

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond



Oliver Ilmjärv 180458IADB

**JUDO VÕISTLUSTE
AUTOMATISEERIMISE JA
REGISTREERIMISE PLATVORM**

Bakalaureusetöö

Juhendaja: Nadežda Furs-
Nižnikova
MBA

Tallinn 2020

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Oliver Ilmjärv

30.04.2020

Annotatsioon

Käesolevas töös luuakse veebirakendus, mis lihtsustab ning kiirendab judo võistluste korraldamisega ning nendele registreerimisega seotud probleeme. Hetkel toimiv meetodika on aeganõudev, vigu pärssiv ning keeruline. Puudub avalik üldine ülevaade eelnevatest võistlustest ning keskne tulemuste kogu.

Töö eesmärgiks on muuta judo võistluste loomine kiiremaks ja lihtsamaks, automatiseerides tegevused mis on seotud võistluste loomisega, alates juhendi loomisest kuni võistluse toimumise hetkeni. Loodav süsteem toetab võistluste loomist peamiselt „export” funktsionaalsusega. Lisaväärtusena tekib võistlustele reklaam, võistlustulemuste reklaamimine ning ühine ürituste kogu. Veebirakendus on peamiselt mõeldud spordiklubide treeneritele, kellel on võimalus lisada sportlasi enda loodud klubi alla ning taaskasutada sisestatud võistlejaid kõikidel soovitud võistlustel.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 8 peatükki, 7 joonist.

Abstract

Platform for Judo Competition Creation and Registration Process

The aim of the thesis is to develop a web application for judo competition creation and registration process simplification and automation. The current system is a slow and error prone. A collection of past competitions and results is not easily accessible for those who need it.

The main goal of the resulting product is to simplify, automate and make faster the current method for judo competition creation and registration process by providing a simpler method for coaches to create and advertise their event. The data for all the athletes will be stored and associated with their coaches and can be reused for all competitions. Registration process is made simpler by calculating an age category for all individual athletes and the remaining action a coach has to perform is to choose the weight category. The data is being collected for a specific competition and is made exportable in a CSV (comma separated value) format to use competitors data in another program.

The thesis is in Estonian and contains 25 pages of text, 8 chapters, 7 figures.

Lühendite ja mõistete sõnastik

| | |
|------|---|
| CDN | <i>Content delivery network</i> – Javascripti teekide jagamiseks mõeldud võrgustik. |
| CSRF | <i>Cross site request forgery (token)</i> – Märgistus takistamaks ründajate poolt sooritatud päringuid teise kasutaja arvelt. |
| CSS | <i>Cascading style sheet</i> – veebilehe stiilimiseks mõeldud keel. |
| CSV | <i>Comma separated value</i> – komadega eraldatud väärtuste failiformaat. |
| DTL | <i>Django template language</i> – Django mallide loomise keel. |
| ERD | <i>Entity relational mapping</i> – Andmebaasimudelivaheliste suhete visuaalne representatsioon. |
| GIT | <i>Version control</i> – Koodi versioneerimise vahend. |
| HTML | <i>Hypertext markup language</i> – Keel disainitud dokumentide veebilehitsejas kuvamiseks. |
| JSON | <i>Javascript object notation</i> – Andmeformaad. |
| ORM | <i>Object Relational mapping</i> – Metoodika andmebaasi manipuleerimiseks läbi objektide. |
| PDF | <i>Portable data format</i> – Dokumentide failiformaat. |
| PIP | <i>Python package manager</i> – Pythoni teekide ligipääsu vahend. |
| REST | <i>Representational State Transfer</i> – Teenustevaheline suhtluste standard. |
| SQL | <i>Structured Query Language</i> – Andmebaasi suhtlus keel. |
| URL | <i>Unified resource locator</i> – Veebiaadress. |
| XSS | <i>Cross site scripting</i> – Veebiteenuste turvaauk. |

Sisukord

| | | |
|-------|---------------------------------------|----|
| 1 | Sissejuhatus..... | 9 |
| 2 | Autori roll..... | 10 |
| 3 | Projekti skoop..... | 11 |
| 4 | Analoogsed lahendused..... | 12 |
| 5 | Analüüs..... | 13 |
| 5.1 | Andmebaas..... | 13 |
| 5.1.1 | Mudelid..... | 13 |
| 5.1.2 | Päringud..... | 15 |
| 5.2 | <i>Kasutajaliidese analüüs</i> | 15 |
| 5.2.1 | Bootstrap..... | 15 |
| 5.2.2 | Javascript..... | 16 |
| 5.2.3 | Ajax..... | 16 |
| 5.2.4 | Django Template Language..... | 16 |
| 5.3 | Tagarakenduse analüüs..... | 16 |
| 5.3.1 | Django veebiraamistik..... | 16 |
| 5.3.2 | Django Countries teek..... | 17 |
| 5.3.3 | Django Crispy forms teek..... | 17 |
| 5.3.4 | Django i18n teek..... | 17 |
| 5.3.5 | Pythoni Environ teek..... | 18 |
| 5.3.6 | Django administraatori teek..... | 18 |
| 6 | Projekti ülesehitus ja struktuur..... | 19 |
| 6.1 | Kasutajaliides..... | 19 |
| 6.1.1 | Mallimärgised..... | 19 |
| 6.2 | Tagarakendus..... | 20 |
| 6.2.1 | Seaded..... | 20 |
| 6.2.2 | Mudelid..... | 20 |
| 6.2.3 | Vaated..... | 21 |
| 6.2.4 | Audentimine..... | 22 |

| | | |
|-------|--|----|
| 6.2.5 | Veebiaadressid..... | 23 |
| 6.2.6 | Vormid..... | 23 |
| 7 | Rakenduse realiseerimine..... | 24 |
| 7.1 | Audentimine..... | 24 |
| 7.2 | Klubiga liitumine..... | 24 |
| 7.2.1 | Klubi loomine..... | 24 |
| 7.3 | Klubi manageerimine..... | 25 |
| 7.3.1 | Klubiväline kasutaja..... | 25 |
| 7.3.2 | Klubi liige..... | 25 |
| 7.3.3 | Klubi treener..... | 26 |
| 7.3.4 | Klubi peatreener..... | 26 |
| 7.4 | Sportlaste lisamine..... | 26 |
| 7.5 | Võistluse loomine..... | 26 |
| 7.5.1 | Vanusekategoriate lisamine, muutmine, kustutamine..... | 27 |
| 7.5.2 | Failide üleslaadimine..... | 28 |
| 7.6 | Võistlusele registreerimine..... | 28 |
| 7.7 | Tühivõistlejate lisamine..... | 29 |
| 7.8 | Võistluse eksportimine..... | 30 |
| 7.9 | Tulemused..... | 31 |
| 8 | Arengusuunad..... | 32 |
| 8.1 | Laienemine..... | 32 |
| 8.2 | Tehnoloogia..... | 32 |
| 8.3 | Lisa teenused..... | 33 |
| 9 | Kokkuvõte..... | 34 |
| | Kasutatud kirjandus..... | 35 |
| | Lisa 1 – Võistluse andmebaasi mudel..... | 36 |

Jooniste loetelu

| | |
|---|----|
| Joonis 1: Rakenduse andmebaasi ERD mudel..... | 22 |
| Joonis 2: Klubi detailvaade..... | 25 |
| Joonis 3: Võistluse Vanusekategoriad..... | 27 |
| Joonis 4: Sportlaste Registreerimine võistlusele..... | 28 |
| Joonis 5: Tühivõistleja registreerimine..... | 29 |
| Joonis 6: Võistluse eksport..... | 30 |
| Joonis 7. Võistluse mudeli näide django ORM põhjal..... | 36 |

1 Sissejuhatus

Eestis toimub keskmiselt 20-30 võistlust aastas, kus võistlejate arv ulatub 150-st kuni 700 võistlejani. Iga võistlus eeldab ligikaudu 30 tundi käsitööd, mis hõlmab endas võistlusjuhendi loomist, võistluste reklaamimist, klubide kaasamist võistlusesse ning võistlejate käsitsi sisestamist võistlusprogrammi, mis osutub kohati kõige mahukamaks ja suurimaks veakohaks. Käsitsi vanuseklasside valideerimine ning sisestamine pärsib vigu, suuresti, kuna see jäetakse viimasele hetkele ning seda tehakse üleöö enne võistlust.

Loodavas veebirakenduses realiseeritakse võimalus kasutajatel liituda ja/või luua klubi, millega saavad teised kasutajad liituda ning läbi selle luua võistluseid, mis võimaldavad teistel treeneritel võistlusele võistlejaid registreerida. Võistlejate registreerimisel toimub võistleja vanuse alusel vanusekategoria valideerimine vastavalt võistluses olevatele vanusekategoriatele. Treener valib võistlejale kaalukategoria ning registreerib sportlase võistlusele.

Veebirakendus on peamiselt suunatud Judole kuid sobib ideaalselt ka muudele spordialadele mis kasutavad vanuse ja kaalukategoriaid, näiteks Sumo, Jiu-jitsu, Karate jne.

Veebirakendus on ehitatud pythoni põhisel veebiraamistikul Django ja rakenduse eesliide genereeritakse Django tagaraamistikku poolt. Eesliide kasutab andmete kuvamiseks „*Django template language*” teeki.

2 Autori roll

Käesolevas töös on autori roll luua veebirakendus mis toestab judo võistluste korraldamist ning võistlustele registreerimist.

Töö raames sooritatavad tööd hõlmavad endas serveri arhidektuuri loomist, andmebaasimudelite disaini ning nendevahelisi seoseid, kasutajaliidese disaini, funktsionaalsuse arendust, funktsionaalsuse testimist, autorile parimate teadaolevate tavade ja teadmiste kohaselt. Loodud veebirakendust rakendatakse reaalse võistluse peal.

Arenduse käigus valitakse tööriistad ja vahendid vastavalt vajadusele.

3 Projekti skoop

Lõputöö käigus valmiv projekt on judo võistluste korraldamiseks ning kõigeks mis hõlmab võistlustele registreerimist. Projekti raamesse jääb samuti ka klubi klubide minimaalse majandamisega seotud tegevused.

Bakalaurusetöö skooopi jäävad

- Projekti ning selle osade analüüsimine
- Projekti kuuluvate teekide valik
- Rakenduse mudelite kirjeldamine
- Rakenduse realisatsiooni kirjeldus
- Rakenduse arendus ja testimine
- Loodud teenuse rakendamine reaalses võistluses

4 Analoogete lahendused

Bakalaurusetöö käigus valminud veebirakendusele on loodud ka sarnaseid teenuseid millel on sarnane eesmärk.

Suurimaks konkurendiks osutub „Sportdata.org” [1] mis on rahvisvaheline spordivõistluste registreerimise platvorm mis hõlmab endas erinevaid spordialasid nagu judo, jiu jitsu, karate, wako, mma, taekwon-do, wushu ja paljud teised. Veebirakendus pakub täielikku funktsionaalsust lõputöös kajastatud probleemide lahenduseks sealhulgas võistluste loomine, registreerimine, kasutajate seostamine klubidega ning võistluste eksportimine.

Analoogsete lahenduste seas on samuti Läti süsteem „Sportists” [2], kuid nende süsteemil on suured vead, mis vähendavad kasutajakogemust. Läti judoliidu poolt välja töötatud süsteemi peamine erinevus ja puudujääk on kasutaja iseseisvus. Kasutajatel ei lubata ise luua võistluseid, vaid makstakse teenuse eest. Sportist ei paku võistluse loojale allalaadimise võimalust, mis tähendab, et see ei lahenda mitte ühtegi probleemi. Sportisti ei paku kasutajatele vabadust. Iga judo klubi jaoks on loodud üks konkreetne kasutaja, mis tähendab, et sportlaste sisestamine on kasutaja omaniku hoolt kanda. Treeneritel pole võimalust hallata enda võistlejate infot, kuna eraldi treeneritele kontosid ei looda.

5 Analüüs

Loodud projektis osalevad erinevad tehnoloogiad, mis edendavad kasutajakogemust ning veebiteenuse funktsionaalsust. Seal hulgas django mudelid andmebaasi ja selle tabelite loomiseks ning sisenditega suhtlemiseks.

5.1 Andmebaas

Lõputöös valmiv projekt kasutab andmete hoiustamiseks MySQL andmebaasi. Andmebaasi tabelid luuakse ja genereeritakse läbi Django mudelite.

5.1.1 Mudelid

Django veebiraamistik defineerib oma andmebaasitabelid läbi mudelite. Läbi nende mudelite manipuleeritakse ka neid tabeleid.

Veebirakenduses on kasutusel järgnevad mudelid: *Club*, *Competition*, *FileInCompetition*, *EmptyJudokaInCompetition*, *CustomUser*, *UserInCompetition*, *Belt*, *UserPendingInClub*, *Category*, *WeightCategoryInCompetition*, *Judoka*, *JudokaInCompetition*, *Results*, *Post*.

CustomUser ehk kasutaja on Django veebiraamistiku poolt loodud tavakasutajamudeli edasiarendus, mis vastaks loodava rakenduse vajadustele. Kasutajamudelile on juurde lisatud vajalikud väljad, et tuvastada kasutaja rolli rakenduses (treener, peatreener, lihtkasutaja). Vastavalt rollile avalduvad erinevad teenused kasutaja jaoks.

Judoka ehk sportlane on mingisuguse klubi alla allutatud sportlase objekt mida saab kasutada võistlustes

JudokaInCompetition ehk judoka võistluses on vahetabel *Judoka* ja *Competition* vahel kus on määratud kindel kategooria ja kaal milles *Judoka* võistlema hakkab.

UserInCompetition ehk kasutaja võistluses on vahetabel kasutaja ja võistluse vahel, läbi mille seostada kindlat kasutajat kindlas võistluses ning kasutada seda informatsiooni võistluse *exportimisel*.

Belt ehk vööde mudel defineerib standard vööd mida määrata kasutajatele ja sportlastele.

UserPendingInClub ehk kasutaja ootel klubis on tabel kus hoiustada informatsiooni kasutajatest kes on esitanud liikmelisustaotluse klubile ning ootab vastuvõtukinnitust. Kui taotlus rahuldatakse kustutatakse tabelist selle kasutaja andmed ning määratakse kasutajale klubi.

Club ehk klubi objekt ühendab omavahel treeneriteks määratud kasutajad oma sportlastega ning ühendab kokku klubi treenerite poolt loodud võistlused. Lisaks hoiustab klubi endas klubi kontaktandmeid, juhul kui kellelgi tekiks vajadus nende järele.

Post ehk postitus on mudel klubisiseste postituste mudel.

Competition ehk võistlus, hoiustab detailset informatsiooni võistluse kohta ning ühendab endaga võistlusesse registreeritud sportlased, vanusekategoriad, täpsustavad failid ning pakub võimalust võistlust *exportida*.

Category ehk kategooria on vanusekategoriate mudel kus on märgitud vanusekategooria tähis ja maksimaalne vanus, mille alusel valideerida sportlaste sobivust võistlusesse. Kasutajatele pakutakse registreerimisel vaid neid kategooriaid kuhu nad algoritmi alusel sobivad.

WeightCategoryInCompetition ehk kaalukategooria võistluses on vahetabel võistluse ja vanusekategooria vahel mis kinnitab millised kategooriad kindlas võistluses osalevad. Mudelis on kirjas ka kaalud mis selles vanusekategorias oodatud on.

FileInCompetition ehk fail võistluses, lisainformatsiooni edastamiseks, juhendite, piltide jagamiseks.

EmptyJudokaInCompetition ehk tühi judoka võistluses, on mudel sportlasest kes ei kuulu ühtegi registreeritud klubisse. Mudeli eesmärk on lisada rakendusele paindlikkust,

juhul kui osad klubid otsustavad mingil muul meetodil registreerida enda sportlasi võistlusele.

Results ehk tulemused on andmebaasi tabel kus määratud võistlusele saab juurde lisada viite selle võistluse tulemustele.

5.1.2 Päringud

Päringud andmebaasi pihta toimuvad läbi Django ORM-i. Django ORM on raamistiku poolt pakutav lahendus andmebaasiga suhelda läbi Python koodi viidates mudelite peale.

Django ORM kiirendab oluliselt arendajatele päringute kirjutamist samas tagades turvalisust, elimineerides ohud *SQL injectioni* ees

Näide:

```
users = CustomUser.objects.all()
club_users = users.filter(club__name="Ookami")
```

5.2 Kasutajaliidese analüüs

Kasutajaliides on osa veebirakendusest mida kasutaja näeb ja saab manipuleerida, et suhelda andmemudelitega.

Veebirakenduse kasutajaliides luuakse kasutades tehnoloogiaid: HTML Javascript, CSS, DTL, JQuery, Ajax. Kasutajaliidese stiilimiseks kasutatakse Bootstrap stiilipakke ning CSS-i. Funktsionaalsuse edestamiseks on kasutuses Javascript, JQuery, DTL.

5.2.1 Bootstrap

„Bootstrap on tasuta tööriistakomplekt veebilehtede ja veebirakenduste loomiseks. Sisaldab JavaScripti pluginaid, HTML- ja CSS-kujundusmalle tüpograafia, vormide, nuppude, navigatsiooni, teadete ja teiste veebiliideste komponentide jaoks.” [3]

Bootstrapi stiilifailid lisatakse veebiliidesele kasutades CDN-i

5.2.2 Javascript

Kasutajakogemuse mugavdamiseks on kasutuses veebipõhine skriptimise keel Javascript, mis lubab teostada asünkroonseid päringuid ja manipuleerida DOM elemente, et saavutada mingisugust soovitud eesmärki.

5.2.3 Ajax

„Ajax on kogum omavahel seotud veebiarenduse tehnikaid, mida kasutatakse rakenduse kliendi poolel interaktiivsete veebirakenduste loomisel. Ajax võimaldab veebilehtedel JavaScriptiga serverisse tagaplaanil päringuid teha ja andmeid vastu võtta, segamata avatud lehe kuvamist ja olekut.” [4] Ajax on kasutuses asünkroonsete päringute teostamiseks, info pärimiseks, et edendada kasutajakogemust ning vähendada mallide üleliigset renderdamist.

5.2.4 Django Template Language

Django template language on tööriist, mis võimaldab manipuleerida DOM-i, kasutades eeldefineeritud märgendeid. Märgenditega saab luua loogikat ning kasutada muutujaid HTML-is, sisu kuvamiseks. „Django mallimise keel on loodud tasakaalustamaks võimsust ja kergust. See on disainitud tundumaks mugav nendele kes on harjunud töötama HTMLiga” [5]

5.3 Tagarakenduse analüüs

Veebirakenduse tagarakendus on kirjutatud pythonipõhises veebiraamistikus nimega Django, mis kasutab erinevaid teeke, et realiseerida vajalik korrektne andmevoog serveri ja kasutajaliidese vahel.

5.3.1 Django veebiraamistik

„Django on kõrgrasemeline python veebiraamistik mis endendab kiiret arendust ja puhast pragmaatilist disaini.” [6]

Django jaguneb suuresti kolmeks osaks, mudel, vaade, mall (*Model, View, Template*), mis surub peale MVT mustrit. Üldiselt tuntud MVC mustri „C” ehk kontrolleri on

realiseeritud Djangosse mistõttu ei pea sellele rõhku panema. Andmevoog läbi rakenduse liigub mallist vaatesse ning sealt mudelisse, mis edastab andmed tagasi vaatele ning edasi malli, renderdamiseks. Renderdamise eest kannab hoolt DTL mis asendab kontekstist kaasa antud väärtused mallides olevate muutujatega.

Django on pythoni põhine veebiraamistik ning python tagab kiire arenduse kuna see on robustne programmeerimise keel mis on kergesti loetav enda disaini poolest ja surub peale puhast koodi ning parimaid tavasid. Django veebiraamistik tagab veebirakenduse turvalisuse, mis suunab arendaja tähelepanu arenduse poole. Djangosse on implementeeritud kasutajamudel ja kasutajate autentimine mis on koheselt valmis kasutamiseks.

5.3.2 Django Countries teek

Countries ehk riigid on vabavaraline teek, mis sisaldab endas kõiki riike ning nende lippe. See pakub mugavat lahendust objektide ja riikide seostamiseks. Antud teek on kasutuses kasutajate ja klubide mudelites, kuna nendel objektidel on vaja seost kindla riigiga. Võistlejatel seostatakse samuti riiki, mis tuleneb klubist mille all objektid registreeritud on.

5.3.3 Django Crispy forms teek

Crispy form on django moodul, et renderdada vorme mallis, kasutades mallimärgistusi. Django vaadetes konteksti lisatud vorm loetakse mallis sisse ning antakse sellele külge „crispy” mallimärgistus, tänu millele stiilitakse antud vorm vastavalt *crispy* stiilireeglitele. Crispy teek tagab vormidele kindlad stiilireeglid ning hoiab kokku arendaja aega, et seda kuskil mujal kasutada.

5.3.4 Django i18n teek

Rakendus toetab erinevaid keeli ning nende vahel vahetamine toimub läbi django-i18n paketi. I18n on djangosse sisse ehitatud ning implementeerimine on kerge. I18n loeb tõlked sisse .po laiendusega failidest mis koosneb „msgid” ja „msgstr” elementidest, mis defineerivad ära, millist teksti milleks tõlkida.

5.3.5 Pythoni Environ teek

Salajaste ja muude muutujate hoiustamiseks kasutatakse .env failiformaadiga faile. Nendele failidele ligipääsemine on loodud python paketiga „environ”. Env failis hoiustatakse salajast võtit mille alusel django genereerib päringute autentimiseks ja valideerimiseks sõnesid, andmebaasi ühenduse sõne ning muid rakenduse tööga seotud muutujaid.

5.3.6 Django administraatori teek

Djangoga tuleb automaatselt kaasa ka administreeriv vaade mudelite manipuleerimiseks, millele saab ligi ainult kindel kasutaja kellele on määratud „superkasutaja” staatus. Administraatoriks registreeritud kasutaja saab manipuleerida mudeleid mis on spetsifitseeritud läbi koodi. Django administraatori vaade võimaldab arendajatel hoida kokku väärtuslikku aega, et luua sarnast teenust andmebaasi sisu kuvamiseks ja manipuleerimiseks. Admin vaade võimaldab arendajal luua uusi objekte, uuendada vanu ning kustutada neid, mis teeb sellest teenusest hädavaljaliku osa rakenduses. Soovi korral saab kerge vaevaga lisada statistika mooduleid ja muud vajalikku nendele vaadetele, mis tõstab veelgi administraatori vaate väärtust arendajatele. „Django üks võimsamaid osi on automaatne administraatori liides. See loeb teie mudelite metaandmeid, pakkudes kiiret ja mudelikeskset liidest, kus usaldusväärsed kasutajad saavad teie saidi sisu hallata.” [9]

6 Projekti ülesehitus ja struktuur

Ülevaade projekti ülesehitusest ning django suhtlusest tema osadega.

Django arendus jaguneb suures osas kaheks: kasutajaliides ja tagarakendus. Selles peatükis kirjeldan Django projekti üldist ülesehitust nii kasutajaliidesele kui ka rakenduse tagarakendusele.

6.1 Kasutajaliides

Kasutajaliidese loomine koosneb mallide loomisest, mis paigutatakse projekti juurkausta alamkausta, nimega „templates”. Käesolevas töös on jagatud iga kasutajaliidesele olev leht kaheks. Navigatsiooni moodul ning ülejäänud lehe sisu. Kuna navigatsiooniriba („nav bar”) on staatiline ning igasse malli selle sisse kirjutamine vähendaks loetavust ning pärsiks probleeme, on navigatsiooni riba paigutatud eraldi faili ning lisatud kasutades django enda poolt pakutud malli märgiseid. Kasutajaliidesele on kasutatud palju mallimärgiseid, et luua loogikat, kui ka vähendada koodi ning muuta liides dünaamiliseks olenemata muutujate sisust või kogusest. Mallid on jagatud kaustadeks vastavalt malli sisule.

6.1.1 Mallimärgised

Loodud rakenduse kasutajaliidesele on kasutatud django poolt loodud mallimärgistusi, et genereerida mallide sisu. Sisus edastatakse mallidele läbi konteksti sõnastiku. Sõnastiku võtmeväärtused tõlgendatakse mallis samanimelisteks väärtusteks ning nendele saab ligi kasutades märgiseid. Muutväärtuste manipuleerimiseks on kasutuses „`{{ }}`” mallimärgis kuhu vahele asetatakse muutuja nimetus.

Et vältida HTML koodi korduvkasutust on kasutuses `{% for x in y %}` mallimärgistus, kus „x” on individuaalne element massiivist „y”

Tõeväärtusloogika kasutamiseks on kasutuses mallimärgistus `{% if x %}` kus „x” on tõeväärtus võrdlus või väärtus, mille korral rakendatakse sellele märgistusele järgnev kood.

6.2 Tagarakendus

Tagarakendus jaguneb funktsionaalsuse põhjal osadeks. Django kaust koosneb rakendustest, mis tähendab et ühte projekti saab paigutada mitu erinevat rakendust. Django jagab ära mudelid, URL-id, vaated, mallid, vormid kindlatesse kaustadesse või failidesse juba projekti genereerimise ajal. Struktuur on projekti genereerimise ajal paika pandud, mis on django arenduse puhul standard.

6.2.1 Seaded

„Django seadete fail sisaldab kogu teie Django installi konfiguratsiooni. Selles dokumendis selgitatakse, kuidas seaded toimivad ja millised seaded on saadaval.” [10]

Django projekti töö toimib vaid settings.py failis asuvate muutujate väärtustele. Seal määratakse ära erinevad rakenduse toimimiseks vajalikud väärtused nagu näiteks: staatiliste failide hoiustamise asukohad, andmebaasi ligipääsuks vajalikud andmed, rakenduse kasutaja vaikemudel, projekti tööks vajalik vahevara, projektis töötavad rakendused ning vajalikud paketid rakenduse tööks.

6.2.2 Mudelid

„Mudel on teie andmete ainus ja kindel teabeallikas. See sisaldab teie salvestatud andmete olulisi välju ja käitumist. Üldiselt kaardistab iga mudel ühe andmebaasi tabeli.” [7]

Mudelid paigutatakse models.py faili sisse mis asub rakenduse kaustas. Mudelid kirjutatakse välja pythoni klassidena mis tulenevad *django models* moodulist. Läbi mudelite saab ka manipuleerida vaikekasutaja mudelit ning lisada vajalikke välju juurde vastavalt vajadusele. Mudelite välju defineeritakse kasutades „models” paketi sisalduvaid klasse. Igale mudelile saab juurde lisada funktsionaalsust mida hiljem kasutada mingisuguse objekti juures.

6.2.3 Vaated

„Vaatefunktsioon või vaade on Pythoni funktsioon, mis võtab vastu veebipäringu ja tagastab veebivastuse.” [11]

Vaated asuvad projekti kaustas ning failis nimega views.py. Vaadete loomisel on mitu varianti. Kas luua funktsioonipõhised vaated või klassipõhised. Antud töö käigus otsustati luua klassipõhised vaated, kuna klassipõhiste vaadete haldamine on selgem ning töökindlam. Klassipõhised vaated suhtlevad mallidega läbi konteksti. Konteksti saab paigutada kõik vajaliku mida vaates vaja läheb aga vaatele kinnitatud mudelis ei eksisteeri. Mudelist tulenevatele vaadetele määratakse kindel mudel mille kohta vaade kehtima hakkab, olgu see loomine, vaatamine, uuendamine või kustutamine.

Klassipõhised vaated pärivad enda funktsionaalsuse vastavalt määratud parameetritele. Erinevaid vaate tüüpe on palju. Peamisteks on *DetailView*, *ListView*, *UpdateView*, *DeleteView*, *FormView*, *CreateView* ja paljud teised

Kõikidele vaadetele, mis kehtivad mingi kindla mudeli objekti kohta, tuleb URL-i lisada juurde täpsustav väli, et tuvastada millist objekti vaadata, uuendada, kustutada soovitakse.

DetailView ehk detailvaade on mingisugusele mudelile vastava objekti detailvaade. Vaade kinnitab enda konteksti URL-ist tuleneva muutujaväärtusele vastava mudeli objekti, mida saab manageerida mallis endas.

ListView ehk massiivi vaade on mudeli kõikide objektide vaade. Konteksti on lisatud mudelile vastavad objektid mida mallis manageerida.

UpdateView ehk uuendusvaade on ühe mudeli objekti andmete uuendamise vaade. Konteksti on paigutatud antud mudeli objekti väljadest genereeritud vorm.

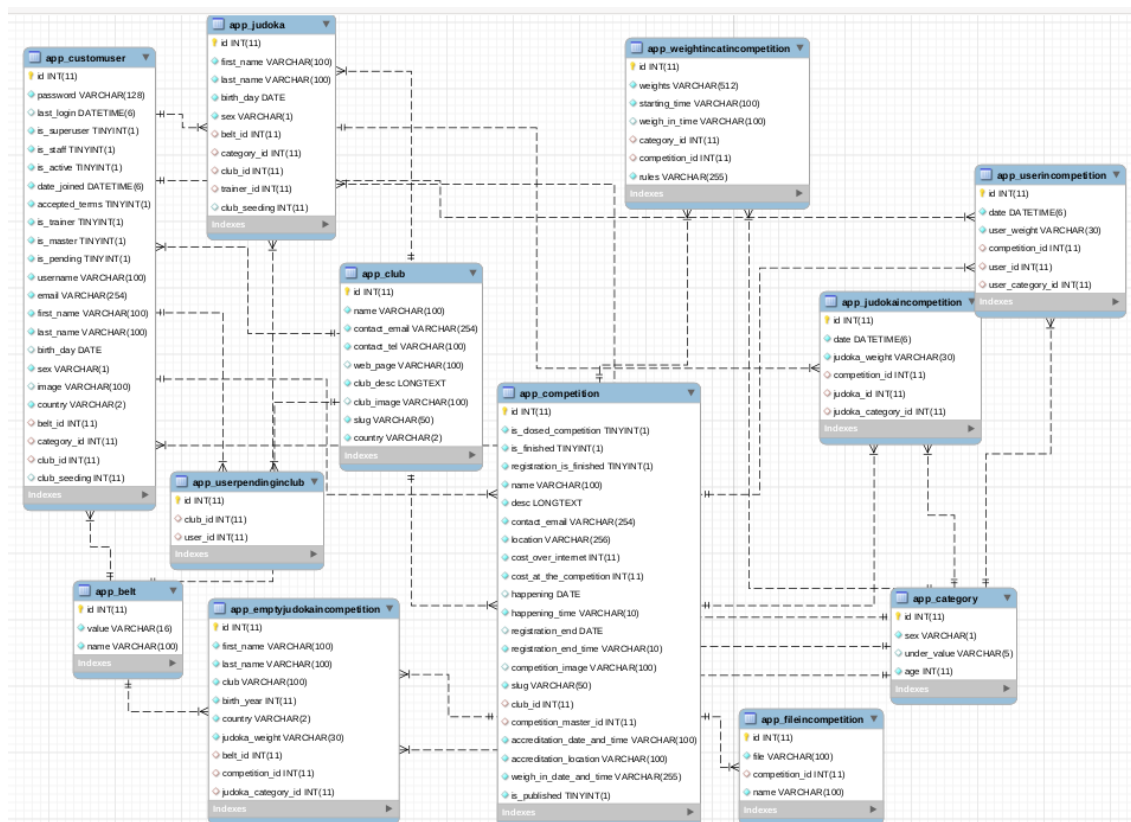
DeleteView ehk kustutusvaade on ühe mudeli objekti kustutamiseks loodud vaade.

FormView ehk vormivaade on vaates täpsustatud formi kuvamise vaade. Konteksti paigaldatakse täpsustatud vorm.

CreateView ehk loomisvaade on täpsustatud mudeli objekti loomise vaade. Konteksti lisatakse mudeli väljade põhjal genereeritud form.

Antud töös on samuti kasutuses ka funktsioonipõhised vaated, et suhelda asünkroonsete ajax päringutega. Nendel juhtudel ei peetud vajalikuks luua klassipõhist vaadet kuna funktsioonil oli ainult üks kindel eesmärk.

Mudelitele vastav ERD on genereeritud kasutades MySQL Workbench-i.



Joonis 1: Rakenduse andmebaasi ERD mudel

(Allikas: autori koostatud)

6.2.4 Audentimine

Teatud vaadetele ja funktsionaalsusele on tagatud piirangud. Vaadete piirangud on teostatud kasutades vaate klassile lisatud *LoginRequiredMixin* parameetriga, mis tagab ligipääsu vaid sisselogitud kasutajale. „Djangoga on kaasas kasutaja autentimissüsteem. See haldab kasutajakontosid, rühmi, õigusi ja küpsistel põhinevaid kasutajaseansse.” [12]. Päringute piiramine mittesobivale kasutajale vastavalt kasutaja staatusele on piiratud kasutades loogikat, võrreldes kasutaja andmeid funktsiooni ligipääsulubadega.

6.2.5 Veebiaadressid

Veebirakenduse aadressid on kirja pandud kahes eri failis. Üks nendest failidest asub projekti kaustas mis suunab liiklust erinevatele rakendustele ning teine asub rakenduse kaustas ning suunab liiklust rakenduse siseselt. Aadresse hoitakse massiivis ning vastavalt päringule suunatakse päring ümber aadressis märgitud vaatele. Aadress koosneb minimaalselt kahest osast: aadressi enda väärtus ning sellele aadressile vastav vaade. Antud projektis on lisatud ka kolmas väärtus ning selleks on „name” ehk nimi, rakendusesiseseks aadressi viitamiseks. Määrates aadressile nime ei pea kasutaja ümbersuunamisel kasutama tervet aadressi sõneväärtust vaid saab vaid viidata selle nimele ning malli renderdamisel asendatakse nimi, aadressi sõneväärtusega.

Näide:

```
urlpatterns = [  
    path("clubs", ClubListView.as_view(), name="clubs"),  
]
```

6.2.6 Vormid

Django vormid on django moodul vormide loomiseks. Vormide eesmärk on kasutaja sisendi pärimine. Django vormid tagavad eos kasutaja sisendi puhastuse mis takistab pahatahtlikke sõnede sisestamist. Vormide sisendid on kaitstud, kasutades CSRF tähiseid ning automaatne puhastus SQL injection rünnakute vastu. Vaikesätetena puhastavad vormid ka sisestatud teksti XSS rünnakute eest kuid neid sätteid saab üle kirjutada vajadusel.

7 Rakenduse realisatsioon

Veebirakendus koosneb erinevatest osadest, mis on kombineeritud vastavalt django standarditele, kasutades autorile parimaid teadaolevaid tavasid ja tehnoloogiaid.

7.1 Audentimine

Veebirakenduse peamine funktsionaalsus on piiratud vaid sisselogitud kasutajatele. Sisselogitud kasutajad jagunevad eri gruppidesse: tavakasutajad, treenerid, peatreenerid ning personal. Kasutajate funktsionaalsusele ligipääsu audentimine käib läbi funktsiooni ligipääsulubade võrdlemise.

7.2 Klubiga liitumine

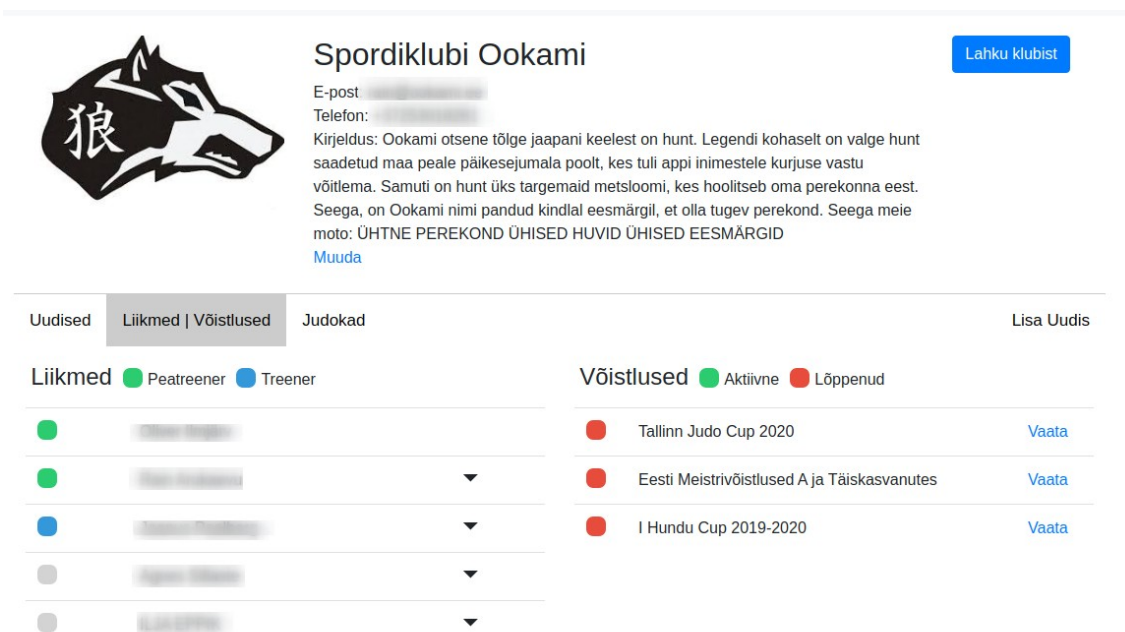
Veebilehe funktsionaalsuse täis kasutust eeldab kasutaja klubis olemist. Klubiga liitumisel peab kasutaja navigeerima ennast klubide lehele ning vajutama „liitu” nuppu, tänu millele kontrollitakse kasutaja klubi staatust (kas on juba liige klubis või ei). Kui kasutaja ei kuulu veel ühtegi klubisse, luuakse andmebaasi *UserPendingInClub* tabelisse sisend kasutaja ja klubi võtmetega. Kasutajale määratakse tõeväärtus staatus *is_pending* tõeseks. Kasutaja muutub nähtavaks klubi „ootel” lehel ning edasiseks tegevuseks on vaja klubi peatreeneri nõusolekut liitumiseks. Kasutaja taotluse rahuldamisel määratakse tema klubi staatus vastavalt.

7.2.1 Klubi loomine

Juhul kui kasutaja klubi ei eksisteeri, saab kasutaja klubi ise luua. Navigeerides klubi loomise lehele, täites ära vajaliku informatsiooni, luuakse klubi ning klubi peatreeneriks lisatakse klubi looja. Kasutaja tõeväärtus muutuja *is_master* ja *is_trainer* väärtus muudetakse tõeseks.

7.3 Klubi manageerimine

Vastavalt kasutaja staatusel, pakutakse klubi liikmele erinevat funktsionaalsust. Klubi alla kogutakse kokku kõik objektid mis omavad selle klubi välisvõtit ning pakutakse kasutajatele kuvamiseks või muutmiseks.



Spordiklubi Ookami [Lahku klubist](#)

E-post: [redacted]
Telefon: [redacted]

Kirjeldus: Ookami otsene tõlge jaapani keelest on hunt. Legendi kohaselt on valge hunt saadetud maa peale päikesejumala poolt, kes tuli appi inimestele kurjuse vastu võitlema. Samuti on hunt üks targemaid metsloomi, kes hoolitseb oma perekonna eest. Seega, on Ookami nimi pandud kindlal eesmärgil, et olla tugev perekond. Seega meie moto: ÜHTNE PEREKOND ÜHISED HUVID ÜHISED EESMÄRGID

[Muuda](#)

Uudised | **Liikmed | Võistlused** | Judokad Lisa Uudis

Liikmed ● Peatreener ● Treener

| Liikme tüüp | Nimi | Statust |
|--------------|------------|---------|
| ● Peatreener | [redacted] | |
| ● Peatreener | [redacted] | ▼ |
| ● Treener | [redacted] | ▼ |
| ● [redacted] | [redacted] | ▼ |
| ● [redacted] | [redacted] | ▼ |

Võistlused ● Aktiivne ● Lõppenud

| Võistluse nimi | Statust |
|---|------------|
| Tallinn Judo Cup 2020 | ● Lõppenud |
| Eesti Meistrivõistlused A ja Täiskasvanutes | ● Lõppenud |
| I Hundu Cup 2019-2020 | ● Lõppenud |

Joonis 2: Klubi detailvaade

(Allikas: autori koostatud)

7.3.1 Klubiväline kasutaja

Klubivälisel kasutajal on võimalus vaadelda klubi ajajoonel asuvaid teateid, klubi liikmeid ning näha klubi kontaktandmeid. Klubivälisele kasutajale pakutakse võimalust liituda klubiga, kuvades „Liitu” nuppu, mida vajutades lisatakse andmebaasi sisend, ootel olevast kasutajast ning kuvatakse klubi treeneritele. Treenerid saavad võtta taotluse vastu või lükata ümber.

7.3.2 Klubi liige

Klubi liige kes ei ole treener ega peatreener saab vaadelda kogu klubisisest infot, selhulgas sportlaseid, liikmeid, võistluseid, üldist klubi infot ning ka klubisiseseid postitusi.

7.3.3 Klubi treener

Klubi treenerile on lisatud ligipääs teatud funktsionaalsusele. Lisaks info vaatlemisele saab treener uuendada klubisisest informatsiooni, lisada endale sportlaseid ning nende andmeid muuta. Treeneril on samuti ligipääs uudiste lisamisele läbi mille informeerida teisi treenereid ning võtta vastu liitumistaotluseid.

7.3.4 Klubi peatreener

Peatreeneritele on reserveeritud funktsionaalsus luua ja muuta võistluseid, ülendada ja alandada klubi kasutajaid (muuta treeneriteks ja alandada ametikohalt), muuta kõiki sportlaseid, luua uusi sportlaseid ning määrata neile treener.

7.4 Sportlaste lisamine

Sportlaste lisamine on reserveeritud klubi treeneritele ja peatreeneritele. Sportlaste olemasolu tagab veebilehe sujuva töö ja muudab treenerite tegevuse efektiivsemaks.. Sportlase objekt koosneb eesnimest, perekonnanimest, sportlase vööastmest sünnikuupäevast, sportlase soost, mille põhjal hilisemaid arvutusi tegema hakatakse. Sportlase objektile lisatakse automaatselt kaasa klubi ja treenerkasutaja välisvõtmetena. Iga sportlase pärimisega arvutatakse tema vanusekategorია ning määratakse talle samuti välisvõtmena, et sooritada edasiseid arvutusi võistluse vanusekategoriasse sobivuse jaoks.

7.5 Võistluse loomine

Võistlust saavad luua peatreeneri saatust omavad kasutajad. Võistluse loomiseks peab kasutaja navigeerima võistluse loomise lehele ning täitma võistluse üldinfo, mis sisaldab endas informatsiooni võistluse asukoha, maksumuse ja kontaktandmete kohta. Esitades vormis täidetud sisu luuakse peidetud võistlus, mida saab kasutaja avalikustada siis kui kõik vajalik info on täidetud.

7.5.1 Vanusekategoriate lisamine, muutmine, kustutamine

Vanusekategoriad võistluses pakuvad võistlejatele kindlat kategooriat kuhu ennast registreerida tänu millele võistlejaid jagada. Kategooria lisamisel võistlusesse tuleb valida rippmenüüst, kategooriaga ning täpsustada sellele vanusekategoriale ka osalevad kaalukategooriad, reeglid ja vanusekategorია orienteeruv algusaeg võistlusel. Kaalukategooriate hoiustamine toimub sõnedena ning töödeldakse käsitsi kasutades loogikat, kuna korraldajatel ei pruugi olla planeeritud mingeid kindlaid kaalukategooriaid vaid need võivad selguda alles peale registreerimise lõppu. Vajaduse korral saab korraldaja sisestada kaalukatekoooria asemel kohahoidja, millele sportlased registreerima hakkavad.

Kategooriate muutmise link asub vaneusekategorია kuva kõrval, individuaalselt igale kategooriale. Sellele funktsionaalsusele on ligipääs vaid võistluse loojal. Kustutamise funktsionaalsus asub kategooria kustutamise lehel.

| | | |
|--|---------------------------------------|---|
| U12 2009-2011 | Starting 8 veb. 9:00-11:00 | Rules: shime- and Kansetsu waza not allowed |
| M -25kg, -27kg, -30kg,-34kg, -38kg, -42kg, -46kg, +46kg | |  |
| W -28kg, -31kg, -35kg, -40kg, +40kg | |  |
| U14 2007-2009 | Starting 8 veb. 11:00-13:00 | Rules: shime- and Kansetsu waza not allowed |
| M -30kg, -34kg, -38kg, -42kg, -46kg, -50kg, -55kg, -62kg, -70kg, +70 kg | |  |
| W -32kg, -36kg, -40kg, -45kg, -52kg, +52kg | |  |

Joonis 3: Võistluse Vanusekategoriad

(Allikas: autori koostatud)

7.5.2 Failide üleslaadimine

Võistluste lisainformatsiooni pakkumiseks on kasutajal võimalus üles laadida lisa faile, kus on kirjas täpsustav info võistluse, majutuse või muu kohta. Failiformaati ei kontrollita ning kasutaja saab üles laadida erinevate laienditega faile. Üles laetud dokumendid hoiustatakse sättes täpsustatud asukohta ning viidatakse mallides, kasutades kausta ja failikombinatsiooni aadressi.

7.6 Võistlusele registreerimine

Sportlaste registreerimine võistlusele on lihtsustatud miinimumini. Edukaks registreerimiseks peab treener valima sportlasele vaid kaalukategooria ja sportlase tugevuse astme klubisiseste sportlaste konkurentsi vähendamiseks. Sportlase registreerimiseks võistlusele arvutatakse igale sportlasele sobivad vanusekategoriad ning kogutakse kokku iga sportlase alla, samuti kogutakse kokku ka kategooriad kus sportlane on juba registreerunud ning neid kuvatakse vastavalt. Igal sportlasel on võimalus osaleda võistlusel kahes eri kategoorias.

Registration for Tallinn Judo Cup 2020 ← tagasi

Search... Open

| Ees- ja perekonnanimi - 2009 - WU12 | | | |
|-------------------------------------|--------|------|--------|
| Category | Weight | Seed | Action |
| WU12 | ---- | ---- | Reg |
| WU14 | ---- | ---- | Reg |

| Ees- ja perekonnanimi - 2009 - MU12 | | | |
|-------------------------------------|--------|------|--------|
| Category | Weight | Seed | Action |
| MU14 | ---- | ---- | Reg |
| MU12 | ---- | 1 | Cancel |

| Ees- ja perekonnanimi - 2007 - MU14 | | | |
|-------------------------------------|--------|------|--------|
| Category | Weight | Seed | Action |
| MU14 | ---- | ---- | Reg |
| MU12 | ---- | 1 | Cancel |

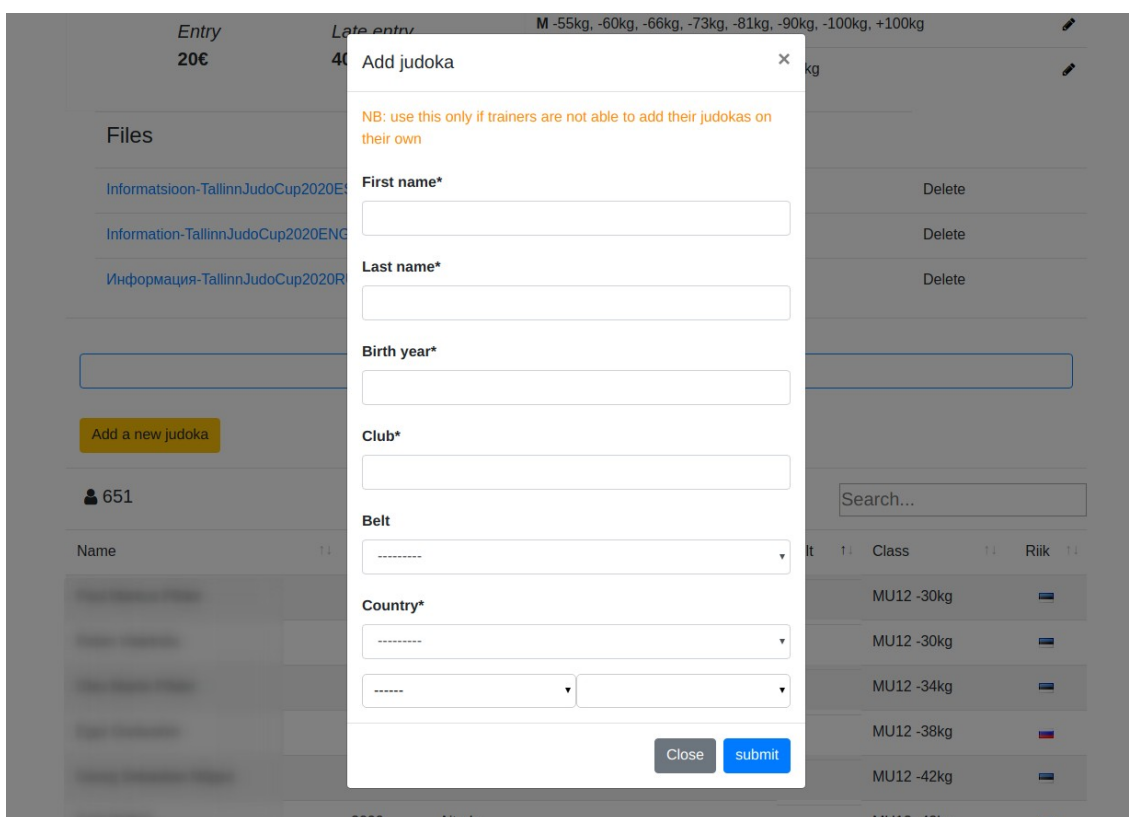
Weight dropdown menu options: -30kg, -34kg, -38kg, -42kg, -46kg, -50kg, ...

Joonis 4: Sportlaste Registreerimine võistlusele

(Allikas: autori koostatud)

7.7 Tühivõistlejate lisamine

Rakendust luues tuleb arvestada olukorraga, kus kõik registreerumised ei toimu läbi pakutava platvormi vaid läbi meilisuhtluse või suusõnalise info edastamisega. Tänu sellele võimalusele peab olema ka pakutud lahendus sisestada sportlasi, keda ei eksisteeri platvormi andmebaasis. Lahendus püstitatud probleemile on käsitsi sportlase sisestamise vorm, mis ei loo uut sportlast vaid lisab ainult sisendi võistlejate nimekirja võistluses. Tühivõistleja objekt on identne sportlasele, kuid klubi välisvõtme asemel on klubi asendatud sõnega. Selline lähenemine muudab tühivõistleja andmete moondamise CSV formaati identseks sportlase objektiga.




The screenshot shows a web application interface. In the foreground, a modal window titled "Add judoka" is open. It contains a form with the following fields: "First name*", "Last name*", "Birth year*", "Club*", "Belt" (a dropdown menu), and "Country*" (a dropdown menu). Below the "Country*" field, there are two more dropdown menus. At the bottom of the modal, there are "Close" and "submit" buttons. A note in orange text reads: "NB: use this only if trainers are not able to add their judokas on their own". In the background, a table is visible with columns for "Name", "Class", and "Riik". The table contains several rows of data, including "MU12 -30kg", "MU12 -34kg", "MU12 -38kg", and "MU12 -42kg".

Joonis 5: Tühivõistleja registreerimine

(Allikas: autori koostatud)

7.8 Võistluse eksportimine

Veebirakenduse täiendav osa on võistluse võistlejate eksportimine teise, võistlust korraldavas rakendusse nimega „JudoShiai” [13]. JudoShiai võimaldab korraldajal luua võistluse ja sisestada võistluses osalejate andmed, kas käsitsi või importides andmed CSV failist. CSV fail genereeritakse kõikidest registreeritud võistlejate andmetest ning pakutakse korraldajale allalaadimiseks võistluste seadete rippmenüüst. Faili read koosnevad eesnimest, perekonnanimest, võistleja soost, spordiklubi nimest, vööastmest, sünnikuupäevast, riigist, vanusekategorias ja kaalukategorias. Kogu info on JudoShiai jaoks vajalik, et võistlust edukalt juhtida. CSV fail kompilleeritakse kasutades python „csv” teeki ning sisu luuakse vastavalt päringutest saadud tulemustele.



Settings ▾

Competition by Spordiklubi Ookami Tallinn Ju

Major judo competition for children and young adults. Tallinn Judo Cup 2020 takes place on February the 8th 2020 in Tallinn Kalev Sports facilities. The tournament is organized by the Judo School Ookami in co-operation with the Estonian Judo Association. The tournament is taking place already 11 years in a row and is one of the biggest of its kind in Scandinavia. Children and young adults in age classes U-12, U-14, U-16, U-18 and U-21 are expected to participate. In addition to Estonia, 15 countries will be represented about 700 participants in total: Finland, Sweden, Latvia, Lithuania, Russia, Ukraine, Belarus, Poland, Ireland, Italy, UK, Norway, Austria, Switzerland, Germany and France. Several companies selling judo sports gear and clothes have their stands where people can purchase them during the competition in the Kalev Sports facilities.

Welcome to the snowy Tallinn!

Registration closes **5. mai 2020**

| U12 2009-2011 | Starting 8 veb. 9:00-11:00 | Rules: shime- and not allowed |
|--|----------------------------|-------------------------------|
| M -25kg, -27kg, -30kg, -34kg, -38kg, -42kg, -46kg, +46kg | | |
| W -28kg, -31kg, -35kg, -40kg, +40kg | | |

| U14 2007-2009 | Starting 8 veb. 11:00-13:00 | Rules: shime- and Kansetsu waza not allowed |
|---|-----------------------------|---|
| M -30kg, -34kg, -38kg, -42kg, -46kg, -50kg, -55kg, -62kg, -70kg, +70 kg | | |
| W -32kg, -36kg, -40kg, -45kg, -52kg, +52kg | | |

| U16 2005-2007 | Starting 8 veb. 13:00-15:00 | Rules: shime waza allowed and kansetsu waza not allowed |
|--|-----------------------------|---|
| M -42kg, -46kg, -50kg, -55kg, -60kg, -66kg, -73 kg, -81kg, +81kg | | |
| W -40kg, -44kg, -48kg, -52kg, -57kg, -63kg, +63kg | | |

Edit Details
Upload file
Add Classes
Export
Get all emails

(Allikas: autori koostatud)

Joonis 6: Võistluse eksport

7.9 Tulemused

Võistlustega kaasnevad tulemused, millele ligipääs on hädavajalik erinevatele organisatsioonidele võistluspunktide määramisel. JudoShiai poolt genereeritud võistlustulemuste failide arv ulatub ligi tuhande eri failini, mis hõlmavad endas PDF, HTML ja javascript faile. Nende failide üleslaadimist ning hoiustamist selle projekti raames ei teostata ning võistluste korraldajad peavad leidma muu lahenduse kuidas neid faile avalikult hoiustada. Võistluste korraldajatele pakutakse lahendust edastada võistlustulemusi inimesteni, hoiustades veebiaadressi linki rakenduse tulemuste lehel. Linkide ja võistluste omavaheliseks seostamiseks on loodud andmebaasi vahetabel *Results*, mis hoiustab nii võistluse välisvõtit kui ka võistlustulemustele vastavat veebiaadressit sõnena. Veebiaadressit hoiustatakse Django *URLField* mudeliväljas mis tõlgendub andmebaasis tavaliseks sõneks, kuid *URLField* mudelile on rakendatud vahekontroll, mis teeb kindlaks et tegemist on legitiimse aadressiga.

8 Arengusuunad

Lõputöös valminud projekt on vaid algeline toode, mille edasiarengule oleks turgu palju. Tulevasse arendusse kuuluksid tehnoloogilised muudatused, uued tooted ning teenuse laiendamine erinevatele spordialadele.

8.1 Laienemine

Loodud rakendus on suunatud peamiselt judole, kuid sellel on potentsiaali ka teiste spordialade võistluseid korraldada. Hetke skoobis on kahevõitlusalad, mis toimivad vanusekategoriate põhjal nagu sumo, jiu jitsu, karate ning muud, kuna eelnevalt mainitud spordialad on otseselt tõlgendatavad loodud teenusesse. Uurides teiste spordialade võistlustel osalejate andmete vajadusi ning nende võistluste korradamiseks vajalike programmide võimalikke sisendeid on võimalik tõlgendada praegu kasutusel olev loogika teistele spordialadele, kasutades samu mudeleid kuid muutes väljundist saadavat sisu vastavalt vajadusele.

8.2 Tehnoloogia

Hetkeseisuga on rakendus kirjutatud täielikult django baasile, kuid edaspidised plaanid hõlmavad endas plaani koodi hajutada mitmesse erinevasse teenusesse. Tulevikus on plaanis viia rakenduse kasutajaliides üle javascripti veebiraamistikule React, kuid jätta tagarakendus siiski django peale ning rakendada Django REST raamistikku selleks. Autor plaanib projekti edasiarenduseks luua platvormile maksevõimaluse, võistluste loomise eest tasumiseks kasutades „Stripe” tehnoloogiat. „Saate kasutada Stripe.js API-sid klienditeabe märgistamiseks, tundlike kaardiandmete kogumiseks kohandatavate triipelementide abil ja maksete aktsepteerimiseks brauseri makse API-dega nagu Apple Pay ja maksetaotluse API.” [8]

8.3 Lisa teenused

Veebirakenduse kõrvalproduktina oleks turgu ka võistluseid korraldavale tootele. Praeguseks euroopas domineeriva „JudoShiai” [13] asendusprogramm tänapäevaste tehnoloogiate ja tavadega. Sellisele tootele oleks suur turg kuna JudoShiai arendus on pooleli jäänud ja samavõrdset tasuta teenust ei ole saadaval. Loodav programm oleks seotav veebirakendusega ning lihtsustaks veelgi võistluste loomist ning samuti pakuks lahendust võistluste korraldamiseks.

9 Kokkuvõte

Töö tulemusena loodi veebirakendus lihtsustamaks judo võistluste loomist, reklaamimist ning nendele registreerimist. Veebirakenduse peamisteks kasutajateks on klubi treenerid.

Klubi treenerid sisestavad rakenduse andmebaasi enda sportlased, mida edaspidi kasutama hakata. Sportlaste andmed taaskasutatakse võistlustele registreerimisel ning vastavalt vanusele genereeritud vanusekategorია välisvõtit uuendatakse igal pärimisel, mistõttu treeneri ainukeseks ülesandeks rakenduses jääb võistlustele registreerimine.

Rakenduse peamiseks eesmärgiks on mugav võistlejate andmete kogumine ning eksportimine võistlust korraldavasse programmi JudoShiai. Võistlusele registreeritute info kogutakse kokku ning kompileeritakse CSV failiks, mis pakutakse võistluse loojale allalaadimiseks. Pakutud lahendus lihtsustab ning kiirendab võistluse loomisega seotud tegevusi ning lisaks pakub töökindlust, vähendab võimalikke vigu.

Valminud rakendus pakub samuti lahendust toimunud võistluste tulemuste kogust. Igale võistlusele vastavad ühed tulemused, millele saab ligi tulemuste lehelt.

Tulevikuplaanid hõlmab endas rakenduse hajutamist erinevatele teenustele, lüües lahku kasutajaliidese ning teenuse tagarakenduse. Edasiarendus toimub samuti ka erinevatele spordialadele laienemisega ning lisateenuste loomisega, mis abistavad korraldada võistluse toimumist.

Veebirakendust on rakendatud 2020 kevadel toimunud ülemaailmse judo võistlusega „Tallinn Judo Cup 2020” kus osales 655 inimest.

Kasutatud kirjandus

- [1] “Sportdata” [Võrgumaterjal]. Available: <https://www.sportdata.org/>
- [2] “Sportists” [Võrgumaterjal]. Available: <http://sportists.info/judo/login.php>
- [3] “Bootstrap” [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Bootstrap> [Kasutatud 9. Aprill 2020]
- [4] “Ajax” [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Ajax> [Kasutatud 10.aprill 2020]
- [5] “Django template language” [Võrgumaterjal]. Available: <https://docs.djangoproject.com/en/3.0/ref/templates/language/>
- [6] “Django” [Võrgumaterjal]. Available: <https://www.djangoproject.com>
- [7] “Django models” [Võrgumaterjal]. Available: <https://docs.djangoproject.com/en/3.0/topics/db/models/>
- [8] “Stripe” [Võrgumaterjal]. Available: <https://stripe.com/docs/js>
- [9] “The Django Admin Site” [Võrgumaterjal]. Available: <https://docs.djangoproject.com/en/3.0/ref/contrib/admin/>
- [10] “Django settings” [Võrgumaterjal]. Available: <https://docs.djangoproject.com/en/3.0/topics/settings/>
- [11] “Django views” [Võrgumaterjal]. Available: <https://docs.djangoproject.com/en/3.0/topics/http/views/>
- [12] “Django authentication” [Võrgumaterjal]. Available: <https://docs.djangoproject.com/en/3.0/topics/auth/>
- [13] “JudoShiai” [Võrgumaterjal]. Available: https://judoshiai.fi/index-en.php?sub=software_en.php

Lisa 1 – Võistluse andmebaasi mudel

```
class Competition(models.Model):
    is_closed_competition = models.BooleanField(default=False)
    is_finished = models.BooleanField(default=False)
    registration_is_finished = models.BooleanField(default=False)
    is_published = models.BooleanField(default=False)
    name = models.CharField(max_length=255)
    desc = models.TextField()
    contact_email = models.EmailField()
    location = models.CharField(max_length=256)
    cost_over_internet = models.IntegerField()
    cost_at_the_competition = models.IntegerField()
    competition_master = models.ForeignKey("CustomUser",
        on_delete=models.SET_NULL, null=True, blank=True)
    club = models.ForeignKey(Club, on_delete=models.SET_NULL,
        blank=True, null=True)
    happening = models.DateField(blank=True, null=True, default='')
    happening_time = models.CharField(max_length=10)
    registration_end = models.DateField(blank=True, null=True)
    competition_image = models.ImageField(upload_to="images/",
        null=True, blank=True)
    accreditation_location = models.CharField(max_length=100)
    slug = models.SlugField(unique=True, null=False, max_length=255)
```

Joonis 7. Võistluse mudeli näide django ORM põhjal