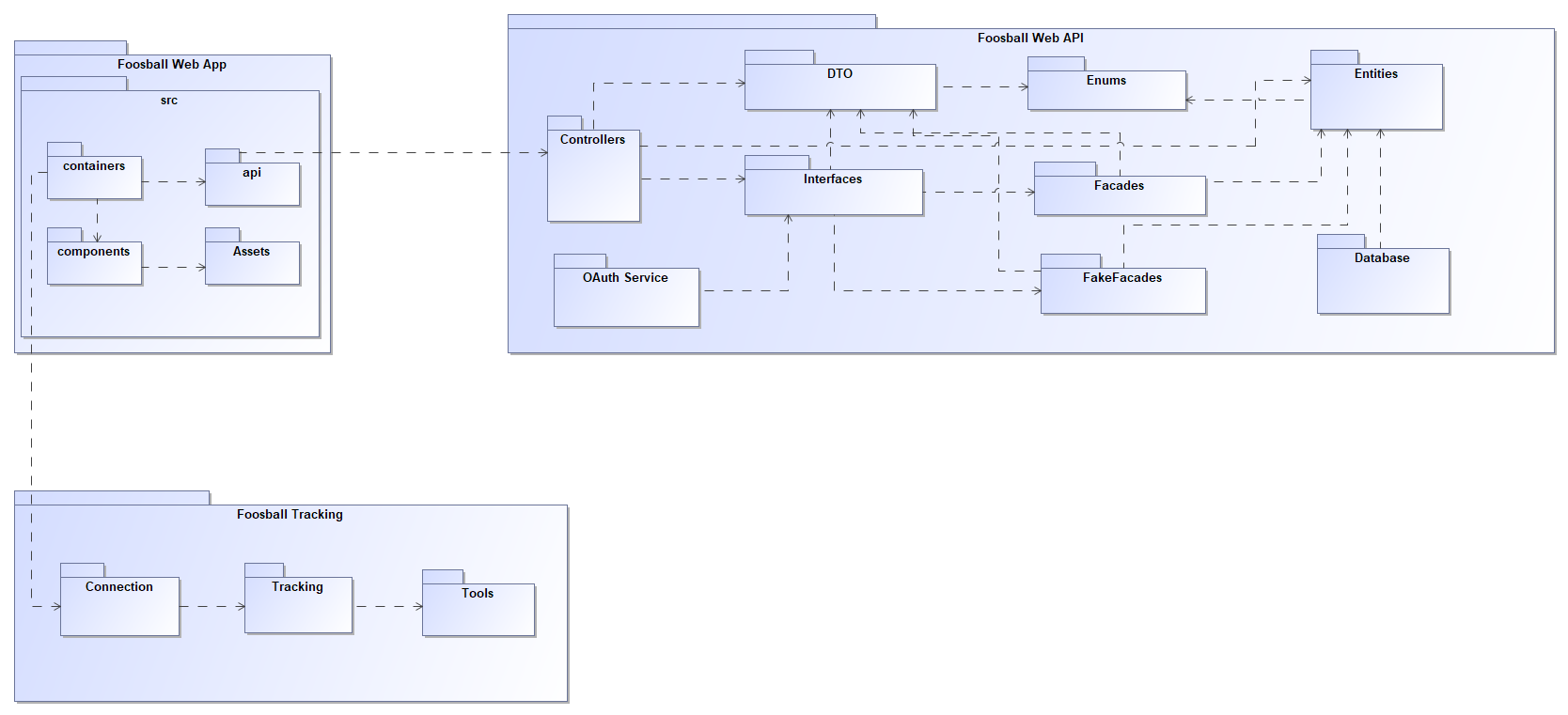
Saityno sistemai buvo pasirinktas pats papraščiausias aplikacijos variantas – naudojama Azure App Services paslauga. Ši Microsoft Azure debesų kompiuterijos paslauga leidžia sudiegti .Net Core projektą su minimalia konfigūracija. Statiniams failams – tarp jų ir naršyklės aplikacijai – naudojama Azure Storage paslauga. Duomenų bazė talpinama specialioje Azure SQL Database paskyroje.

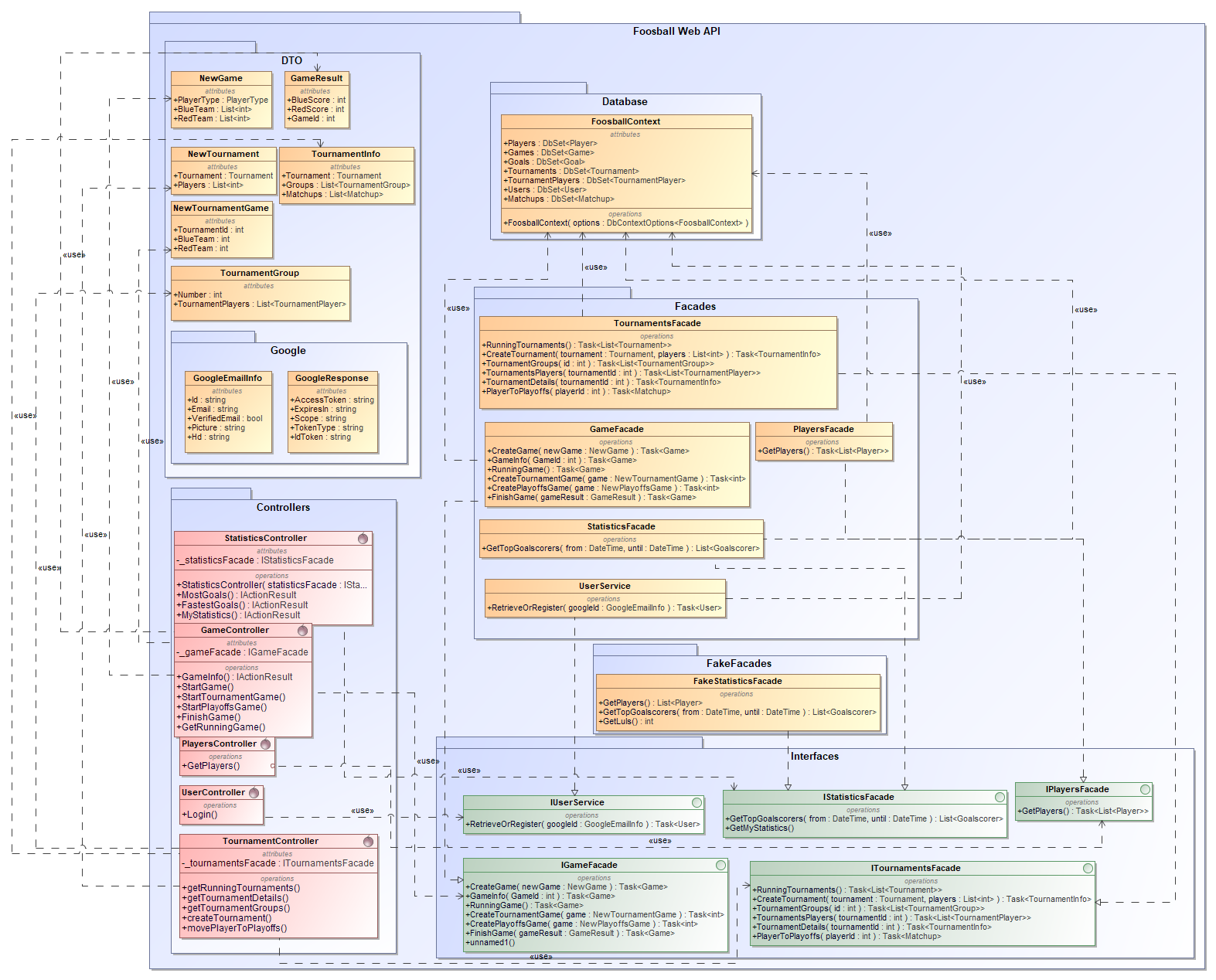
Vaizdo analizės modulis bus naudojamas asmeniniame kompiuteryje, Windows 10 operacinės sistemos aplinkoje, per Python 3.7 versijos interpretatorių. Bus naudojama Logitech C922 vaizdo kamera, sujungta su kompiuteriu per USB 3.0 jungtį.

Duomenų bazė buvo generuojama su EF Core įrankiu tiesiai iš C# kodo. Duomenų bazės schema vaizduojama paveikslėlyje.

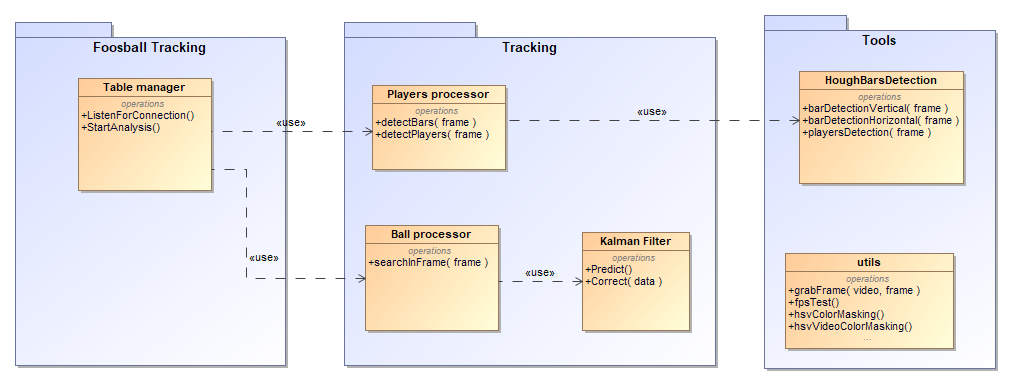


Projektą sudaro trys atskiri sprendimai, jų paketai nurodyti diagramoje. Diegimo metu Foosball Web App yra suspaudžiamas į kelis failus ir perkeliamas į Web API statinių failų saugyklą.





Vaizdo analizės sprendimą sudaro trys aplankai. Tools aplanke svarbiausi įrankiai sujungti į utils programą, kuri yra leidžiama savarankiškai, per Python interpretatoriu. Joje laikomi įrankiai, kurie padeda sistemos programuotojui testuoti sistemos spalvų filtravimą: išsisaugoti atskirus kadrus iš kameros, išbandyti vaizdo transliavimą iš prijungtos kameros su įvairiomis kadrų, rezoliucijos konfigūracijomis. Taip pat prie įrankių laikomi lazdų ir žaidėjų aptikimo algortimai, kuriuos testavimo tikslais galima pritaikyit nuotraukoms, filmuotiems įrašams ar transliacijoms.



// TODO: Tracking reikia pridėti dar vieną apjungiantį VideoTracking.py

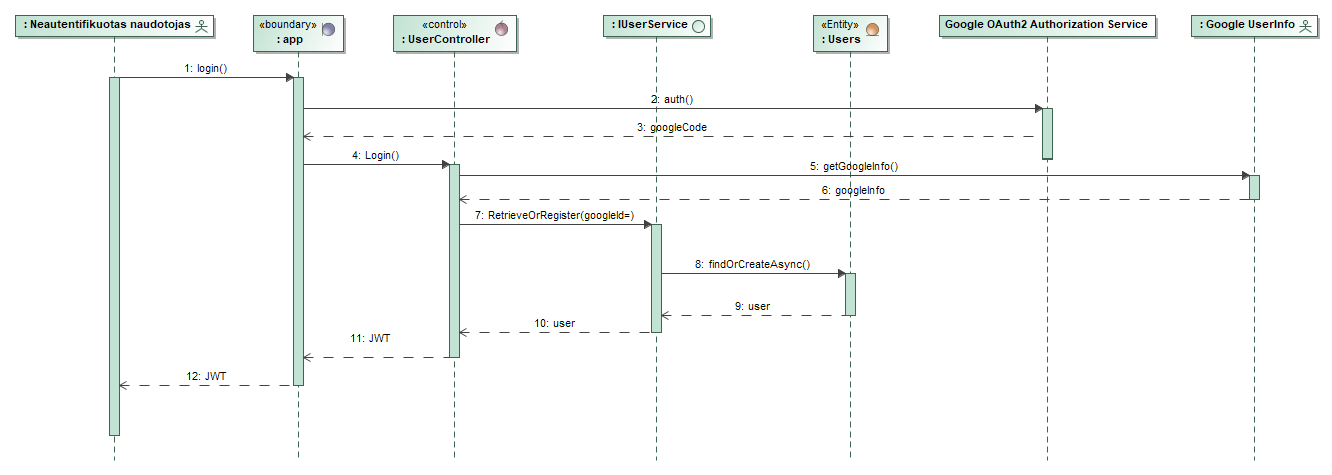
Tracking aplanke laikomos programos, skirtos pritaikyti bei apjungti aptikimo algoritmus.

VideoTracking programa prisijungia prie kameros transliacijos bei lygiagrečiai paleidžia analizės algoritmus.

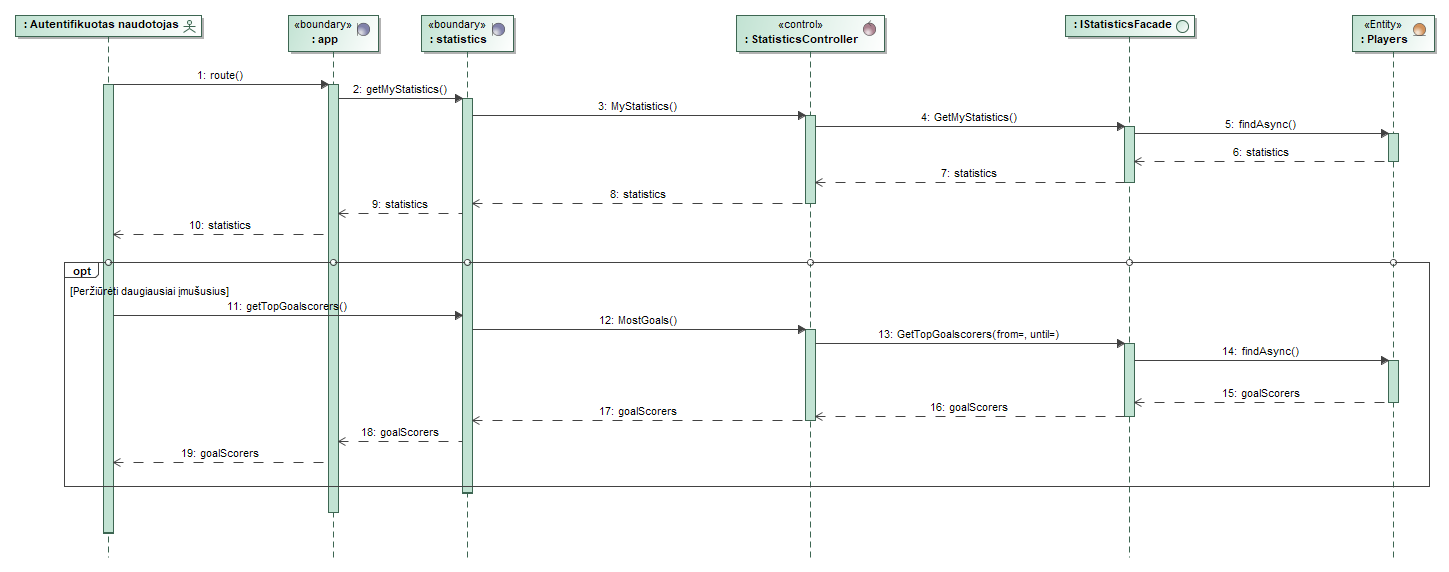
Foosball Tracking paketas skirtas išorinėms integracijoms. Jame esanti Table manager programa – tai vaizdo analizės serveris, kuris prisijungus naudotojui, paleidžia VideoTracking programą, ir siunčia naudotojui jos rezultatus.

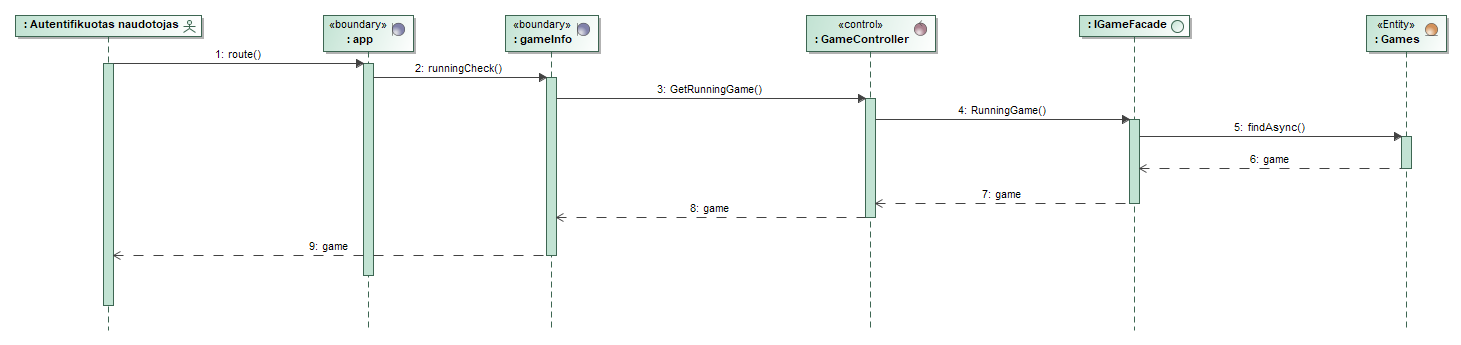
### Dinaminis sistemos vaizdas

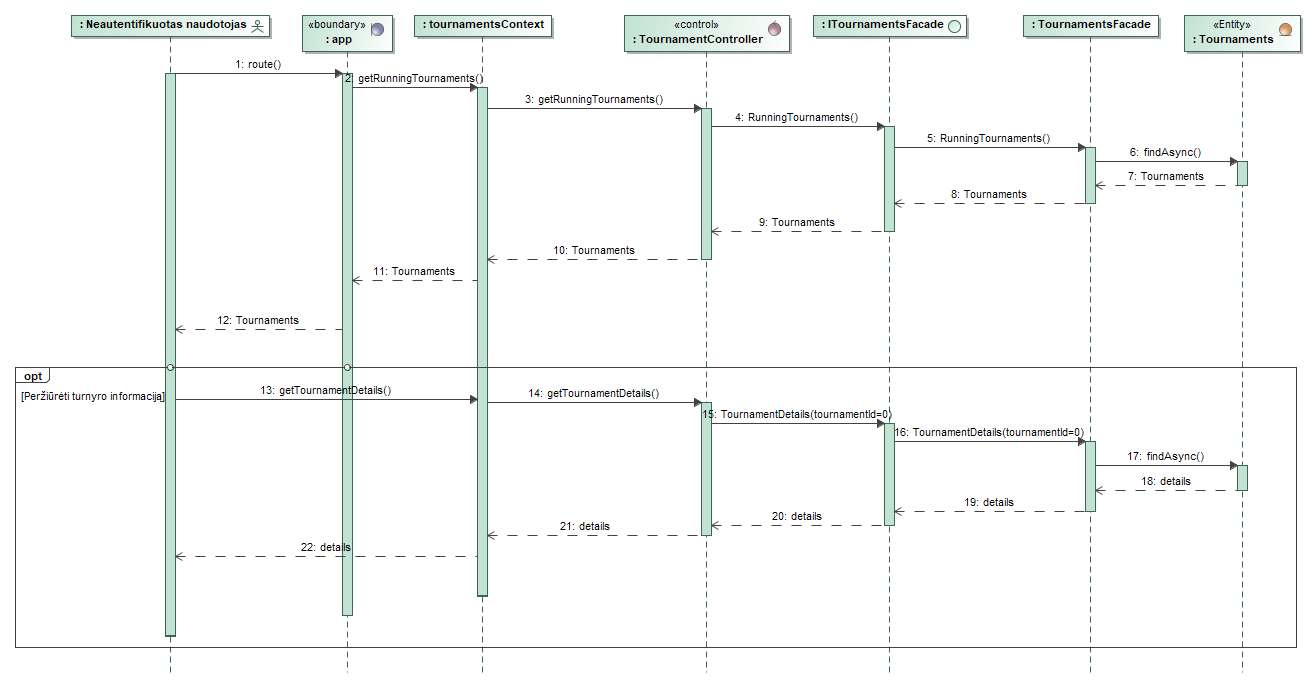
Sekos diagramoje nurodytas naudotojo prisijungimas. Per saityno sasąją pasirinkus prisijungimą, naudotojas nukreipiamas į išorinį Google OAuth servisą. Jame pasirinkus paskyrą, Google servisas nukreipia vartotoją į serverio valdiklio GoogeAuth metodą. Serveris atlieką užklausą į Google UserInfo servisą, kad verifikuoti naudotojo duomenis. Gavus atsakymą, patikrinama informacija duomenų bazėje ir naudotojui suteikiamas autentifikacijos žetonas.

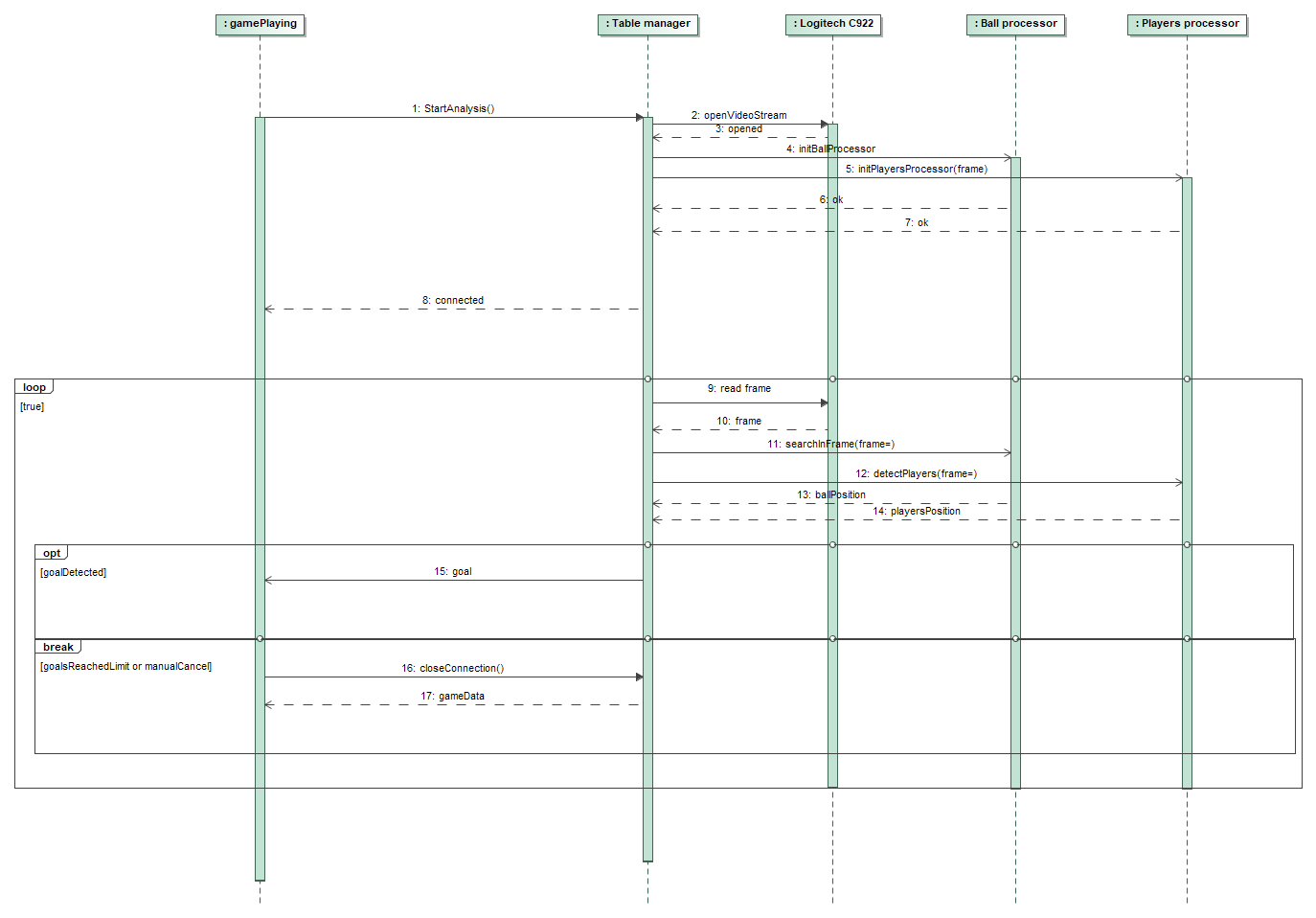


Paveikslėliuose [ ] nurodomos standartinės duomenų užklausos ir atsakymo operacijos, skirtos gauti vykstančioms rungtynėms, naudotojo statistikai.

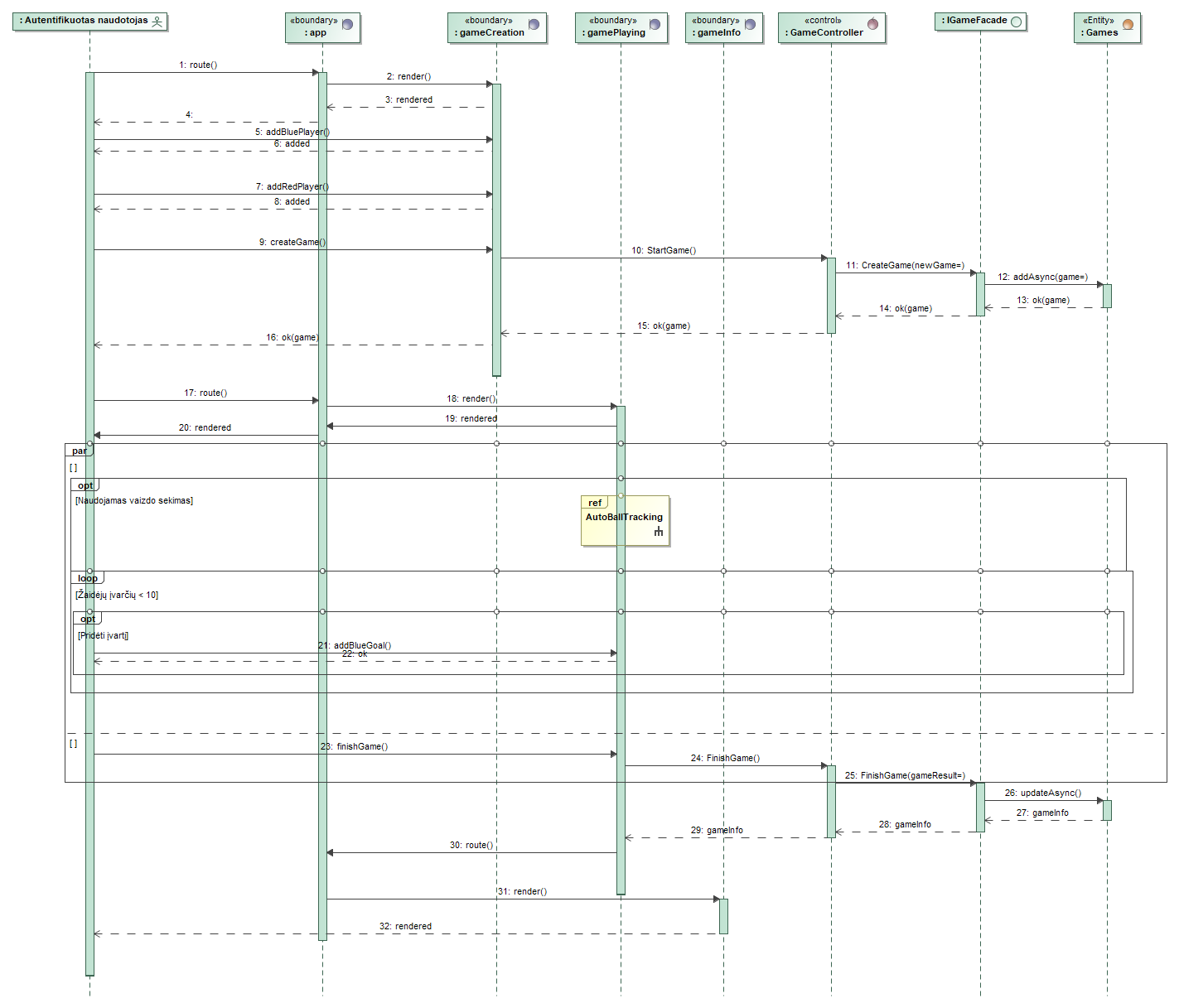






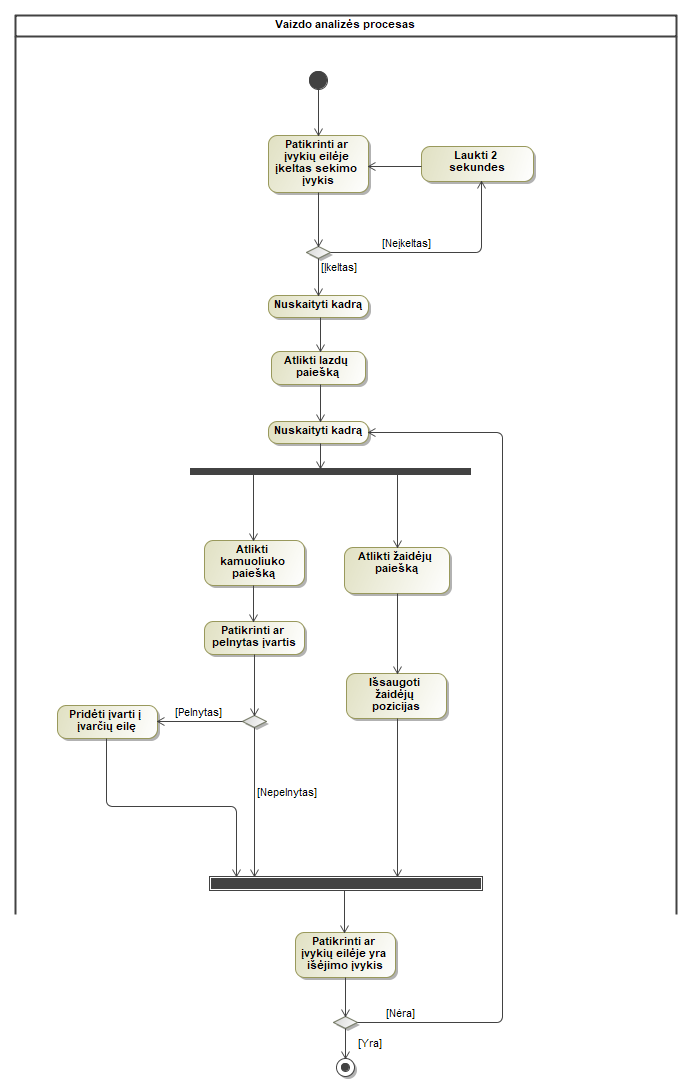


Rungtynių žaidimas prasideda žaidėjui atsidarius rungtynių puslapį. Jame naudotojas turi pasirinkti, kas iš sistemoje esančių žaidėjų žais už raudoną komandą, kas už mėlyną. Toliau pasirinkus pradėti rungtynes, išsiunčiama žinutė į serverio žaidimų valdiklį, kad būtų sukurtas įrašas apie pradėtas rungtynes. Gavus patvirtinimą iš serverio, naudotojas perkeliamas į vykstančių rungtynių langą. Lange galima pasirinkti, ar norima naudoti vaizdo analizę. Jei pasirenkama ją naudoti, naudotojo sąsaja bando prisijungti prie vaizdo analizės serverio. Pavykus prisijungti,vaizdo analizės serveryje sukuriami kamuoliuko ir žaidėjų sekimo procesai. Naudotojo sąsajai gavus patvirtinimą, jog prisijungta prie vaizdo analizės serverio, išsiunčiama žinutė pradėti sekimą. Vaizdo analizės serveris gaves šį pranešimą, nuskaito kadrą iš kameros. Tuomet iki kol yra gaunamas pranešimas, jog reikia sustabdyti vaizdo analizę, skaitomi kadrai iš vaizdo kameros ir perduodami kamuoliuko bei žaidėjų procesams. Jie grąžina objektų koordinates, kurios išsaugomos vaizdo analizės valdiklyje. Jei aptinkama, jog kamuoliukas pasiekė įvarčio linijos koordinates, siunčiamas įvarčio pranešimas bei paleidžiamas garsinis signalas. Gavus analizės sustabdymo žinutę iš kliento, arba nutraukus WebSocket kanalo prisijungimą, vaizdo analizės serveris perkeliamas į prisijungimo laukimo būseną. Naudotojo sąsajai nusiunčiama informacija apie kamuoliuko bei žaidėjų koordinates. Naudotojo sąsaja serveriui išsiunčia serveriui gautą informaciją iš vaizdo analizės serverio, gavus sėkmingą atsakymą, naudotojas perkeliamas į rungtynių apžvalgos puslapį.

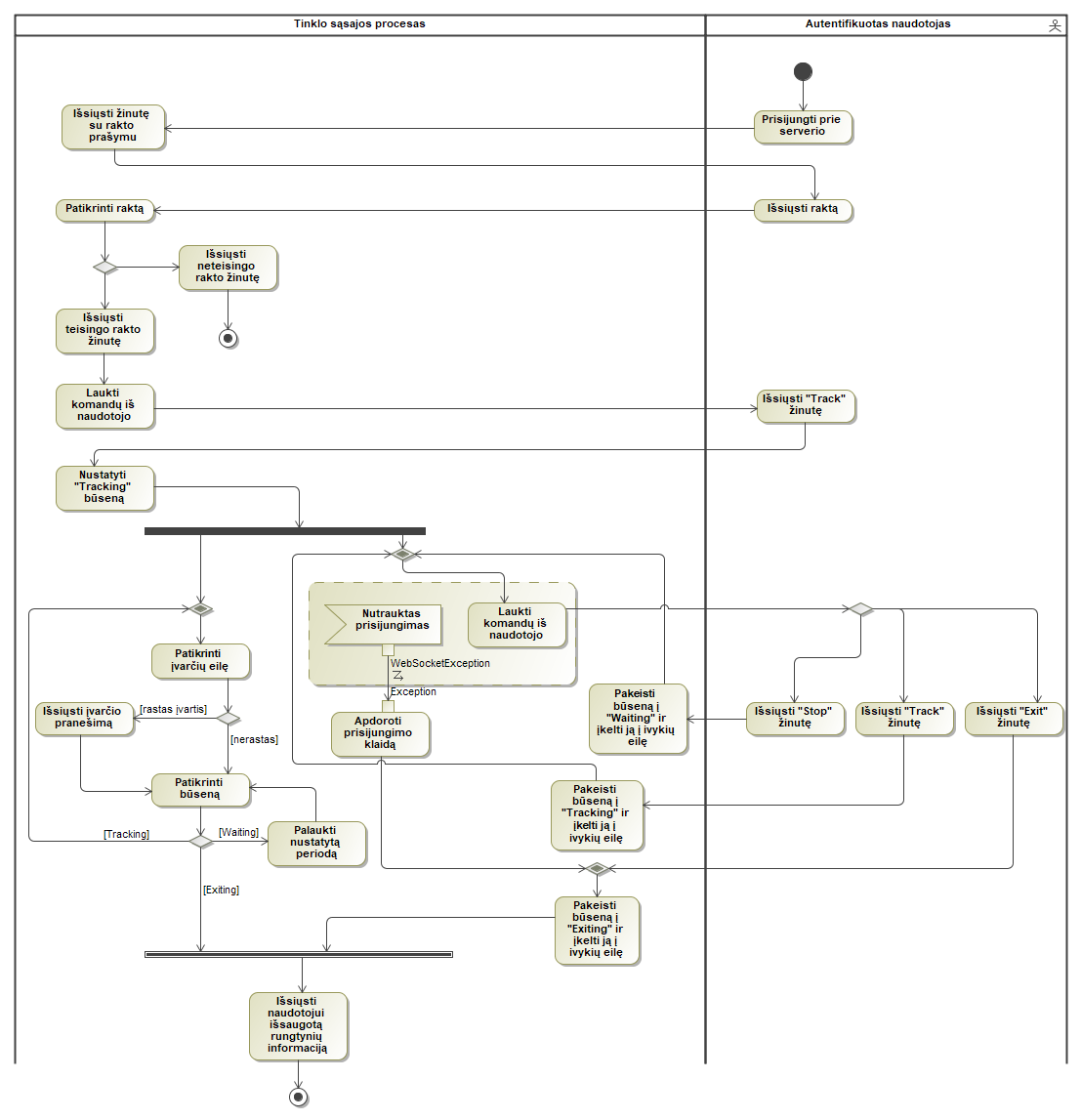


#### Vaizdo analizės serverio statinis vaizdas

Vaizdo analizės serveryje veikia du procesai, vienas iš jų skirtas tinklo komunikacijai su naudotoju, kitas vaizdo analizės operacijoms. Taip siekiama atskirti vaizdo analizės algoritmų architektūrą nuo bendravimo tinklu bei sumažinti greitaveikos nuostolius. Procesai tarpusavyje naudodami bendras eilių duomenų struktūras (angl. *Queues*), perduodami vieni kitiems įvykių žinutes. Atliekant vaizdo analizę taip pat sukuriami dar keli pagalbiniai procesai, naudotojui prisijungus į WebSocket kanalą ir pradėjus analizę, bendravimui sukuriamos dvi gijos, viena – siųsti pranešimus, kita klausytis jų iš naudotojo.



Vaizdo analizės procesas iš ivykių eilės gavęs pranešimą, jog reikia pradėti vykdyti sekimą, sukuria du vaikinius procesus, kurie lygiagrečiai vykdo ankščiau aprašytą algoritmą. Po kiekvieno kadro patikrinama įvykių eilėje, ar nėra išėjimo pranešimo, jei nėra, algoritmas kartojamas, jei yra, procesas ir jo vaikiniai procesai pasibaigia.



Naudojantis saityno sąsaja, prisijungimas prie saityno sąsajos yra supaprastintas ir valdomas saityno programos. Tačiau norint testuoti vaizdo analizės serverį, galima prisijungti iš bet kokios aplinkos, palaikančios WebSocket protokolą.

Tinklo procesas laukia, kol naudotojas prie jo prisijungs. Vienu metu gali būti prisijungęs tik vienas naudotojas. Prisijungus naudojui prašoma atsiųsti slaptažodį. Jei slaptažodis teisingas, naudotojas gali naudotis sistema, kitaip prisijungimas yra pabaigiamas. Norint pradėti naudojimą, tinklo procesas laukia komandos „Track“. Gavus šį įvykį, tinklo sąsaja sukuria dvi gijas, pranešimų priėmimui ir pranešimų siuntimui. Pranešimų gavimo sąsaja laukia iš naudojo pranešimo sustabdyti analizę. Jei toks pranešimas gaunamas, pakeičiama būsena ir įvykis idedamas į įvykių eilę analizės procesui. Naudotojui atsijungus iš tinklo, serveris veikia taip pat, kaip gavus išėjimo įvykį.

# Testavimas

## Testavimo planas

Suprojektavus sistemą buvo sudarytas toks testavimo planas. Vienetų testai serverio pusės metodams buvo rašomi prieš realizuojant sistemą. Statinė serverio pusės kodo analizė buvo vykdoma viso projekto realizacijos metu. Projekto testavimo planą sudaro šie žingsniai:

1. Vienetų testai serverio pusės metodams.
2. Statinė serverio pusės kodo analizė.
3. Rankinis serverio valdiklių kodo testavimas.
4. Rankinis vaizdo analizės testavimas.
5. Rankinis integracijų testavimas.
6. Automatinis vaizdo analizės testavimas su duomenų rinkiniais.

## Testavimo kriterijai

Sistemos testavimas laikomas sėkmingu, kai įvykdomi šie kriterijai

1. Visas sistemos kompiliuojamas kodas turi sėkmingai kompiliuotis.
2. Serverio pusės statinio kodo analizė neaptinka jokių klaidų ar įspėjimų.
3. Apsaugoti metodai neprieinami neautentifikuotiems ar neautorizuotiems naudotojams.
4. Visi vienetų testai turi būti sėkmingi.
5. API ir vartotojo sąsajos testavimo rezultatai turi atitikti sistemos reikalavimus.

## Komponentų testavimas

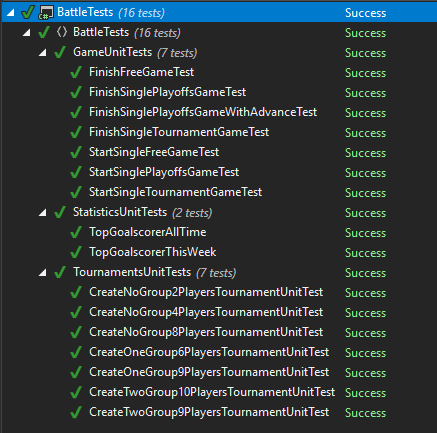
### Statinė kodo analizė.

Serverio pusėje statinė kodo analizė buvo atliekama naudojant ReSharper Ultimate 2018 statinės kodo analizės įrankį. Buvo naudojamas numatytosios statinės kodo analizės taisyklės, kurių yra 439. Statinė kodo analizė buvo vykdoma visu realizacijos metu, galutiniame projekte nei klaidų, nei įspėjimų neaptikta, todėl kodas kompiliuojasi ir statinė kodo analizė sėkminga.

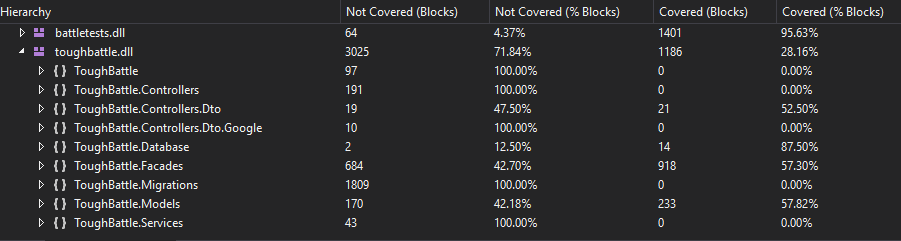
### Vienetų testai

Kadangi serverio turnyrų kūrimas užima didžiąją dalį kodo eilučių, todėl nuspręsta šiam funkcionalumui sukurti daugiausia vienetų testų. Vienetų testai buvo kuriami atsižvelgiant į visus turnyrų formatus bei įvairius žaidėjų, grupių skaičius. Taip pat sukurta po vieną ar du vienetų testus kitiems fasadų, į kuriuos kreipiasi valdikliai, metodams, nes jų veikimo logika nuo duomenų beveik nesiskiria.

Vienetų testai buvo kuriami naudojant xUnit 2.4.0 karkasą. Testų paleidimui, bei kodo padengimo skaičiavimui buvo naudojamas Resharper Ultimate 2018 įskiepis. Viso sukurti 16 vienetų testų. Visi testai buvo įvykdyti sėkmingai.



Serverio daliai buvo nuspręsta padengti tik svarbiausius scenarijus, todėl pasiektas nedidelis kodo padengimas. Visi vienetų testai yra sėkmingi, todėl šis kriterijus yra išpildytas.



## Rankinis testavimas

### Vaizdo analizės rankinis testavimas

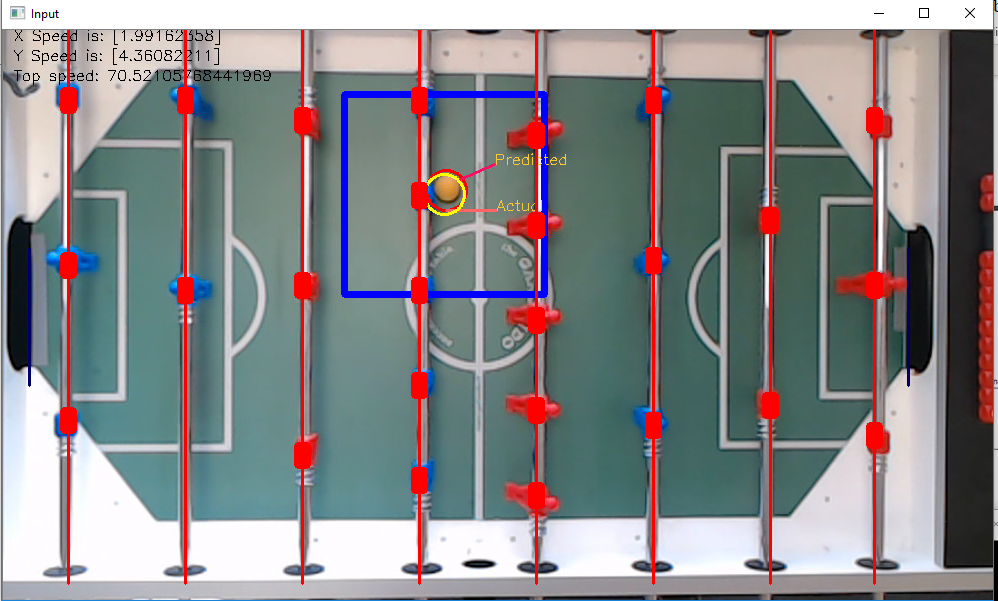
aiškus – kamuoliukas arba matosi ir turėtų būti aptinkamas, arba nesimato, ar matosi labai nedaug, ir turėtų būti neaptinkamas.

Testavimas buvo atliekamas viso kūrimo metu. Galutinėje sistemos versijoje buvo naudojami keli įrašai ir skaičuojama, kiek kadrų buvo teisingai aptikti žaidėjai bei kamuoliukas.

Žaidėjai visuose kadruose buvo aptinkami korektiškai – 100% sėkmingumas. Tad šis kriterijus išpildytas.

Kamuoliuko testavimo kriterijaus išpildymas buvo kiek mažiau apibrėžtas, nes įmanomi atvejai, kuomet įraše jis ir stebinčiam žmogui dėl persidengimo matomas mažiau nei 10% kadrų. Tačiau atsižvelgus į žaidėjų tendencijas, įprastose rungtynėse kamuoliukas turėtų matytis virš 80% kadrų.

Šis rezultatas nebuvo verifikuotas skaičiuojant kiekvieną kadrą, tačiau paėmus dviejų rungtynių įrašus iš duomenų saugyklos pasirenkant atsitiktiniu būdų, programa pateikė, jog kamuoliukas buvo aptikas 90% kadrų, tad sprendžiama, jog kriterijus – išpildytas.



### API rankinis testavimas

Serverio API buvo testuojamas naudojant Postman įrankį HTTP užklausų siuntimui. Užklausos buvo siunčiamos į lokalų API serverį, tačiau lokaliame serveryje buvo sugeneruota pakankamai duomenų, atvaizduojančių produkcinį serverį. Buvo testuojami visi API galiniai taškai, viso jų yra 13. Pateikiami keli daugiausiai naudojamų galinių taškų testavimo atvejai.

|  |  |
| --- | --- |
| API galinis taškas | GET /api/game/2 |
| Scenarijaus apibūdinimas | Bandoma gauti informacija apie sistemoje esantį žaidimą nr. 2, kuris egzistuoja |
| Prieigos žetonas | Nėra |
| Laukiamas rezultatas | Grąžinamas HTTP 200 rezultatas su Game klasės objektu, JSON formatu |

|  |  |
| --- | --- |
| API galinis taškas | GET /api/game/200 |
| Scenarijaus apibūdinimas | Bandoma gauti informacija apie sistemoje esantį žaidimą nr. 200, kuris neegzsistuoja |
| Prieigos žetonas | Nėra |
| Laukiamas rezultatas | Grąžinamas HTTP 404 rezultatas |

|  |  |
| --- | --- |
| API galinis taškas | POST /api/tournament |
| Scenarijaus apibūdinimas | Bandoma sukurti naują turnyrą |
| Prieigos žetonas | Nėra |
| Laukiamas rezultatas | Grąžinamas HTTP 401 Unauthorized rezultatas |

|  |  |
| --- | --- |
| API galinis taškas | POST /api/tournament |
| Scenarijaus apibūdinimas | Bandoma sukurti naują turnyrą |
| Prieigos žetonas | Paprastas naudotojas |
| Laukiamas rezultatas | Grąžinamas HTTP 403 Forbidden rezultatas |

|  |  |
| --- | --- |
| API galinis taškas | POST /api/tournament |
| Scenarijaus apibūdinimas | Bandoma sukurti naują turnyrą |
| Prieigos žetonas | Administratorius |
| Laukiamas rezultatas | Grąžinamas HTTP 200 OK rezultatas su Tournament klasės objektu, JSON formatu |

Ištestavus visus atvejus buvo gauti laukiami rezultatai, todėl API testavimo rezultatai kriterijus yra įvykdytas, apsaugoti metodai neautentifikuotiem ir neautorizuotiem naudotojam neprieinami.

## Vartotojo sąsajos testavimas

Buvo įvertinta, kad naudotojo sąsajai užtenka tik rankinio testavimo, todėl buvo vykdomi įprasti naudotojo veiksmai saityno programoje.

|  |  |
| --- | --- |
| **Pradinės sąlygos** | |
| Naudotojai kurie nori žaisti rungtynes yra užregistruoti sistemoję. Šiuo metu stalas nėra užimtas. Naudotojas atsidaręs „New game“ sistemos langą. | |
| **Veiksmas** | **Laukiamas rezultatas** |
| Paspaudžiama ant pirmojo žaidėjo nuotraukos. | Pirmas žaidėjas pridedamas į mėlynąją komandą. |
| Paspaudžiama ant antrojo žaidėjo nuotraukos. | Antras žaidėjas pridedamas į raudonąją komandą. |
| Paspaudžiama ant „Naudoti vaizdo analizę“ mygtuko. | Pateikiama, jog bus bandoma prisijungti ir naudoti vaizdo analizę. |
| Paspaudžiama „Start game“. | Atidaromas vykstančių rungtynių puslapis. |
| Paspaudžiamas „Add blue goal“ mygtukas. | Pridedamas įvartis mėlynajam žaidėjui. |
| Sąsajoje veiksmai nevykdomi, tačiau vaizdo analizė užfiksuoja mėlynojo žaidėjo įvartį. | Pridedamas įvartis mėlynajam žaidėjui. |
| Paspaudžiamas „Add blue goal“ mygtukas aštuonis kartus. | Mėlynajam žaidėjui pridedami aštuoni įvarčiai. Atidaromas rungtyniu apžvalgos langas. |

Ištestavus visus sąsajos testavimo scenarijus, visuose buvo gauti tikėtini rezultatai, todėl vartotojo sąsajos testavimo kriterijus buvo įvykdytas.

# Dokumentacija naudotojui

## Apibendrintas sistemos galimybių aprašymas

Realizuota sistema teikia realaus laiko stalo futbolo rungtynių vaizdo analizę. Analizei konfiguruoti yra sukurta keletas įrankių, kurių pagalba prižiūrintis asmuo, pasikeitus sekamų objektų spalvoms ar stalo pozicijai, perkonfiguruoti analizės programą. Analizės programa teikia realaus laiko informaciją apie pelnomus įvarčius bei kamuoliuko ir žaidėjų pozicijas. Šiai informacija gauti ir pateikti naudotojui skirta saityno programa, kuri prisijungia prie analizės programos WebSocket kanalu.

Sistemos duomenys saugomi duomenų bazėje, prie kurios prieigą ir duomenų pateikimą valdo serveris. Žmonėms patogiu formatu duomenys pateikiami per serverio API, kuriuo naudojasi saityno programa ir juos pateikia per grafinę sąsają.

## Vartotojo vadovas

### API specifikacija

|  |  |
| --- | --- |
| **HTTP Kodas** | **Paaiškinimas** |
| 200 | Užklausa sėkminga |
| 400 | Neteisingas užklausos turinys |
| 401 | Naudotojas neautentifikuotas |
| 403 | Naudotojas neautorizuotas prieiti šį turinį |
| 500 | Vidinė serverio klaida |

Pagal HTTP protokolo rekomendacijas, GET metoduose visi parametrai nurodomi per adresą, todėl sistemos GET metodai užklausų skilties neturi. Atsakymai serializuojami ir perduodami JSON formatu.

|  |  |  |  |
| --- | --- | --- | --- |
| Adresas | | GET /api/game/{id} | |
| Paskirtis | | Informacija apie rungtynes | |
| Priegos teisės | | Visi | |
| **Atsakymo modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| ID | Int | Taip | Rungtynių ID |
| RedTeamScore | Int | Taip | Raudonosios komandos įvarčiai |
| BlueTeamScore | Int | Taip | Mėlynosios komandos įvarčiai |
| BluePlayer | Player | Taip | Mėlynosios komandos žaidėjas |
| RedPlayer | Player | Taip | Raudonosios komandos žaidėjas |
| StartDate | DateTime | Taip | Pradžios laikas |
| EndDate | DateTime Nullable | Taip | Pabaigos laikas |
| TournamentID | Int | Ne | Turnyro sistemos ID |
| MatchupID | Int | Ne | Turnyro atkrintamųjų rungtynių sistemos ID |
| GameType | GameType | Taip | Rungtynių tipas |

|  |  |  |  |
| --- | --- | --- | --- |
| Adresas | | POST /api/game/ | |
| Paskirtis | | Naujų rungtynių sukurimas | |
| Priegos teisės | | Žaidėjai | |
| **Užklausos modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| BluePlayer | Int | Taip | Mėlynosios komandos žaidėjo ID |
| RedPlayer | Int | Taip | Raudonosios komandos žaidėjo ID |
| **Atsakymo modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| ID | Int | Taip | Rungtynių ID |
| StartDate | DateTime | Taip | Rungtynių pradžios laikas |

|  |  |  |  |
| --- | --- | --- | --- |
| Adresas | | POST /api/game/playoffsGame/ | |
| Paskirtis | | Naujų atkrintamųjų rungtynių sukurimas | |
| Priegos teisės | | Žaidėjai | |
| **Užklausos modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| BluePlayer | Int | Taip | Mėlynojo žaidėjo ID |
| RedPlayer | Int | Taip | Raudono žaidėjo ID |
| MatchupId | Int | Taip | Atkrintamųjų poros sistemos ID |
| **Atsakymo modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| ID | Int | Taip | Rungtynių ID |
| StartDate | DateTime | Taip |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Adresas | | POST /api/game/groupGame/ | |
| Paskirtis | | Naujų grupės rungtynių sukurimas | |
| Priegos teisės | | Žaidėjai | |
| **Užklausos modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| BluePlayer | Int | Taip | Mėlynojo žaidėjo ID |
| RedPlayer | Int | Taip | Raudonojo žaidėjo ID |
| TournamentId | Int | Taip | Turnyro sistemos ID |
| **Atsakymo modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| ID | Int | Taip | Rungtynių ID |
| StartDate | DateTime | Taip | Rungtynių pradžios laikas |

|  |  |  |  |
| --- | --- | --- | --- |
| Adresas | | POST /api/game/finish/ | |
| Paskirtis | | Rungtynių pabaiga | |
| Priegos teisės | | Žaidėjai | |
| **Užklausos modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| BlueScore | Int | Taip |  |
| RedScore | Int | Taip |  |
| GameId | Int |  |  |
| **Atsakymo modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| ID | Int | Taip | Rungtynių ID |
| RedTeamScore | Int | Taip | Raudonosios komandos įvarčiai |
| BlueTeamScore | Int | Taip | Mėlynosios komandos įvarčiai |
| BluePlayer | Player | Taip | Mėlynosios komandos žaidėjas |
| RedPlayer | Player | Taip | Raudonosios komandos žaidėjas |
| StartDate | DateTime | Taip | Pradžios laikas |
| EndDate | DateTime Nullable | Taip | Pabaigos laikas |
| TournamentID | Int | Ne | Turnyro sistemos ID |
| MatchupID | Int | Ne | Turnyro atkrintamųjų rungtynių sistemos ID |
| GameType | GameType | Taip | Rungtynių tipas |

|  |  |  |  |
| --- | --- | --- | --- |
| Adresas | | GET /api/players/ | |
| Paskirtis | | Visi sistemoje užregistruoti žaidėjai | |
| Priegos teisės | | Visi | |
| **Atsakymo modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| Players | List<Players> | Taip | Rungtynių ID |

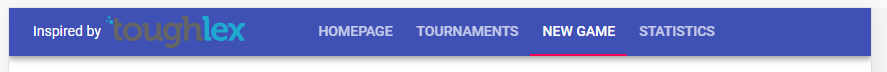
|  |  |  |  |
| --- | --- | --- | --- |
| Adresas | | GET /api/tournament/running | |
| Paskirtis | | Informacija apie dar vykstančius turnyrus | |
| Priegos teisės | | Visi | |
| **Atsakymo modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| Tournaments | List<Tournament> | Taip | Visi sistemos turnyrai Tournament klasės objektuose JSON formatu. |

|  |  |  |  |
| --- | --- | --- | --- |
| Adresas | | GET /api/ tournament /{id} | |
| Paskirtis | | Informacija apie turnyrą | |
| Priegos teisės | | Visi | |
| **Atsakymo modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| Tournament | Tournament | Taip | Turnyro objektas |
| Matchups | List<Matchups> | Ne | Turnyro atkrintamųjų porų objektų sąrašas. |
| Groups | List<Groups> | Ne | Turnyro grupių objektų sąrašas. |

|  |  |  |  |
| --- | --- | --- | --- |
| Adresas | | POST /api/ tournament / | |
| Paskirtis | | Sukurti naują turnyrą | |
| Priegos teisės | | Administratorius | |
| **Užklausos modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| Players | List<int> | Taip | Žaidėjų, dalyvausiančių turnyre, sistemos ID sąrašas |
| TournamentData | NewTournament | Taip | Turnyro konfigūracijos parametrų objektas |
| **Atsakymo modelis** | | | |
| **Pavadinimas** | **Tipas** | **Privalomas** | **Komentaras** |
| Tournament | Tournament | Taip | Turnyro objektas |
| Matchups | List<Matchups> | Ne | Turnyro atkrintamųjų porų objektų sąrašas. |
| Groups | List<Groups> | Ne | Turnyro grupių objektų sąrašas. |

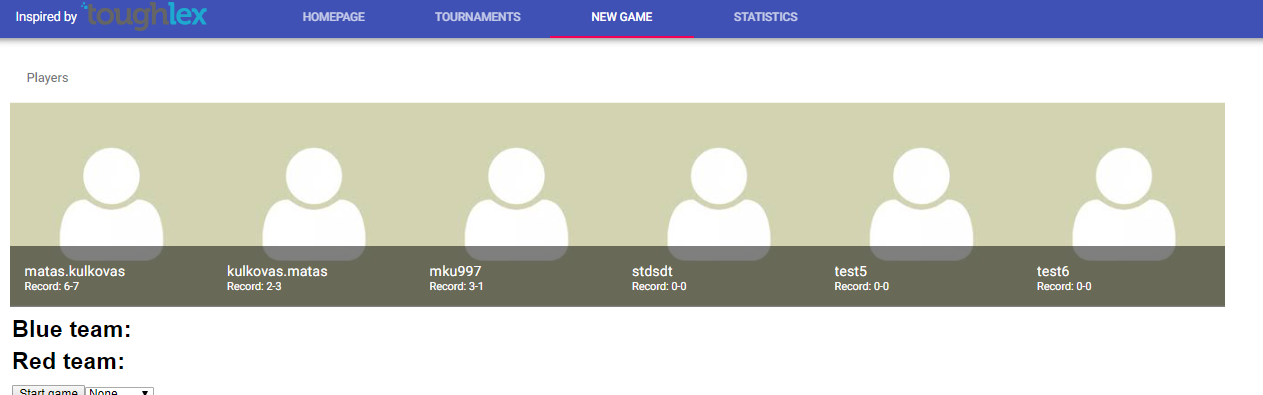
### Naudojimo eiga vartotojui

Puslapio funkcionalumas pateikiamas meniu juostoje.

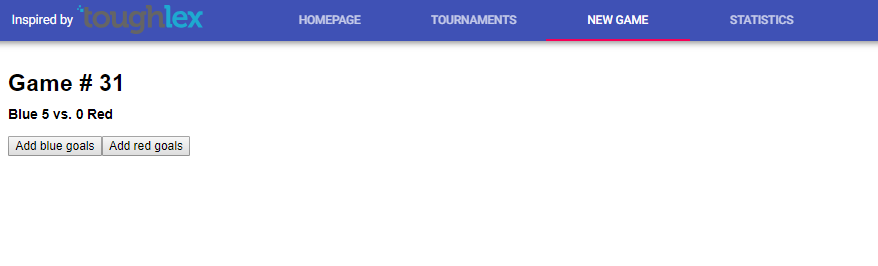


Norint peržiūrėti dabar vykstančias naujienas – pasirenkamas „Homepage“ puslapis, jame taip pat pateikiama informacija apie prisijungusį naudotoją.

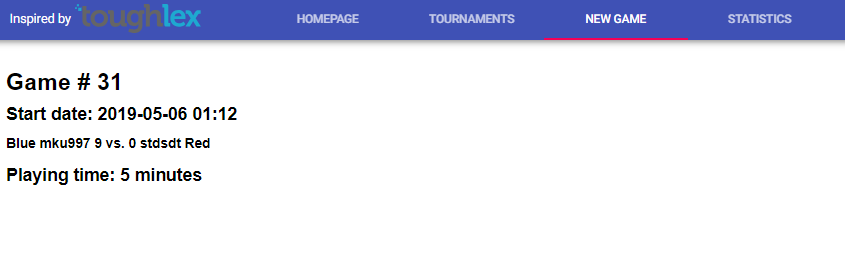
Norint pradėti rungtynes, pasinrekamas „New Game“ langas, kurį paspaudus, yra nukreipiama į žaidimo kūrimo puslapį. Jame pasirenkamas mėlynosios ir raudonosios komandos žaidėjas. Taip pat pasirenkama, ar tai yra turnyrinės rungtynės. Pasirinkus žaidėjus ir norint pradėti rungtynes, spaudžiamas „Start game“ mygtukas. Tuomet, kai gaunamas patvirtinimas iš serverio, jog galima pradėti rungtynes, naudotojas nukreipiamas į rungtynių langą.



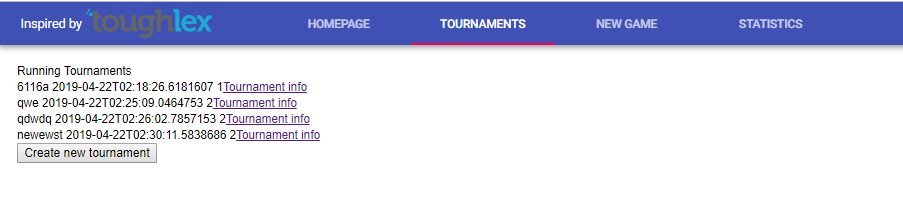
Rungtynių lange naudotojui rodomas dabartinis rungtynių rezultatas. Jei naudojama vaizdo analizė, rezultatas atsinaujina automatiškai įmušus įvartį, jei ne – mygtukais galima pridėti įvarčių. Kai kuris nors žaidėjas pasiekia 10 įvarčių, rezultatas išsiunčiamas į serverį ir naudotojas perkeliamas į rungtynių pabaigos langą.



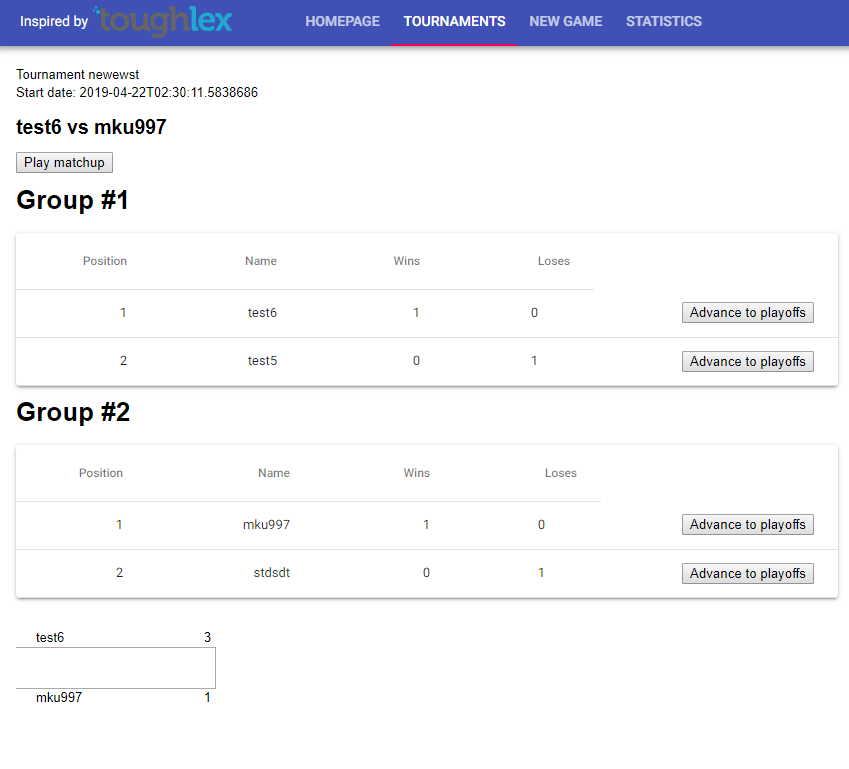
Rungtynių pabaigos lange galima peržvelgti rungtynių informaciją.



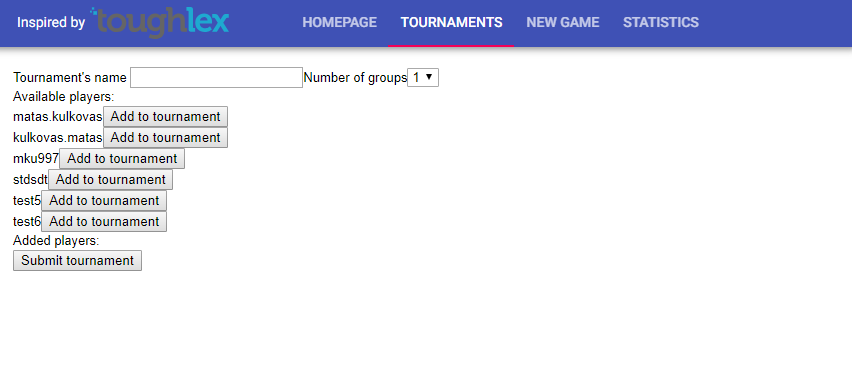
Norint susikurti paskyrą, užtenka prisijungti su savo įmonės suteiktu @toughlex.com e-paštu per Google vartus. Tai galima atlikti „Homepage“ puslapyje. Jei esate administratorius, galite sukurti turnyrą „Tournaments“ puslapyje. Jame taip pat visi naudotojai gali peržvelgti dabar vykstančius turnyrus ir jų struktūrą. Dabartinėje puslapio stadijoje taip pat nėbuvo implementuotas, tačiau ateityje bus reikalingas, automatinis turnyrų progresijos algoritmas, todėl administratorius žaidėją į atrinktamasias turėtų perkelti pasirinkdamas jį. Žaidėjai lentelėje rikiuojami pagal poziciją grupėje, todėl tai nesudaro sunkumų.



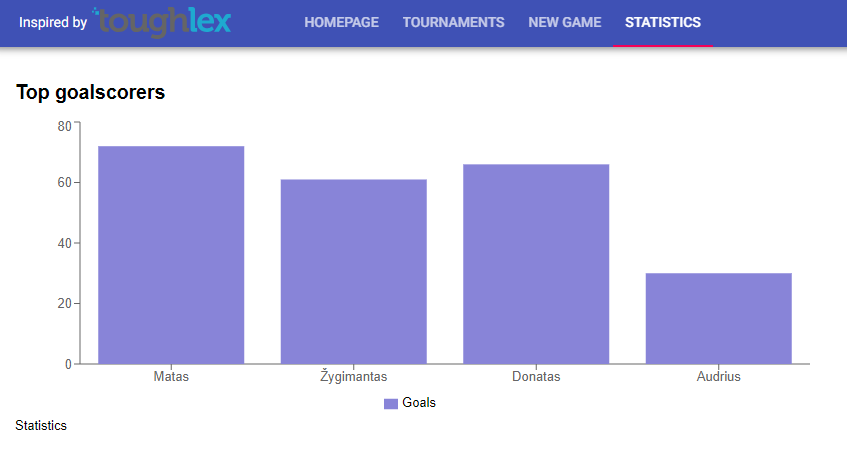
Turnyro informacijos puslapyje grupių etapas pateikiamas lentele, atkrintamųjų būsena – medžiu.



Turnyrų informacijos lange pasirinkus sukurti naują turnyrą, administratorius turėtų parinkti žaidėjus, kurie dalyvaus turnyre. Jis taip pat gali parinkti, ar turnyre bus grupių etapas, ir jei taip, kiek grupių turėtų būti.



Daugiausiai įvarčių pelniusių žaidėjų statistiką galima peržiūrėti „Statistics puslapyje. Jame informacija pateikiama histograma.



## Diegimo vadovas

Vaizdo analizės programa leidžiama per Python interpretatorių lokaliame įrenginyje, todėl atskiras diegimo scenarijus nebuvo kuriamas. Jei reikia – failai perkeliami per FTP protokolą ir programa paleidžiama iš naujo.

API serveris talpinamas Microsoft Azure App Services platformoje, todėl jo diegimams galimi du papraščiausi būdai – susiejus su talpykla arba naudojantis Visual Studio įrankiu. Diegimams iš Visual Studio įrankio, reikia atlikti šiuos žingsnius:

1. Azure portale prisijungiama prie savo paskyros, App Services platformoje sukuriama nauja Web App direktorija.
2. Visual Studio įrankyje prisijungiama prie Azure paskyros, pasirenkama naujai sukurta direktorija, nustatomi diegimai iš kodo.

Nuo šiol norint atnaujinti serveryje esančią programą per Visual Studio pakanka pasirinkti „Publish“ funkciją sudiegti kodui.

## Administravimo vadovas

Pagal techninę projekto specifikaciją, saityno aplikacija talpinama Microsoft Azure App Services platformoje. Azure portal puslapyje galima rasti visą reikalingą informaciją apie sistemos būseną, vykdomas užklausas ir klaidas.

Pagal specifikaciją, virš stalo turi būti pritvirtinta Logitech C922 kamera. Tačiau tinka bet kuri kita vaizdo kamera, kuri fiksuoja vaizdą bent 30 kadrų per sekundę 720p kokybe ir jungiasi USB 3.0 jungtimi.

Sistema fiksuoja iš anksto sukonfiguruotą geltoną kamuoliuką, bei raudonus ir mėlynus žaidėjus. Pasikeitus kurio nors iš objektų spalvai, Vaizdo analizės projekto Tools pakete yra įtraukti įrankis ColorMasker kuriuo naudojantis galima atrasti ir sukonfiguruoti tinkamus spalvų parametrus dabartiniam vaizdui. Paleidus BallTracking projekto įrankį Debug režimu yra vizualiai pateikiamos ribos kameros transliuojamame vaizde, kuriose turi būti pastatytas futbolo stalas.

Jei vaizdo analizės posistemė bandoma pasiekti ne iš lokalaus tinklo, posistemės turėtų atverti 42069 tinklo prievadą (angl. Port).

Rezultatai ir išvados

1. Konkurentų analizės metų nustatyta, jog daugumą atviro kodo stalo futbolo vaizdo sekimo sistemų norint integruoti su internetiniu servisu tektų atlikti nemažai pakeitimų. Todėl nuspręsta kurti sistemą kuri iškarto realiu laiku perduoda informaciją per WebSocket kanalą.
2. Vaizdo analizės sistemos projektavimo metu įsigilinta į OpenCV biblioteką, išnagrinėti bazinių operacijų su vaizdais metodai bei keli kontūrų atpažinimo algoritmai, taip pat išnagrinėtas Kalmano filtro veikimas. Tai leido suprojektuoti ir realizuoti vaizdo analizės posistemę su kuo mažiau operacijų.
3. Vaizdo analizės sistemos realizavimo metu pagilintos žinios lygiagrečiame programavime su Python kalba bei jos standartine biblioteka.
4. Duomenų saugojimo ir naudotojo sąsajos projektavimo ir realizavimo metu buvo praplėstos žinios su ASP.NET Core 2.2 karkasu bei React.js biblioteka. Buvo įsigilinta į sluoksniuotos architektūros principus, priklausomybės injekcijos, fasado projektavimo pavyzdžius.
5. Kuriant vienetų testus su xUnit 2.4 karkasu buvo išmokta kaip susikonfiguruoti testavimo projektą, kaip testus vykdyti su ReSharper įskiepiu.
6. Sukurta sistema yra naudoja įmonės stalo futbolo turnyrams. Šiuo metu yra tobulinama sistemos naudotojo sąsaja. Taip pat sistemą dažnai nepatogu naudoti dėl dedikuoto asmeninio kompiuterio reikalavimo.
7. Ruošiant sistemos dokumentaciją buvo daugiau kreipiama dėmesio į prieiga prie duomenų saugojimo API, nes numatoma, jog sistemos plėtimas bus vykdomas naudojant saugyklos pateiktus duomenis.

Literatūros sąrašas

1. 1. **Rekognition, AWS.** [Tinkle] Amazon Web Services. [Cituota: 2019 m. 05 19 d.] https://aws.amazon.com/rekognition/pricing/.
2. 2. **Apie LITNET. *Litnet.* [Tinkle] 2012 m. birželio 05 d. [Cituota: 2013 m. balandžio 04 d.] http://www.litnet.lt/index.php/apie-litnet.**
3. **3. *Transforming Ontology Representation from OWL to Relational Database.* Vyšniauskas, E. ir Nemuraitė, L. 3, 2006 m., Information Technology and Control, T. 35A, p. 333–343.**
4. **4. Masiulis, K. ir Krupavičius, A. *Valstybės tarnyba Lietuvoje: praeitis ir dabartis: kolektyvinė monografija.* Vilnius : Praction, 2007. p. 430.**
5. **5. *Spaudos draudimo klausimai.* Biržiška, V. 5, 1929 m., Kultūra, p. 249-235.**
6. **6. Valiulytė, Ieva. Išlaidos krašto apsaugai, jų pagrįstumas ir tikslingumas. *Sociumas.* [Tinkle] 2000 m. vasaris. [Cituota: 2001 m. gruodžio 12 d.] http://www.sociumas.lt.**
7. **7. Library, Dalhousie University. IEEE Citation style guide. [Tinkle] 2009 m. [Cituota: 2013 m. 04 11 d.] http://libraries.dal.ca/content/dam/dalhousie/pdf/library/Style\_Guides/IEEE\_Citation\_Style\_Guide.pdf.**
8. **8. *Hibridinis velomobilis.* Gradauskas, R. Kaunas : s.n., 2000. Transporto priemonės - 99. p. 81-83.**

<https://aws.amazon.com/rekognition/pricing/>

<https://kicktrack.app/#features>

Darbe naudotos literatūros sąrašas (1 – 3 lapai). Sąrašas sudaromas vadovaujantis ISO 690 priimtu literatūros sąrašo ir citavimo stiliumi (1). Kaip sudarinėti literatūros sąrašą Word priemonėmis galite paskaityti <http://office.microsoft.com/en-us/word-help/create-a-bibliography-HA102809686.aspx> arba <http://office.microsoft.com/lt-lt/word-help/create-a-bibliography-HA102809686.aspx>.

Literatūros sąrašas turėtų apimti visus naudotus šaltinius. Literatūros šaltiniai pateikiami sunumeruoti citavimo tvarka. Darbo apraše turi būti pacituoti visi naudoti šaltiniai, pateikiant tekste nuorodas. Daugiau informacijos apie bendras citavimo taisykles galite rasti <https://biblioteka.ktu.edu/mokymai/#mokymosi-medziaga> „Kaip cituoti šaltinius ir parengti literatūros sąrašą. ISO 690:2010 standartas (skirta technologijos mokslams)“.

Priedai

Papildoma informacija ir dokumentai, neįeinanti į pagrindinį dokumentą. Dažniausiai į priedus keliamos specifikacijos, sąsajų (API) aprašai, diagramos ar kita informacija, kuri yra svarbi projektui, tačiau užima dokumente ganėtinai daug vietos. Jei priedai nėra naudojami, šis puslapis yra ištrinamas.

1. priedas. Priedo pavadinimas

**2 lentelė.** Pagrindiniai baigiamojo projekto stiliai ir jų aprašymai

| **Stiliaus pavadinimas** | Stiliaus pavadinimas galerijoje | Stiliaus formalieji reikalavimai | Stiliaus naudojimo aprašymas |
| --- | --- | --- | --- |
| Antraštė non-TOC | Antraštė non-TOC | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po antraštės – 10 pt, centruota lygiuotė. | Antraštėms, kurios nėra įtraukiamos į turinį: „Santrauka“, „Summary“, „Turinys“. |
| Antraštė be nr. | Antraštė be nr. | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po antraštės – 10 pt, centruota lygiuotė, antraštė rašoma naujame puslapyje – po puslapio skirtuko. | Antraštėms, kurios įtraukiamos į turinį, bet nėra numeruojamos: „Lentelių sąrašas“, „Paveikslų sąrašas“, „Santrumpų ir terminų sąrašas“, „Įvadas“, „Išvados“, „Literatūros sąrašas“, „Informacijos šaltinių sąrašas“, „Priedai“. |
| 1. Heading 1,Skyrius | Skyrius | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas po antraštės – 10 pt, abipusė lygiuotė, antraštė rašoma naujame puslapyje – po puslapio skirtuko. | Skyrių antraštėms, kurios įtraukiamos į turinį ir yra numeruojamos. |
| 1.1. Heading 2,Poskyris | Poskyris | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po antraštės – 10 pt, abipusė lygiuotė, numeracija siejama su aukštesnio lygio antrašte. | Poskyrių antraštėms, kurios įtraukiamos į turinį ir yra numeruojamos. |
| 1.1.1. Heading 4,Skyrelis | Skyrelis | Skyrelių antraštėms, kurios įtraukiamos į turinį ir yra numeruojamos. |
| Tekstas | Tekstas | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, atstumas po pastraipos – 10 pt, abipusė lygiuotė. | Tekstui visose projekto dalyse (santraukose, įvade, skyriuose, poskyriuose ir t.t.). |
| List Bullet;Sąrašas (suženklintas) | Sąrašas (suženklintas) | Pirmos pastraipos eilutės įtrauka – 0,63 cm, šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, atstumas tarp tokio paties stiliaus pastraipų – 0 pt, atstumas po sąrašo – 10 pt, abipusė lygiuotė. | Tekstui, kuris pateikiamas suženklintu sąrašu. |
| List Number;Sąrašas (numeruotas) | Sąrašas (numeruotas) | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, atstumas tarp tokio paties stiliaus pastraipų – 0 pt, atstumas po sąrašo – 10 pt, abipusė lygiuotė. | Tekstui, kuris pateikiamas sunumeruotu sąrašu. |
| Footnote Text;Išnašos tekstas | Išnašos tekstas | Šrifto dydis 10 pt, intervalas tarp eilučių – 1,15, atstumas prieš ir po sąrašo – 0 pt, abipusė lygiuotė. | Tekstui, kuris pateikiamas išnašose. |
| Lentelės pavad. | Lentelės pavad. | Šrifto dydis 11 pt, intervalas tarp eilučių – 1,15, atstumas prieš pavadinimą – 10 pt, po pavadinimo – 3 pt, lygiuotė prie kairiojo krašto. | Lentelių pavadinimams (numeris ir žodis lentelė rašomas paryškintu šriftu). |
| Lentelės I eil. | Lentelės I eil. | Šrifto dydis 10 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po pastraipos – 3 pt, lygiuotė prie kairiojo krašto. | Tekstui lentelės antraštinei (pirmai) eilutei. |
| Lentelė | Lentelė | Šrifto dydis 10 pt, intervalas tarp eilučių – 1, atstumas prieš ir po pastraipos – 3 pt, lygiuotė prie kairiojo krašto. | Tekstui lentelėje. |
| Caption,Paveikslo pavad. | Paveikslo pavad. | Šrifto dydis 11 pt, intervalas tarp eilučių – 1,15, atstumas prieš ir po pavadinimo – 10 pt, centruota lygiuotė. | Paveikslų pavadinimams (numeris ir santrumpa pav. rašoma paryškintu šriftu). |
| Figure;Paveikslas | Paveikslas | Atstumas prieš ir po paveikslo – 10 pt, centruota lygiuotė. | Paveikslui, iliustracijai . |
| Bibliography,Bibliografija | Bibliografija | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, atstumas tarp tokio paties stiliaus pastraipų – 2 pt, abipusė lygiuotė. | Literatūros ir Informacijos šaltinių sąrašuose nurodytiems šaltiniams. |
| Priedas | Priedas | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, atstumas prieš ir po antraštės – 10 pt, lygiuotė prie kairiojo krašto. | Priedo numeriui, žodžiui *priedas*, priedo pavadinimui. |
| TOC 1,Turinys 1 | Turinys 1 | Šrifto dydis 12 pt, šriftas paryškintas, intervalas tarp eilučių – 1,15, įtrauka – 0,64. | Turinyje esančioms antraštėms, kurios nėra numeruojamos („Lentelių sąrašas“, „Paveikslų sąrašas“, „Santrumpų ir terminų sąrašas“, „Įvadas“, „Išvados“, „Literatūros sąrašas“, „Informacijos šaltinių sąrašas“, „Priedai“) ir numeruojamai antraštei „Skyriaus pavadinimas“. |
| TOC 2,Turinys 2 | Turinys 2 | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, įtrauka – 0,96. | Turinyje esančiai antraštei „Poskyrio pavadinimas“. |
| TOC 3,Turinys 3 | Turinys 3 | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, įtrauka – 1,28. | Turinyje esančiai antraštei „Skyrelio pavadinimas“. |
| TOC 4,Turinys 4 | Turinys 4 | Šrifto dydis 12 pt, intervalas tarp eilučių – 1,15, įtrauka – 0,64. | Numeruotiems priedams. |