

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Renata Muru 184947IADB

# **Veebirakendus Leia eraisiku ürituste reklaamimiseks ja otsimiseks**

Bakalaureusetöö

Juhendaja: Meelis Antoi  
Magistrikraad

Tallinn 2021

## **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: Renata Muru

15.05.2021

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärgiks on luua prototüüp veebirakendusest Leia, mille saaks edasi arendada vähimaks elujõuliseks tooteks. Veebirakendus on mõeldud aitamaks inimestel leida grupitegevusteks kaaslaseid ning teiselt poolt pakkuda võimalust endale tegevust leida.

Töö käigus luuakse piiratud funktsionaalsusega veebirakendus, mis on kohandatud mobiilil kasutamiseks. See rakenduse võimaldab registreerida, sisse-välja logida, teiste üritusi vaadata, neid juurde luua.

Bakalaureusetöö rõhuasetus on ühelt poolt nõuete kogumisel, et prototüüp edasi arendada rakenduseks, mida inimesed päriselt kasutada tahaksid. Teiselt poolt on olulisel kohal tänapäevaste, kiiret arendust ja laienemist võimaldavate tehnoloogiate ning raamistike valik. Töö tulemusena valmib prototüüp, mis verifitseerib valitud tehnoloogiate õigsust.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 44 leheküljel, 8 peatükki, 11 joonist, 2 tabelit.

## **Abstract**

### **Web Application Leia for Advertising and Finding Events Organized by Private Persons**

The goal of this bachelor thesis is to create a prototype of web application Leia which would allow for future development into a minimum viable product. The web app is to help people find mates for group activities and offer possibilities to find activities for oneself.

As part of this thesis, a web application with limited scope, adjusted for use on mobile phones, will be created. The application will allow users to register, log in and out, view events and create events.

On one hand, the emphasis of this thesis is on gathering requirements, so the prototype could be developed into a product people would really want to use. On the other hand, the emphasis is on choosing modern technologies and frameworks, which would allow for rapid development and easy changes in the future. The result of the development process is a prototype that verifies the suitability of chosen technologies.

The thesis is in Estonian and contains 44 pages of text, 8 chapters, 11 figures, 2 tables.

## Lühendite ja mõistete sõnastik

|         |   |
|---------|---|
| ABP     | ASP.NET Boilerplate   |
| API     | <i>Application Programming Interface</i> , rakendusliides               |
| CRUD    | <i>Create, read, update, delete</i> , - loo, loe, uuenda, kustuta       |
| CSS     | <i>Cascading Style Sheets</i> , märgistuskeel                           |
| DB      | <i>Database</i> , andmebaas   |
| DTO     | <i>Data Transfer Object</i> , andmevahetusobjekt                        |
| EF      | <i>Entity Framework</i> , olemiraamistik                                |
| ERD     | <i>Entity Relationship Diagram</i> , olemi-suhte diagramm               |
| HTML    | <i>Hypertext Markup Language</i> , programmeerimiskeel                  |
| IT      | Infotehnoloogia   |
| JS      | JavaScript, programmeerimiskeel   |
| JSON    | <i>JavaScript Object Notation</i> , andmevahetusformaad                 |
| MVC     | <i>Model View Controller</i> , mudeli vaate kontrolleri                 |
| MVP     | <i>Minimum Viable Product</i> , vähim elujõuline toode                  |
| PHP     | <i>Hypertext Pre-processor</i> , programmeerimiskeel                    |
| REST    | <i>Representational State Transfer</i> , API arhitektuuristiil          |
| SPA     | <i>Single Page Application</i> , üheleherakendus                        |
| SQL     | <i>Structured Query Language</i> , struktuurpäringukeel                 |
| Swagger | Projekt, mis kirjeldab ja dokumenteerib REST API-sid                    |
| Tenant  | Kasutajate grupp, kes jagavad rakenduse kindlale eksemplarile ligipääsu |

## Sisukord

|   |    |
|---|----|
| 1 Sissejuhatus .....                                      | 10 |
| 1.1 Metoodika.....  | 11 |
| 2 Probleemi kirjeldus.....                                | 12 |
| 2.1 Olemasolevad lahendused .....                         | 12 |
| 2.1.1 Facebook.....                                       | 12 |
| 2.1.2 Meetup.....   | 14 |
| 2.1.3 Teeme koos.....                                     | 14 |
| 3 Uus lahendus: Leia .....                                | 16 |
| 3.1 Küsitluse tulemused.....                              | 16 |
| 3.2 Nõuded.....   | 19 |
| 3.3 Tasuvuse analüüs.....                                 | 22 |
| 4 Tehnoloogia valik.....                                  | 24 |
| 4.1 Raamistikud.....                                      | 25 |
| 4.2 Andmebaasi valik .....                                | 29 |
| 5 Veebirakenduse disain.....                              | 31 |
| 5.1 Kasutajakogemus.....                                  | 31 |
| 5.2 Andmebaasi disain.....                                | 33 |
| 6 Veebirakenduse arendus .....                            | 36 |
| 6.1 Veebiteenuse arendus .....                            | 36 |
| 6.1.1 ABP projekti struktuur ning veel mõned eelised..... | 37 |
| 6.1.2 Turvalisus .....                                    | 38 |
| 6.2 Klientrakenduse arendus.....                          | 39 |
| 6.3 Testimine .....                                       | 41 |
| 7 Hinnang tehtud rakendusele .....                        | 42 |
| 7.1 Uudsus .....  | 42 |
| 7.2 Võimalused edasiarenduseks .....                      | 42 |
| 8 Kokkuvõte .....   | 44 |
| Kasutatud kirjandus .....                                 | 45 |

|  |    |
|--|----|
| Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks ..... | 47 |
| Lisa 2 – Veebiteenuse API-d JSON kujul .....   | 48 |

## Jooniste loetelu

|   |    |
|---|----|
| Joonis 1. Küsimustikule vastanute vanuseline jaotus.....  | 16 |
| Joonis 2 Küsitluses osalenute hinnang välja pakutud ürituste loomise omaduste olulisusele. .... | 18 |
| Joonis 3. Huvi erinevate tegevuskategooriate vastu.....   | 19 |
| Joonis 4. Armastatuimad veebitehnoloogiad 2020. aastal kogutud andmete põhjal. ....             | 27 |
| Joonis 5. Sisse logimise leht ning lihtsasti suletav mobiilimenüü infolehe kohal.....           | 32 |
| Joonis 6. Ürituste vaade.....   | 33 |
| Joonis 7. Leia ERD skeem vähendatud kujul. ....   | 34 |
| Joonis 8. ABP projekti struktuur. ....  | 37 |
| Joonis 9. Lihtne CRUD teenus. ....  | 38 |
| Joonis 10. Identiteediga seotud paketid projektis.....  | 39 |
| Joonis 11. Lihtne kate kasutaja andmete salvestamiseks luba küsimiseks.....                     | 40 |



## **Tabelite loetelu**

|   |    |
|---|----|
| Tabel 1. Serveri kirjutamiseks kasutatavad populaarsemad keeled ja seotud raamistikud.<br>..... | 27 |
| Tabel 2 ASP.NET Boilerplate kasutamise plussid ja miinused.....                                 | 29 |

## 1 Sissejuhatus

Arenenud ühiskondades räägitakse järjest enam üksildusest ja sotsiaalsest eraldatusest kui tõsisest tervist ohustavast probleemist [1]. Sotsiaalmeedia suurenevat kasutust kahtlustakse üksildustunde suurendamises [2], mitte selle parandamises. Sotsiaalsed tegevused ja grupid on inimesele suures plaanis olulised, ei saa loota, et ainult virtuaalsetes keskkondades toimuvad suhtlused täidaks ära inimese vajadused.

Käesoleva lõputöö eesmärgiks on luua piiratud funktsionaalsusega veebirakenduse prototüüp, mis aitaks endale tegevust otsivaid inimesi kokku tuua. Isegi inimestel, kes üldiselt ei tunne end üksildasena ning kellel on tugev sotsiaalne võrgustik, ei ole alati kõigiks soovitud tegevusteks sobivat seltskonda – mõne hobi või huvi teostamiseks on vaja rohkem kui ühte-kahte inimest. Näiteks ei ole mõistlik üksi keerulisel maastikul matkata, võrkpalli ja teisi võistkonnamänge saab nautida keskmise suurusega grupis ning paljud arvuti- ja konsoolimängude mängijad otsivad punkti, kellega erinevaid kampaaniaid koos läbida.

Lahenduse on inspireeritud igavlevast, endale tegevust otsivast eestlasest. Maailmas leidub erinevaid platvorme enda ürituste korraldamiseks – kõige populaarsem on Eestis tõenäoliselt Facebook, kuid see on küllaltki kohmakas väiksemate ürituste korraldamiseks, eriti kui on soov oma sõpruskonnast kaugemale vaadata. Tuntust on kogunud ka Meetup, kuid see on jäänud pigem professionaalseks kasutuseks. Uuemate platvormide seas on huvitav Teeme koos, mis pakub just tegevuste kuulutamise võimalust, kuid keskendub ainult ühe kaaslaste leidmisele ja meenutab seetõttu kohtingusaiti (sellisele suunale viitab ka asjaolu, et rakendus on mõeldud ainult täisealistele).

Just grupi- ja seltskonnategevuste leidmiseks ja kuulutamiseks pole väga head mugavat võimalust. Veebirakendusega Leia proovitakse eelmainitud probleeme lahendada.

## 1.1 Metoodika

Käesoleva lõputöö esimeseks suuremaks sammuks on analüüsida probleemi, kirjeldada olemasolevaid lahendusi ning pakkuda välja omapoolne parem lahendus. Planeeritava veebirakenduse nõuete kogumisse on kaasatud ka potentsiaalsed tulevased kasutajad – selleks koostati Google'i vormi küsimustik ning levitati seda sotsiaalmeedia kanalites.

Küsimustiku tulemuste põhjal uuritakse võimalike kasutajate vanusegruppide vahelisi eelistusi ning tuuakse välja kõige olulisemad nõuded. Saadud tulemusi arvesse võttes prioriseeritakse planeeritavad omadused ning luuakse prototüübi skoop. Samuti hinnatakse uue rakenduse majanduslikku tasuvust.

Järgmiseks oluliseks sammuks on võimalike tehniliste lahenduste analüüs ning tööks sobiva lahenduse valimine. Otsitakse tehnoloogiapinu, mis oleks vastavuses autori oskustega ning võimaldaks prototüüpi edasi arendada esialgu vähimaks elujõuliseks tooteks, kuid ei piiraks tulevikus toote laienemist teistesse riikidesse. Seejärel disainitakse kasutajakogemust, välimuse disain jääb pigem järgmisesse etappi – lahenduse arendamisse.

Viimaks hinnatakse valminud prototüübi vastavust soovitud nõuetele, testitakse seda nii kasutaja kui arendaja aspektist ning antakse hinnang ka tehnoloogiapinu sobivusele prototüübi laiendamiseks.

## **2 Probleemi kirjeldus**

Maailma vallutanud pandeemia valguses räägitakse järjest rohkem ka inimeste sotsiaalsetest vajadustest. Pandeemiad ja nendega seotud piirangud tavapärasele elutegevusele suurendavad stressi, ärevust ja isoleeritustunnet [3]. Peagi on aga oodata olukorra olulist paranemist, nii et ühistegevusi saab jälle ilma piiranguteta ette võtta.

Suuremal osal inimestest on olnud elus perioode, kus endale meeldivate tegevuste tegemiseks on raske seltskonda kokku saada või on lihtsalt olnud igav ja pole leidnud endale mõistlikku tegevust. Antud lõputöö raames koguti tavalistelt inimestelt Google'i vormide kaudu tagasisidet planeeritava rakenduse kohta. Küsimusele, kas tuleb ette, et õhtuks ei leia endale tegevust, vastas 75% rohkem kui kaks palli (skaala 1 = mitte kunagi ning 5 = tihti).

Eestis kasutusel olevatel lahendustel on hulganisti probleeme. Järgnevalt vaadeldakse mõningaid eksisteerivaid platvorme lähemalt.

### **2.1 Olemasolevad lahendused**

Nii Eestis kui mujal kasutatakse ürituste korraldamiseks palju Facebooki. Samuti on töö autoril mitmeid kogemusi ürituste loomise ja haldamisega selles keskkonnas. Lisaks on Eestis võrdlemisi populaarne Meetup platvorm. Lõputöö raames korraldatud küsitluse käigus mainiti, et Leia-le väga sarnane tundub Teeme koos keskkond. Neid kolme saiti kohe lähemalt vaadeldaksegi.

#### **2.1.1 Facebook**

Facebook on nii kuulus sotsiaalmeediaplattform, et ei vaja enam tutvustamist. Ka võrdlemisi kõrge eani on pigem normiks, et Facebooki konto on olemas kui see, et seda ei ole. Küll aga on Facebookis ürituste loomine ja haldamine küllalt ebamugav. Selgelt on viimase aja suund olnud müügitegevuse konkurentidelt üle võtmine Facebook Marketplace näol ning ürituste korraldamine on üsna nõrgalt arendatud toode.

#### Plussid:

- väga levinud, palju kasutajaid;
- palju kategooriafiltreid;
- ürituste soovitusel algoritmi poolt;
- igalt kontolt saab üritusi luua (nii eraisik kui ettevõtte);
- üritusi saab samal platvormil reklaamida;
- saab korraldada üritusi ainult oma sõpradele;
- saab korraldada ka täiesti avalikke üritusi.

#### Miinused:

- mobiiliversioonis ei ole kõiki funktsioone;
- palju infomüra, üritused ei ole platvormil eriti olulised;
- üritustele reageerimised ei tähenda suurt midagi – selle väärtus on devalveerunud;
- privaatset üritust ei saa sõprade ringist väljapoole jagada;
- privaatset üritust ei ole võimalik enam avalikuks teha;
- avalik üritus on nähtav kõigile soovijatele;
- filtrid ei tööta hästi – näiteks ei õnnestu autoril leida spordiüritusi vastava kategooria märkimisel, tulemusteks olid ainult veebipõhised komöödiaõhtud;
- huvitavaid üritusi ei kuvata ilma neid ise otsima minemata.

### **2.1.2 Meetup**

Meetup on platvorm kogukondade leidmiseks ja loomiseks. Eestis on see peamiselt kasutusele võetud infotehnoloogiaga seotud gruppide poolt. Tihti on grupid seotud mõne ettevõtte või organisatsiooniga.

Plussid:

- saab liituda huvi pakkuvate gruppidega;
- saab luua uusi gruppe;
- saab vabalt otsida üritusi ning gruppe;
- saab lihtsalt üritusi luua;
- keskendunud enesearengule ja kontaktide loomisele – palju harivaid üritusi.

Miinused:

- sihtgrupp on pigem infotehnoloogiaga seotud inimesed;
- kasutatakse palju enda ettevõtte ja isiku reklaamimiseks;
- pigem korraldatakse karjääri ja tööga seotud üritusi.

### **2.1.3 Teeme koos**

Teemekoos.ee on professionaalselt teostatud veebirakendus kaaslaste leidmiseks. See on koondunud tegevuste ümber – saab valida endale meeldivad hobid ning otsida samade huvidega seltsilist. Samuti pakub leht soovitusi ettevõtetest, mille kaudu antud tegevusi sooritada. Kuigi keskkonna tutvustus seda otseselt ei väljenda, viitab kasutajakogemus sellele, et tegemist on pigem kohtingute kokku leppimise platvormiga. Lõputöö autoriga pärast konto loomist ühendust võtnud meesterahvas ütles nii: „Ma ei ütleks, et see kohtingu sait on, kuid seda saab sedasi tõlgendada.“

Plussid:

- võimalik otsida oma hobide sooritamiseks kaaslast;

- pakub soovitusi, kus antud tegevusi teha saab;
- kasutaja saab valida, mis vanuses ja mis soost inimestega kontakt luua;
- palju erinevaid valdkondi, mille vahel valida;
- kui kaaslase otsija ja vahel on *match* (platvormil kasutatud sõna selle kohta, kui kasutajatel on vähemalt üks sarnane huvi), saab alustada vestlust.

Miinused:

- rakendus on mõeldud ainult täisealiste;
- tegevusi ei kuulutata, kaaslase otsija saab oma otsinguparameetritega sobiva nimekirja inimestest, kellega vestlust alustada;
- vestlust alustatakse privaatset juba ühiste huvide alusel, seda ka ilma planeeritava ürituseta;
- mitmed eelmainitud punktid viitavad, et tegemist on kohtingusaidiga.

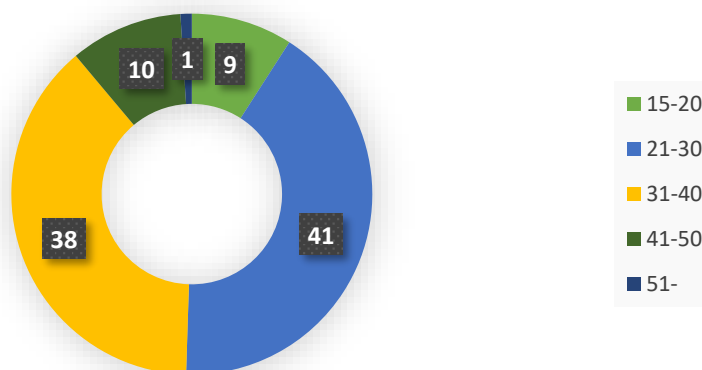
### 3 Uus lahendus: Leia

Lõputöö eesmärgiks on luua rakendus, mille abil oleks lihtne endale tegevust ja seltskonda leida – juba samal päeval. Üheks põhjuseks on töö autori halvad kogemused Facebooki ürituste korraldamise ja sealt info otsimisega. Kuigi ka uues rakenduses on inimestel vaja omavahel kontakt luua, siis peaks põhirõhk olema mitte sotsiaalsel suhtlusel, vaid just üritustel endil. Esialgsete ideede valideerimiseks viidi läbi küsitlus, millele vastas 99 inimest. Küsitluse tulemusi võeti arvesse kasutajalugude koostamisel ja prioriseerimisel.

#### 3.1 Küsitluse tulemused

Küsitlus viidi läbi anonüümselt ning vastajate endi kohta ei küsitud rohkem andmeid kui sugu ning vanust. Üks vastanutest ei soovinud sugu avaldada, mehi oli vastajate seas 34 ning naisi 64. Küsimustikus olid vanusegrupid seatud järgnevalt: 15-20, 21-30, 31-40, 41-50 ja 51-. Põhiosa vastajatest moodustasid inimesed vanuses 21-40: 41 vanuses 21-30 ning 38 vanuses 31-40 (vt Joonis 1), seega 79,8% vastajatest olid vanuses 21-40.

#### Vastanute vanuseline jaotus



Joonis 1. Küsimustikule vastanute vanuseline jaotus.



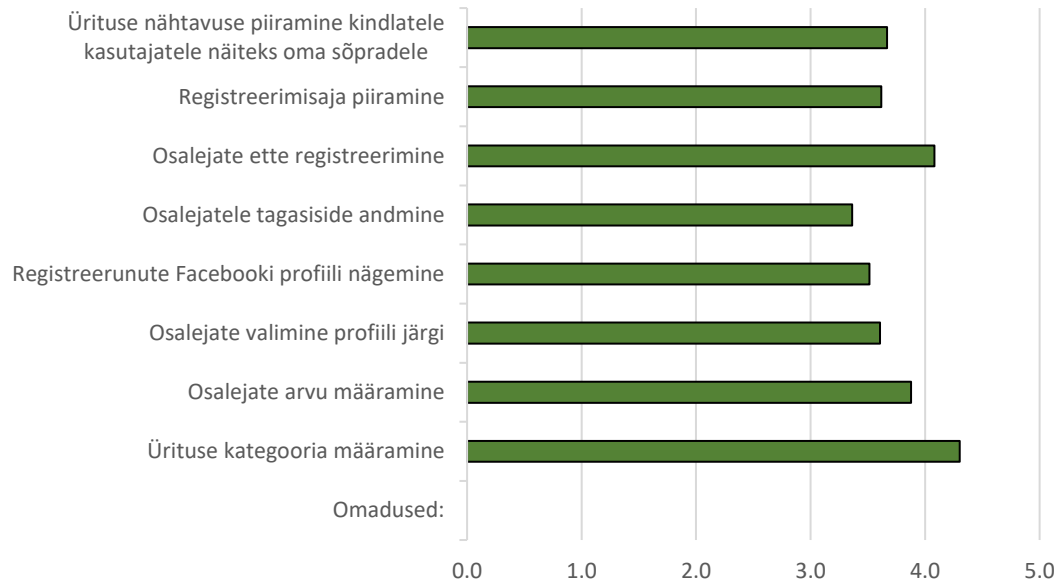
Küsimusele „Kas tuleb ette, et õhtuks ei leia endale huvitavat tegevust?“ vastas „mitte kunagi“ 9 inimest. Viie palli skaalal „mitte kunagi“ kuni „tihti“, kus viimane oli võrdne viie palliga, vastas „tihti“ 13 inimest. 23 valis sellel skaalal neli palli ning 36 kolm palli. Seega on õigustatud varasem oletus, et inimestel on tõesti mõnikord igav.

Järgmisele küsimusele „Kas juhtub, et on mõte, mida väga tahaks teha, kuid sõpruskonnas pole piisavalt huvitatuid?“ vastas „tihti“ 26 inimest, nelja palliga hindas olukorda 27 inimest ning ainult kahel juhul ei juhtu mitte kunagi, et oma sõpruskonnast jääb väheks. Seega on õigustatud oletus, et leidub inimesi, kes tahaksid ise mingisuguse ürituse korraldada. Veelgi enam, vastanutest 17 oleks valmis endale meeldivaid ettevõtmisi „kindlasti“ ka koos võõraste inimestega tegema. Samale küsimusele vastas 33 inimest nelja palliga ning 30 inimest kolme palliga. Pigem jaatamisele kaldusid seega pea 50% vastanutest.

Edasi uuriti, kui paljud oleksid valmis ise oma häid mõtteid kuulutama ja endale tundmatuid inimesi kaasama. Siin olid vastajad tagasihoidlikumad. Ainult üheksa vastajat olid kindlasti seda valmis tegema, nelja palliga hindas selle võimalikust 25 inimest. Siiski on see üsna lootustandev, kuid pikemas perspektiivis tuleb arvestada, et tõenäoliselt kõikidele kasutajatele ei sobi võõrastega läbi käimine. Üllatavalt ei olnud siin märgata väga tugevat vanuselist trendi – või oli trendi kujunemiseks vastajaid liiga vähe. Kui välja jätta kõige vanemate inimeste grupp, siis oli keskmine vastus vahemikus 2,4 (41-50 aastased) kuni 3 (31-40 aastased). 15-20 aastaste grupis oli keskmine 2,78 ja 21-30 aastaste grupis 2,85.

Vastajatelt küsiti tagasisidet rakenduse omaduste olulisuse kohta ürituse loomisel (vt Joonis 2). Väga oluliseks peeti ürituse kategooria määramist (keskmiselt 4,30 palli) ja osalejate ette registreerimist (4,08 palli). Osalejate arvu piiramine sai keskmiselt 3,88 palli, ürituse nähtavuse piiramine kindlale grupile 3,67 palli, osalejate nende profiili järgi valimine ning registreerimisaja piiramine olid üsna võrdsed – vastavalt 3,61 ja 3,62 palli. Selgelt kõige vähem pakkus huvi ürituse kutse saatmine sõprade e-posti aadressile (3,06 palli). Osalejate Facebooki profiili nägemine ning neile tagasiside andmine pärast üritust olid samuti üsna vähe populaarsed valikud (3,52 ja 3,36 palli). Üldiselt on kõige olulisem üritus korralikult kategoriseerida, nii et inimesed selle ülesse leiaksid ning lisaks tahetakse, et osalejad kindlasti ette registreeruks. Mõningane huvi oli ka selle vastu, et kuulutus päris kõigile avalik poleks ja taheti teada, kes on registreerunud inimesed.

## Erinevate omaduste keskmistatud olulisus

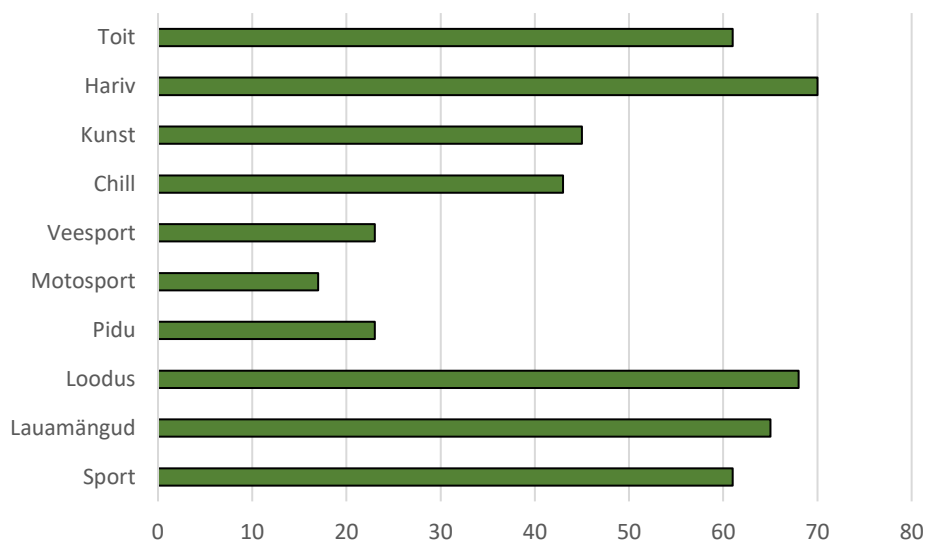


Joonis 2 Küsitluses osalenute hinnang välja pakutud ürituste loomise omaduste olulisusele.

Autori hinnangul võiks proovida ka kasutajate reitingu-süsteemi kasutamist, sest eesmärgiks ei ole olla järgmine sotsiaalmeediaplattform. Seega võiks nii korraldajad kui üritusel osalejad jätta hinnangu – näiteks huviline, kes pärast registreerumist üritusele kohale ei ilmu või kes kohatult käitub, saaks halvema hinnangu ning peagi teda üritustele enam ei oodataks.

Lisaks uuriti küsitluses, missugused ürituste kategooriad võiksid kasutajatele huvi pakkuda (vt Joonis 3). Valikusse pandi üsna laiad kategooriad, et küsitluse täitmine muu hulgas liiga kurnav poleks. Suurim oli huvi harivate ürituste vastu, mida märgiti 70 korral. Looduse kategooria järgnes 68 häälega, lauamängud said 65 huvilist, sport ja toit said mõlemad 61 märkimist. Ilmselt oleks mõistlik ka edaspidi kasutajatelt küsida, mis kategooriaid nad rakenduses näha tahaks.

## Huvi erinevate kategooriate vastu



Joonis 3. Huvi erinevate tegevuskategooriate vastu.

Küsitlusega tutvudes ning seda täites said vastajad juba mingi ettekujutuse, mis rakendust plaanitakse luua. Seetõttu esitati neile ka küsimus, kas seda rakendust oleks nende ellu vaja – sellele vastas jaatavalt lausa 68 inimest. Seega on igati põhjendatud ära proovida, kas see rakendus päriselt ka kasutusele võetaks ja edu saavutaks.

### 3.2 Nõuded

Nõuete sorteerimiseks kasutati MoSCoW meetodit [4]. Eesmärgiks on luua prototüüp, mida peagi edasi arendada vähimaks elujõuliseks tooteks, seega *could have* (võiks olla ing. k) ja *won't have* (ei tee ing. k) osad jäävad tulevikus tehtavaks. Esimesena arendatakse *must have* (peab olema ing. k) ja seejärel *should have* (peaks olema ing. k) nõuded, need kaks kategooriat peaksid kokku andma vähima elujõulise toote, mida juba lõppkasutajatele esitleda.

*Must have:*

- Kasutajana soovin luua konto, et platvormi kasutada.
- Kasutajana soovin sisse ja välja logida, et rakendus oleks seadmel kasutatav ainult mulle sobivatel aegadel.

- Kasutajana soovin lihtsalt leida mulle sobivaid üritusi, mille kohta on osalemiseks antud piisavalt infot, et teaksin, mis, kus ja millal toimumas on.
- Kasutajana soovin kiirelt luua üritusi, et kutsuda sinna ka teisi inimesi.
- Kasutajana soovin oma profiili muuta, et see alati peegeldaks õiget infot.
- Kasutajana soovin oma parooli muuta, kui olen selle unustanud, et platvormi edasi kasutada.
- Korraldajana tahan näha oma veel toimumata üritusi, et neid hallata.
- Administraatorina soovin teha andmetes manuaalseid muudatusi, et aidata kasutajaid platvormi puudujääkide esinemise korral.
- Kasutajana soovin, et minu andmeid hoitaks turvaliselt, et neid ei saaks koguda ja pahatahtlikult ära kasutada.

*Could have:*

- Kasutajana soovin ürituste veelgi kiiremaks leidmiseks kasutada kategooriafiltreid nagu „loodus, chill, sport“ jne, et mitte kulutada aega teemadele, mis mulle huvi ei paku.
- Korraldajana soovin oma üritusele lisada kategooriaid, et need teistele lihtsalt leitavad oleks.
- Kasutajana soovin näha, mis üritustele registreerunud olen, et neid mitte ära unustada.
- Kasutajana soovin üritustele registreeruda, et teada anda enda soovist osaleda.
- Korraldajana soovin näha, mitu inimest ja mis profiiliga inimesed on mu üritusele registreerunud, et teada, kas piisavalt sobivaid inimesi on tulemas.
- Kasutajana soovin rakendust eelkõige kasutada mobiilsel seadmel, et infole igal ajal mugavalt ligi pääseda.

- Korraldajana soovin enda üritusi tühistada, juhul kui mingil põhjusel üritus toimuda ei saa.
- Kasutajana soovin ürituse korraldaja profiili vaadata, et saada kindlustunnet temaga kohtumise osas.
- Korraldajana soovin iseloomustada üritust, lisada toimumiskoha ning muu olulise info kuulutusse, et potentsiaalsed osalejad saaksid registreerumiseks piisavalt infot.
- Korraldajana soovin valida, millised registreerunud üritusele lubatakse, et kokku tuleks hea seltskond.

*Should have:*

- Osalejana soovin saada teavitust, et üritus on tühistatud, et mitte ilma asjata kohale minna.
- Korraldajana soovin blokeerida teatud kasutajaid oma kuulutuste nägemisest, et nad ei saaks neile registreeruda.
- Kasutajana soovin anda avalikku tagasisidet üritusest osa võtnutele, et head inimesed saaks märgatud ja kehvad osalejad enam üritustele oodatud poleks.
- Kasutajana soovin, et saaksin endale meeldivat ürituste otsingufiltrit salvestada, et hiljem ühe klikiga õige otsing sooritada.
- Korraldajana soovin oma kontaktinfot jagada ainult nende registreerunutega, kelle olen üritusele lubanud.
- Korraldajana soovin piirata registreerimiste arvu, et ei peaks liiga paljusid kandidaate üle vaatama.
- Korraldajana soovin oma üritusi jagada otselingina, et oma tuttavaid üritusele kutsuda.

*Won't have:*

- Kasutajana soovin, et saaksin näha enda ja teiste korraldatud ürituste ajalugu, et meenutada olnud kohtumisi ja hinnata nende aktiivsust.

### **3.3 Tasuvuse analüüs**

Katsetusperioodiks on rakenduse ligipääsetavuse tagamiseks loodud virtuaalmasin IP-ga 193.40.231.161 Tallinna Tehnikaülikooli IT Kolledži serveritesse, töö valmimise hetkel ei ole see veel kasutusel. Prototüübiga saab toimetada tasuta ja ka lokaalses masinas, seega on see hea valik esialgseks arenduseks.

Rakenduse jagamiseks kasutajatele tuleks osta domeen ja mõistlik oleks mõne teenusepakkuja lahendust kasutada. See kulu koos domeeniga on ligikaudu 180 EUR aastas, kui kasutajaid on võrdlemisi vähe (need kulud võivad kiiresti suureneda koos kasutajate arvu suurenemisega) [5]. Kindlasti peaks kulude katmiseks lisama vähemalt Google Ads-i ning mõistlik oleks pakkuda ka reklaamiteenust erinevatele teenusepakkujatele, kelle tegevus sobitub hästi veebirakenduse teemaga - need võiksid olla erinevad seiklusturismi ettevõtted, spordivahendite renti pakkuvad ettevõtted (SUP, jalgrattad, kanuud jms). Reklaamiteenuse pakkumiseks peaks koguma ka andmeid kasutajate ja sessioonide arvu kohta, aktiivsuse kohta, eriti peaks vaatama ka reklaamide efektiivsust, kui tegemist on näiteks klikitava lingiga. Reklaamiteenuse ostja peab teadma, mille eest ta maksab. Näiteks sobiks info kogumiseks Google Analytics. Kui päises oleva reklaami eest on üks ettevõtte nõus maksma 15 EUR kuus, on sellega juba esialgsed kulud kaetud.

Käesoleva lõputöö skooopi ei kuulu tasuliste teenuste lisamine rakendusele, kuid pikas perspektiivis on mõeldud ka sellele võimalusele. Väga populaarne kohtingusait Tinder teenib oma tasulistelt teenustelt hästi kasumit, kuid sellised arendused on mõistlikud vaid siis, kui kasutajate hulk on ületanud mingi kriitilise piiri – vähemalt antud töö raames, sest selle esmane eesmärk ei ole teenida kasumit. Tinderi tasuliste teenuste kasutajate hulk on aasta-aastalt tõusnud alates nende välja andmisest aastal 2015, 2020 viimases kvartalis oli Tinderis 6,4 miljonit tasulise teenuse kasutajat ning kasutajate koguarv on hinnatud 66 miljonile [6]. Ligikaudu 10 protsenti aktiivsetest Tinderi kasutajatest tarbivad tasulisi teenuseid. Kui veebirakendus Leia kogub soovitude teel vähemalt 10 000

igapäevast kasutajat, on mõistlik tõsisemalt mõtlema hakata ka tasulistele teenustele ja nende väärtusele kasutajale. Kuludesse ei ole arvestatud platvormi aktiivset tasulist reklaamimist, kuid tasuliste teenuste müümiseks on kindlasti vajalik ka läbimõeldud turundusstrateegia.

## 4 Tehnoloogia valik

Kui on teada, mida teha, siis järgmine suur küsimus on, kuidas teha. Tehnoloogiapinude valik on tänapäeval väga suur [7]. Järgnevalt tuuakse välja olulised parameetrid tehnoloogiapinu valimisel.

- Soovitakse ehitada veebirakendus (välditakse mobiilirakendust). Arvestades eesmärki ei ole mobiilirakendus õigustatud, sest sund tundmatu rakendus alla laadida võib esialgu kasutajate arvu kasvu pärssida. Samuti puudub soov teenida tulu rakenduse põhifunktsionaalsuste kasutamise pealt.
- Tehnoloogiapinu peaks järgima kui mitte parimaid, siis häid praktikaid.
- Lähenemine peaks olema paindlik ja lihtsasti täiendatav, eriti paindlik peaks olema kasutajale nähtav osa.
- Soov on valida tänapäeval populaarseid lahendusi, seda ka just kogemuse saamise tõttu.
- Üheks piiravaks faktoriks on arendaja varasemad kogemused – mõistlik ei ole alustada päris võõraste tehnoloogiatega, et vältida väikeste probleemide väga ajamahukat lahenduskäiku.
- Vähese ajaga peab jõudma kaugele, seetõttu on mõistlik kasutada olemasolevaid raamistikke.
- Tehnoloogiapinu peab olema hästi dokumenteeritud, veel parem, kui selle kasutamise kohta leidub tasuta tutvustavaid kursuseid või loenguid.



- Lõpptulemus peab olema turvaline, st andmed ei tohi olla vabalt kättesaadavad ning kindlasti ei tohiks eksisteerida klassikalisi vigu nagu andmebaasi avatult salvestatud salasõnad või SQL<sup>1</sup> süstimisnõrkused.
- Valitud raamistikud ei tohi nõuda finantseerimist, sest autor teeb tööd iseseisvalt ning see ei ole seotud ühegi veel eksisteeriva ettevõttega, mis rahaliselt tööd toetaks.

Mõningate eeltoodud punktide põhjal osutus valituks klient-server arhitektuur [8]. See on laialdaselt kasutatud tavapärane praktika, mida ka Tallinna Tehnikaülikoolis põhjalikult õpetatakse. Klient-server arhitektuuri olulisteks eelisteks loetakse ka andmete tsentraliseeritud haldamist ja paindlikkuse lubamist kliendi poolel. Samas on erinevate blogide ja artiklite autorid eri seisukohtadel, kas seda arhitektuuri on kallim üleval pidada kui teisi lahendusi ning kas andmeid on parem kaitsta kui need paiknevad tsentraalselt või hajutatult [9] [10]. Tõenäoliselt sõltub eelistus nii postituse tegemise hetkest, postitaja isiklikest eelistustest ja kogemustest ning ka parasjagu kättesaadavast võrdlusmaterjalist.

## 4.1 Raamistikud

Nagu eelpool mainitud, on mõistlik võtta aluseks korralik raamistik, mis vähendab prototüübi valmimise aega. Kui õppimisprotsessis on oluline alustada täiesti baasteadmistest ja palju asju käsitsi teha, siis konkurentsirohkes turumajanduses on turule jõudmise aeg tihti väga oluline.

Valiku alustuseks on mõistlik vaadelda erinevaid keeli. Tabel 1 annab ülevaate mõningatest serveri poole programmeerimiskeeltest, seotud raamistikest, autori kogemusest ja kogukonna suurusest [11]. Viimane on oluline indikaator tehnoloogia elujõulisusest. Suur kogukond aitab üldiselt kaasa ka raamistike kvaliteedi tagamisele.

---

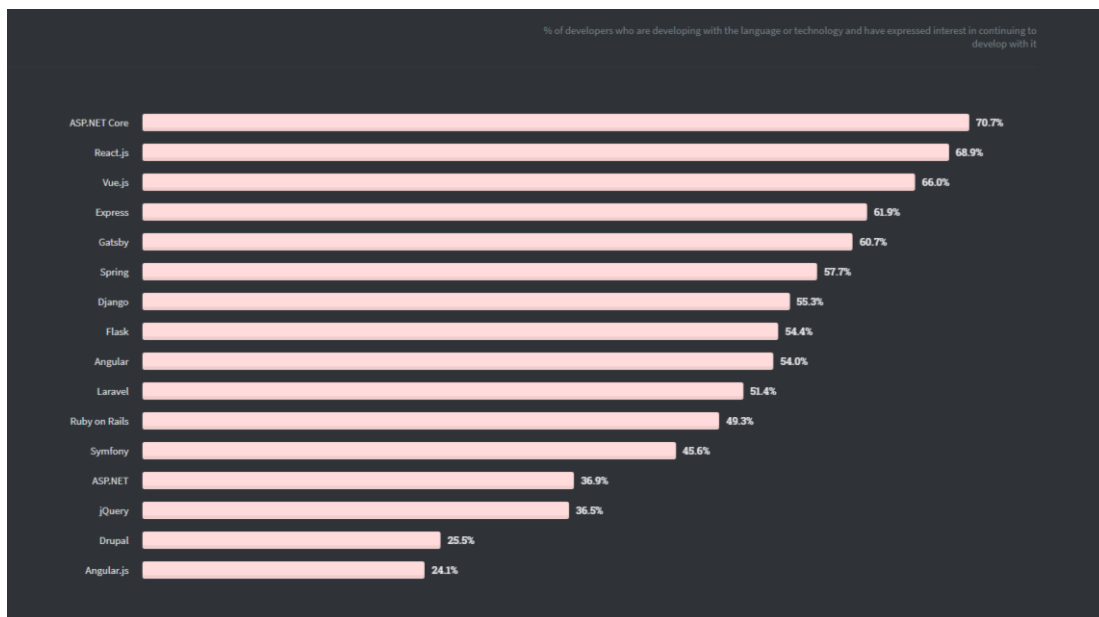
<sup>1</sup> *Structured Query Language* - struktuurpäringukeel

| Keel       | Mõni tuntud raamistik  | Autori kogemus  | Kogukond  |
|------------|------------------------|---|---|
| Python     | Django, Flask, FastAPI | Mõningane kogemus Pythoniga, raamistikke pole kasutanud.          | Django on väga populaarne ning reaalselt tehnoloogiahiidude nagu Instagrami poolt kasutuses. Flask on samuti võrdlemisi populaarne, FastAPI on samas uus raamistik ja seetõttu vähe kasutuses. [12] |
| PHP        | Symfony, Laravel       | Mõningane kogemus PHP-ga, raamistikke pole kasutanud.             | Laravel on siiani väga populaarne, Symfony on võrdlusena marginaalne [13].  |
| JavaScript | Node.js                | Mõningane kogemus JavaScriptiga, väga vähene kogemus Node.js-iga. | Node.js on väga populaarne ning sellele tugineb ka mitmed raamistiku edasiarendusi, nt Express.js. Raamistikku aitab edasi arendada suurte ettevõtete huvi tehnoloogia vastu. [14]                  |
| Go         | Gin, BeeGo             | Kogemus puudub.   | Juba keel ise on PYPL indeksi järgi 13. kohal ning seega võrdlemisi ebapopulaarne, kuigi seda veab Google [15].   |
| Java       | Spring                 | Mõningane kogemus.  | Üleüldiseltki populaarne, on see vaieldamatu favoriit Java raamistike seas [16].  |
| C#         | ASP.NET                | Nimetatutest kõige suurem kogemus nii C# kui ASP.NET-iga.         | Väga populaarse .NET raamistiku laiendus ning veebiarenduses populaarne [17].   |

| Keel | Mõni tuntud raamistik | Autori kogemus  | Kogukond   |
|------|-----------------------|-----------------|--|
| Ruby | Ruby on Rails         | Kogemus puudub. | Juba keel ise on PYPL indeksi järgi 16. kohal, seega pole ei keel ega raamistik väga populaarsed. [15] |

Tabel 1. Serveri kirjutamiseks kasutatavad populaarsemad keeled ja seotud raamistikud.

Antud kokkuvõtte põhjal on välistatakse Go ja Ruby liiga väikese kogukonna tõttu. Python, JavaScript, Java ning C#, eriti viimane, on kõik tugevad kandidaadid. Eelmisel aastal kogutud andmete põhjal armastavad arendajad kõige rohkem ASP.NET Core veebiraamistikku (vt allolevat pilti) [18].



Joonis 4. Armastatuimad veebitehnoloogiad 2020. aastal kogutud andmete põhjal.

Allikas: Kuvatõmmis Stack Overflow Insights lehelt.

Edasi on aeg vaadata ka kliendi poolt. Keelte valiku osas on asi siin palju lihtsam, sest tänapäeval klient-server arhitektuuris peale JavaScripti/Typescripti valikuid pea polegi. -

Kõikides raamistikes kasutatakse HTML-i<sup>1</sup>, CSS-i<sup>2</sup>, Bootstrapi, kombinatsiooni nendest või nende edasiarendusi. Töö autor on kliendi arendamisega oluliselt vähem kokku puutunud ning populaarsemad raamistikud, vähemalt Stack Overflow<sup>3</sup> statistikale tuginedes, nagu React.js, Angular, jQuery, Vue.js, on ühtmoodi vähe tuttavad [19].

Eelnevaid tingimusi arvestades osutus väga sobilikuks ASP.NET Boilerplate<sup>4</sup>, ühendades autorile sobiliku C# ASP.NET serveri projektipõhja ning pakkudes sinna juurde ka valikut klientide põhjadest [20]. Edasi vaadeldakse selle baasprojekti valimise plusse ja miinuseid.

| Plussid   | Miinused  |
|---|---|
| Palju olulisi korduvaid funktsionaalsusi on otsast lõpuni töökorras (nt sisse logimine, kasutajate ja rollide loomine). | Sellises suures raamistikus on palju osi, mida otseselt kohe (või ka hiljem) vaja pole. Näiteks <i>multi-tenancy</i> ehk üürniku mudel, mis võib hiljem mitmetesse riikidesse laienedes kasuks tulla. |
| Võimalik valida tänapäevases kliendiramistikus kirjutatud SPA ning seda täiendada.                                      | Suurde võõrasse koodibaasi keeruline alguses sisse elada, selle tundma õppimine võtab aega.   |
| Baasprojekt on väga hea dokumentatsiooniga ning parimaid praktikaid järgiv lahendus.                                    | Suure tõenäosusega ei saa autor veel mõnda aega kõikidest baasprojekti toimimise mehhanismidest aru, kuigi oskab seda enda tarbeks rakendada, millega omakorda kaasnevad riskid.                      |

<sup>1</sup> *Hypertext Markup Language*

<sup>2</sup> *Cascading Style Sheets*

<sup>3</sup> Stack Overflow on kõige populaarsem arendajate vestlusplatvorm küsimuste-vastuste vahetamiseks (autori märkus).

<sup>4</sup> Boilerplate – standardne projektipõhi (autori märkus).

| Plussid  | Miinused   |
|--|--|
| Kuna järgitakse parimaid praktikaid, on tagatud kõige olulisemate osade turvalisus nagu kasvõi kasutajate salasõnade hoiustamine varjatud kujul. | Antud baasprojekti kasutades peab töö autor ette tulevad probleemid ise lahendama, sest õppetöös kõiki raamistiku tehnoloogiaid ei kasutata. |
| Olemasolevast projektist saab õppida täiesti teisi lähenemisi, kui ülikoolis seni demonstreeritud on ning sellega teadmisi täiendada.            |  |
| Baasprojekti kasutamine aitab tõenäoliselt oluliselt kiiremini jõuda lihtsasti MVP-ks arendatava prototüübi valmimiseni.                         |  |

Tabel 2 ASP.NET Boilerplate kasutamise plussid ja miinused.

Plusse on baasprojekti kasutamisel rohkem kui miinuseid. Kliendi osas on valiku tegemine oluline samm. Et baasprojekti tutvustavad videod olid kasutanud just Angulari, on kiireks alguseks see raamistik parim valik. Prototüübi tasemel ei ole klientrakenduse raamistiku hilisem muutmine ka väga suur töö, kui sobivamaks peaks osutama näiteks väga suure kogukonnaga react.js.

## 4.2 Andmebaasi valik

Andmebaase grupeeritakse mitut moodi. Struktuuri järgi jaotatakse neid ajalooliselt näiteks võrkstruktuuriga ja puustruktuuriga andmebaasideks, viimasel juhul võib tabel olla seotud ainult ühe teise tabeliga üks-mitmeses seoses mitmese pooles, st olla teise tabeli alluv. Alluvaid võib ühel tabelil olla mitu. Tänapäevased tabelid on pea kõik võrkstruktuuriga, kus seoste loomine tabelite vahel ei ole piiratud. [21]

Andmebaase kategoriseeritakse erinevatel viisidel, samas on juba teada, et kasutatakse ASP.NET raamistikku, mis toetab peamiselt suurt hulka relatsioonilisi andmebaase ning Azure Cosmos DB NoSQL andmebaasi, mis on graafi tüüpi andmebaas [22]. Viimasega autoril kogemus puudub ning üldiselt on NoSQL kasutamine õigustatud väga spetsiifilistes situatsioonides, eriti, kui on vaja suurel hulgal andmeid töödelda. Relatsiooniline andmebaas tugineb tabelitele, milles on veerud ja read, kusjuures iga rida iseloomustab unikaalne võti. Veerud ise on semantilise tähendusega ja iseloomustavad väärtuste tüüpe. Kahe tabeli sidumiseks peab olema neis sama väärtuse ja semantikaga väli. [21]

Relatsioonilistest andmebaasidest toetab EF Core tasuta näiteks Microsoft SQL Serverit, SQLite'i, PostgreSQL, MariaDB. Et .NET on Microsofti toode, siis sobib sellega suurepäraselt ka Microsoft SQL Server. Microsoft SQL 2019 Express on tasuta toode, kuigi oluliste piirangutega. Nimelt on lubatud maksimaalne suurus tasuta versioonil 10 GB, piiratud on tuumade ja mälu kasutus ja puhvri suurus. [23] Prototüübi arendamisel ei ole eelmainitud piirangud väga olulised ning seega esialgu sobib Microsoft SQL 2019 Express tööks hästi.

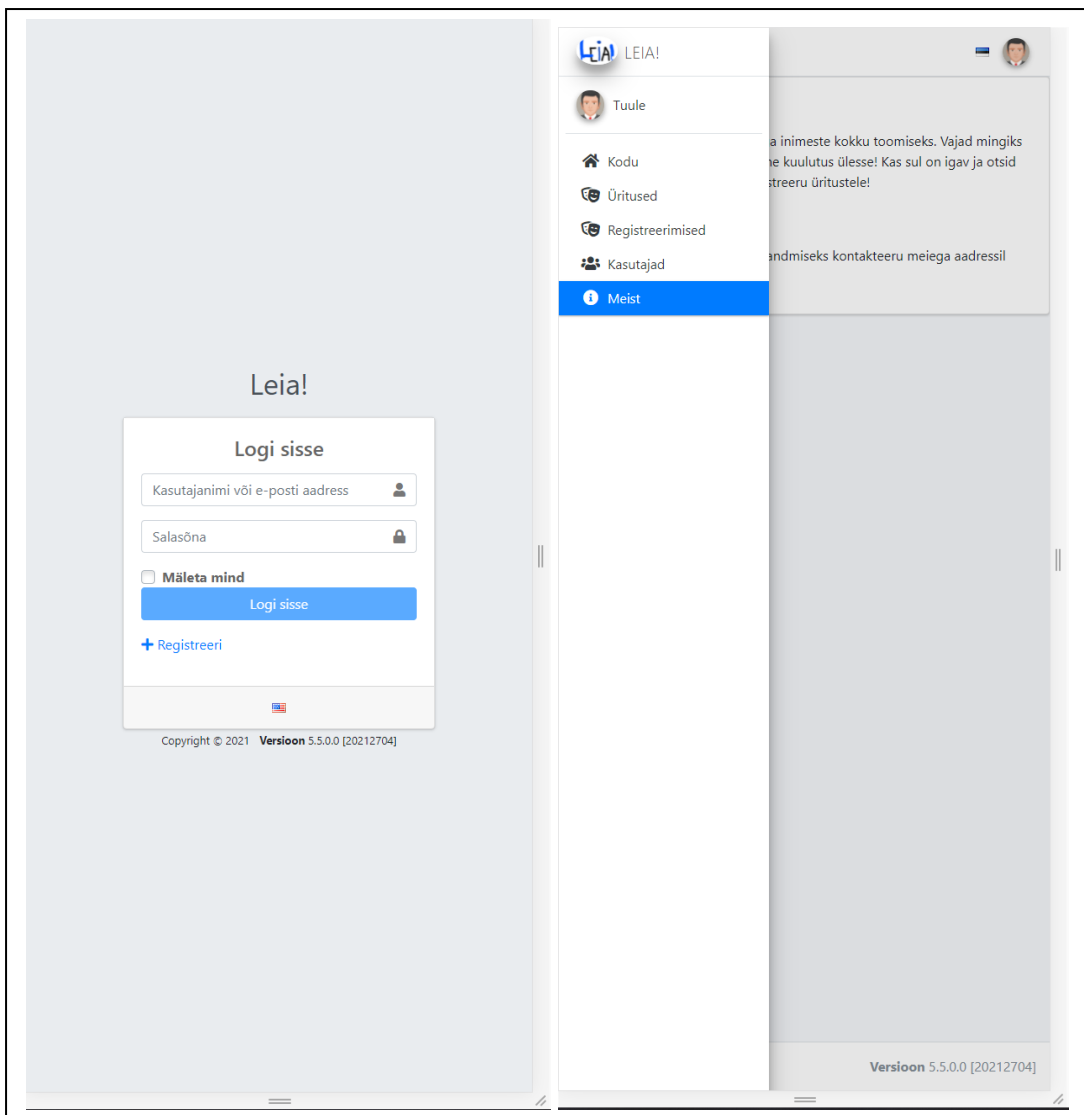
## 5 Veebirakenduse disain

Antud rakenduse disaini juures võeti aluseks teadmine, et üle 50% maailma veebilehtede külastustes tehakse mobiilidelt [24]. Üldiselt saab mobiilile disainitud rakendusi kasutada ka tavalise arvutiga, kuid vastupidi see loogika ei tööta. Hea praktika on kasutada reageerivat veebidisaini, nii et sait suudaks kohaneda erineva suurusega ekraanidega. Prototüübi välimus valiti arenduse käigus katsetades.

### 5.1 Kasutajakogemus

Kogu rakenduse süda on ürituste vaade ning otsing. Pärast sisse logimist peaks avanema just ürituste üleüldine vaade, kus saab neile registreerida, üritusi juurde luua ning üritusi filtreerida.

Et rakendus on eelkõige veebile suunatud, valiti lihtne ning helge, st heledates värvides välimus (vt Joonis 5). Väga väikeseid tekste, nuppe jm on mobiiliekraanil raske kasutada ja näha, seetõttu on nii mõnelgi pool võetud aluseks „mobiil esimesena“ (*mobile-first* inglise keeles) kasutajakogemuse loomise printsiip.



Joonis 5. Sisse logimise leht ning lihtsasti suletav mobiilimenüü infolehe kohal.

Töö autor disainis oma piiratud kunstimeelega lehele ka logo, mis sobitub sinine-must-valge-hall värvikavaga.

Ürituste vaade nõuab autori hinnangul veel tööd, sest mobiilivaates esialgu valitud tabel ei ole piisavalt lihtsasti loetav, kui infot natuke rohkem on (vt Joonis 6).



Üritused

Otsi...

| Nimi                   | Kirjeldus  | Asukoht  | Aeg                                  | Telefon    |
|------------------------|--|--|--------------------------------------|------------|
| Among Us               | Ma olen natuke liiga algaja, et oma sõpradega Among Us-i hästi mängida. Harjutame koos! 8+ inimest täiesti sobiv.          | Kohtume Discordis! <a href="https://discord.gg/U63f">https://discord.gg/U63f</a> | Tue Apr 27 2021<br>06:15:20 GMT+0300 | 5322342023 |
| Võrkpall Emajõe rannas | Emajõe rannas on liivaväljakud võrkpalli mängimiseks juba avatud. Ootan 3-5 inimest, kes sooviksid juba hooajaga alustada. | Tähtvere park, Emajõe rand, kolmas mänguväljak                                   | Tue Apr 27 2021<br>06:11:57 GMT+0300 | 32452321   |

Kokku: 2

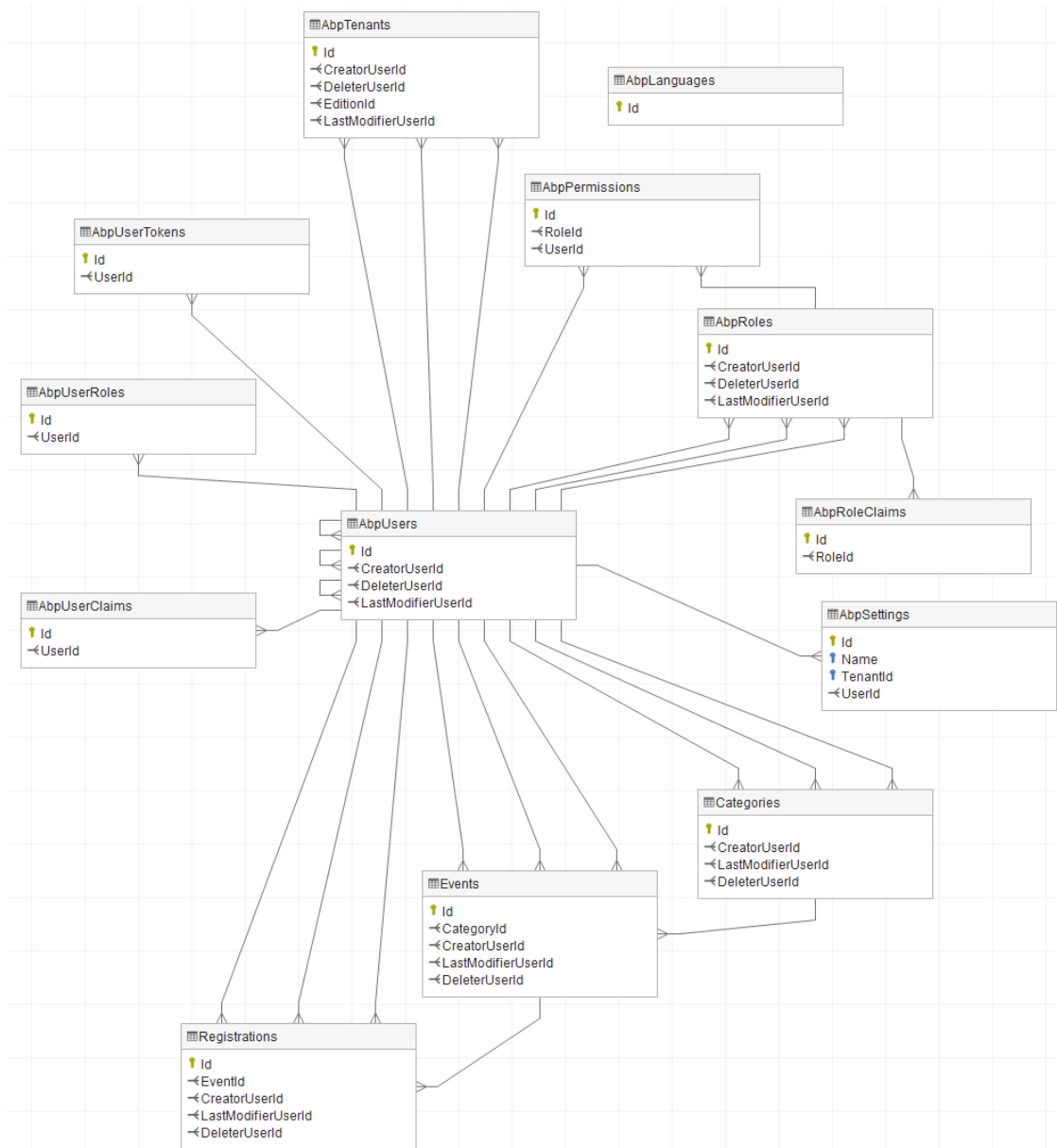
Copyright © 2021 leia. Versioon 5.5.0.0 [20212704]

Joonis 6. Ürituste vaade.

## 5.2 Andmebaasi disain

Andmebaasi disaini juures on oluline valitud raamistik. Selle võimekus on suurem kui antud töö raames tarvilik on. Näiteks on sisse ehitatud kasutajate sisselogimiskatsete kogumine, millega esialgu pole suurt midagi peale hakata. Küll ei tasu selliseid võimekusi kohe välja visata, sest peagi võib neid vaja minna ning prototüübi koostamist raamistiku suurem võimekus ei häiri. ERD skeemi loomiseks kasutati Dataedo tarkvara ning näitamiseks valiti just sisulist väärtust omavad olemid.

*AbpUser* tabelisse lisati mõned vajalikud veerud, lisaks loodi ürituste (*Events*), registreerimiste (*Registrations*), kategooriate (*Categories*) tabelid, millest viimane täideti ka algandmetega. Seda tabelit tavakasutajad täiendama ei peaks hakkama, küll on see võimekus administraatoritel.



Joonis 7. Leia ERD skeem vähendatud kujul.

Muudest tabelitest eraldiseisev on keelte (*Languages*) tabel, mille algandmetesse lisati ka eesti keel. Oodatult on keskseks tabeliks kasutaja tabel. ERD skeemil on näidatud ka *tenants* tabel, sest valitud raamistik on paljuski ehitatud ümber kontseptsiooni, et eri kasutajatele saab serverida mõneti isoleeritud süsteeme, mille kasutajad omavahel kokku

ei puutu. Näiteks võivad kliendipoolsed rakendusedki olla täiesti erinevad. Edaspidi võiks kaaluda selle kasutusele võtmist kasutajate riikide kaupa eristamiseks.

## 6 Veebirakenduse arendus

Tallinna Tehnikaülikool tagab tudengitele õppuri litsentsid, et kasutada parimaid programme koodi kirjutamiseks ja haldamiseks. Kliendi poole arenduses on kasutatud Visual Studio Code'i, C# keeles veebiteenuse arenduseks JetBrains Rider'it. Koodihalduseks oli kasutusel laialt levinud Git, samuti oli võtta kooli GitLab keskkond. Pikemas perspektiivis tuleks kindlasti migreerida keskkonda, mis ei ole tudengistaatusega seotud, kuid lõputöö raames on kooli poolt pakutu sobiv.

### 6.1 Veebiteenuse arendus

Veebiteenuse serveripoolne loogika, mis vastutab andmete ja kasutajate halduse, valideerimise jm eest, tugineb ASP.NET raamistikule. ASP.NET rakendusi saab käivitada ja arendada nii Windows, Linux kui MacOS operatsioonisüsteemides, kasutades selleks .NET platvormi. Lisaks C#-le saab ASP.NET rakendusi kirjutada ka F# või Visual Basicus. .NET koosneb erinevates tööriistadest, programmeerimiskeeltest ja teekidest, mille abil saab kirjutada erinevat tüüpi rakendusi. ASP.NET on .NET laiendus just veebirakenduste arendamiseks, näiteks on sinna juba lisatud autentimise süsteem erinevate teekidega, andmebaas ja põhiteenused sisse logimise haldamiseks. .NET rakendused jooksevad tavaliselt Dockeri konteineris, st Dockeri tugi on raamistikku lisatud. [25]

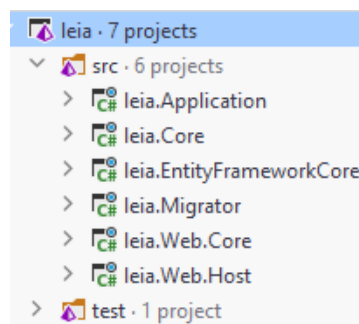
Tööks valitud raamistik on ASP.NET Boilerplate (ABP), mis on veel järgmine kiht ASP.NET Core ja Entity Framework Core peal. See pakub lisaks rakenduse raamistikule ka arhitektuurilist mudelit, mis järgib parimaid praktikaid ning tugineb domeenipõhise disaini printsiibile (*Domain Driven Design* inglise keeles) - see tähendab, et kood, k.a klassinimed ja muutujaid, meetodid, peaksid peegeldama ärivaldkonda ja rolle, milles seda tarkvara kasutatakse. [20]

Boilerplate baasprojekt on läinud sammu traditsioonilistest rakendustest kaugemale ning sisaldab endas ka SignalR-i. SignalR on vahevara, mis võimaldab tekitada sõlmpunktide (*hub*) API. Selle kaudu saavad klient ja server väga kiiresti suhelda, mis võimaldab sellele

raamistikule ehitada isegi mängu. SignalR-i kasutamisel defineeritakse serveri koodis meetodid, mida kutsub välja klient ning kliendi koodis defineeritakse meetodid, mida kutsub välja server. Võimalusel kasutab SignalR suhtluseks veebisoklit (*websocket* inglise keeles), kuid selle puudumisel muid variante. Selle veebirakenduse arendamise kontekstis polnud vaja otseselt mõelda SignalR kasutamise peale, kuid see võib väga oluliseks osutuda rakenduse avalikustamisel. [26]

### 6.1.1 ABP projekti struktuur ning veel mõned eelised

ABP projekti hoitakse päevakohasena ja seega polnud probleeme kõige uuema .NET versiooniga ühildumisega (5.0.202). Projekti põhistruktuuri näeb allolevalt pildilt.



Joonis 8. ABP projekti struktuur.

Projekti juurde kuulub ja MVC võimekus, siinse arenduse juures seda vaja polnud, seega keskendumine muule loogikale.

EntityFrameworkCore teek lubab lihtsalt olemi (*Entity*) objektidest luua migratsioonid, st skriptid, millega uuendada andmebaasi. Nendes on mugav ka lisada baasandmeid, mis uues andmebaasis alati olemas peavad olema. Siinsel juhul on heaks näiteks ürituste kategooriad, mida tavakasutajad ise muuta ei saa. Kõik baasklassid, mille pealt migratsioonid loodi, asuvad projektis leia.Core.

Loodud uued klassid pandi pärima nii *Entity<int>* kui *IFullAudited<User>* klassist. Esimene selleks, et automaatselt lisataks täisarv tüüpi identifikaator mudelisse, teine selleks, et lisanduks baasis hoitav ajalugu, mis muuhulgas lubab teha loogilist kustutamist (*soft delete* inglise keeles), st lubab andmeid kustutada nii, et neid märgitakse kustutatuks, kuid eksisteerivad baasis edasi. Üldiselt peetakse andmete lõplikku kustutamist halvaks

praktikaks. Lisaks on kasutatud annotatsioone, et paika panna piirangud sisestatavatele andmetele andmebaasi tasemel.

ABP raamistiku heaks jooneks on ka projekti näiline lihtsus – taustal on palju abstraktsioone, mille peale arendaja enamasti mõtlema ei pea. *Leia.Application* on projekt, kus paikneb nii äriloogika kui rakenduse kiht. Andmed, mida rakendusega vahetatakse, defineeritakse DTO-des (*data transfer object* inglise keeles). Need on automaatselt kaardistatud mudelite pealt (need peavad kasutama samu nimesid). Lisaks on mugav kasutada annotatsioone, millega jällegi määrata nii piirangud kui ka veateated.

Äriloogika paikneb teenustes. Lihtsa CRUD veebiteenuse loomiseks piisab õigete asjade pärimisest (vt allolevat pilti).

```
using Abp.Application.Services;
using Abp.Authorization;
using Abp.Domain.Repositories;
using leia.Authorization;
using leia.Categories.Dto;

namespace leia.Categories
{
    [AbpAuthorize( params permissions: PermissionNames.Pages_Categories )]
    public class CategoryAppService : CrudAppService<Category, CategoryDto>
    {
        public CategoryAppService(IRepository<Category, int> repository) : base(repository)
        {
        }
    }
}
```








Joonis 9. Lihtne CRUD teenus.

Siin on kasutatud annotatsioone märkimaks õiguseid, mis peavad kasutajal olema teenuse kasutamiseks. ABP raamistik teeb kontrollid lennult. Neid saab ka manuaalselt lisada, kuid see ei ole enamasti vajalik. Põhimõtteliselt saab kõik lahendada teenuste defineerimisega, millele igale ühele eraldi saab anda konkreetseid ligipääsuõigused (vt *AbpAuthorize*). Raamistik teeb taustal ära ka tööüksused (*unit of work* inglise keeles) ning hoidlad (*repository* inglise keeles). [20]

### 6.1.2 Turvalisus

Turvalisus ja andmete kaitse on alati olnud oluline, kuid on tänapäeval järjest suureneva tähelepanu all. Üks ASP.NET Core plussidest on see, et kuna kasutajahaldus on juba nõ

kasutusvalmis (vt joonist 9), siis mõned tüüpilised vead, nt paroolide salvestamine avatud kujul ja kaitsetus SQL süstimise rünnakute vastu, on juba lahendatud. Turvalisuse tagamiseks on loodud rollid User ja Admin, kellest viimasel on õigused kõiki lehti näha ning pea kõike kustutada ja muuta. Uued rakenduses loodud kontod on User tüüpi ja Admin tüüpi kasutajaid registreerida ei saa. User tüübil on standardsed õigused, mida saab näha, muuta ja kustutada. Näiteks on oluline küll kõikide kasutajate nägemine, aga muuta saab ainult enda andmeid.

| Implicitly Installed Packages in Solution: 7                                      |  |        |
|---|--|--------|
|  | Microsoft.Identity.Client • (4.14.0)                             | 4.30.1 |
|  | Microsoft.IdentityModel.JsonWebTokens • (5.6.0; 6.7.1)           | 6.11.0 |
|  | Microsoft.IdentityModel.Logging • (5.6.0; 6.7.1)                 | 6.11.0 |
|  | Microsoft.IdentityModel.Protocols • (5.6.0; 6.7.1)               | 6.11.0 |
|  | Microsoft.IdentityModel.Protocols.OpenIdConnect • (5.6.0; 6.7.1) | 6.11.0 |
|  | Microsoft.IdentityModel.Tokens • (5.6.0; 6.7.1)                  | 6.11.0 |
|  | System.IdentityModel.Tokens.Jwt • (5.6.0; 6.7.1)                 | 6.11.0 |

Joonis 10. Identiteediga seotud paketid projektis.

Kuna vähemalt esialgu, kui pole loodud veel keerulisemaid lahendusi, on Leia rakenduses ürituste kuulutustes näha ka korraldaja telefoni number ja ürituse asukoht, siis on ka üritused näha ainult autoriseeritud, st sisse loginud kasutajatele.

## 6.2 Klientrakenduse arendus

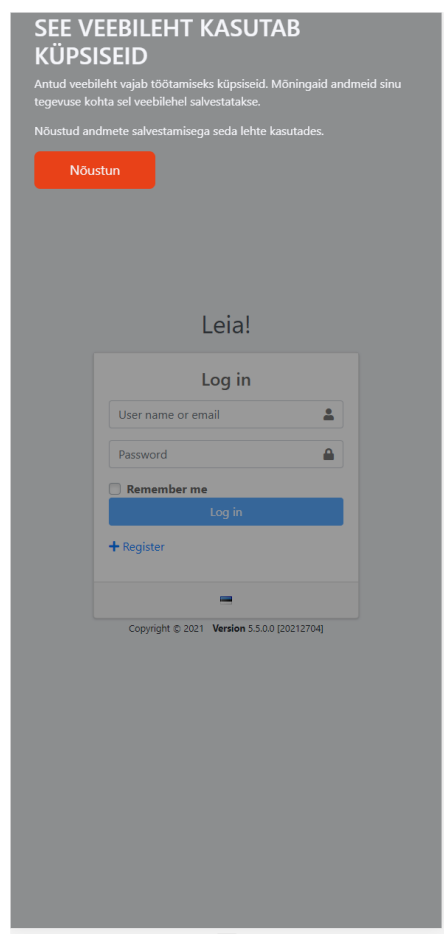
Klientrakenduse arenduseks valiti Angulari raamistik (versioon 11.0.7). ABP-d kasutatakse tihti just Angulariga, mis tähendab, et selle kohta on rohkem juhiseid ja õpetusi kui muude klientrakenduste kohta.

Peamised CRUD teenused Angularis sai genereerida automaatselt Swaggeri faili pealt (vt Lisa 2), mis on kokkuvõttev JSON fail API-de kohta, kasutades selleks nswaggerit.

Esimese suurema probleemina lahendati *tenant*-i valiku küsimus. Nimelt on baasprojektis ette nähtud selle valmine. Peamisesse, st omaniku (*host* inglise keeles) ruumi ei saa luua uusi kasutajaid. Seega muudeti veebirakendust nii, et see initsialiseeritaks alati konkreetse *tenant* identifikaatoriga, mida ei saa muuta.

Tänapäeval ei või Euroopa Liidus küpsiseid ja muud infot kasutaja kohta salvestada, kui selleks ei ole küsitud luba. Selle jaoks lisati rakendusele väga lihtne pealmine kiht, mis eest ära lükatakse, kui sellega nõustuda (vt joonis 11). Tegemist on kõige pealmise kihiga (kõrge z-indeks), mis on poolläbipaistev segaduse vältimiseks, kuid läbi sellest klikkida ei saa ilma lehte muutmata. Tasub mainimist, et kui muidu raamistik toetab lokaliseerimist, st erinevaid keeli, siis lihtsuse huvides on antud teade ning ka mitmed veateated tõlkimata. Üleüldiselt lisati serverisse nii eestikeelsed tõlked kui eesti keele tugi.

Rakenduse välimuse haldamiseks on kasutusele võetud Bootstrap raamistik (versioon 4.6.0). See on Twitteri poolt loodud raamistik, mis on mõeldud reageeriva disaini loomiseks. Selle üks eesmärke on luua kasutajale sarnane kogemus eri saitidel. Lisaks, et kasutajaliidest veelgi kohandada, on CSS asemel (õigemini selle täiendusena) kasutusel veidi mugavam Less.js (versioon 3.12.2). Ise tutvustab see end sõnadega „See on CSS, millel lihtsalt veidi rohkem.“ [27]



Joonis 11. Lihtne kate kasutaja andmete salvestamiseks luba küsimiseks.



Suureks väljakutseks oli ka vähegi mugavalt kasutatava kuupäeva ja kellaaja sisestamise pakkumine. Selleks katsetati erinevaid variante, mis muu hulgas tagastasid erinevaid objekte, kuid võitjaks osutus teek „@danielmoncada/angular-datetime-picker“ (versioon 11.0.0) [28]. Erinevaid teeki on aegade jooksul tehtud mitmeid, kuid Angulari uued versioonid on tutvustanud väga suuri muudatusi, millega teegid sammu pole pidanud. Antud lahendus näeb hea välja ning seda on üsna lihtne projekti lisada.

Baasprojekti oli leht jagatud juba mõistlikult erinevateks aladeks – jalus, päis (keel), päis (vasakpoolne navigatsiooniriba), päis (menüü), päis (komponendid), külgriba (logo), külgriba (menüü), külgriba (kasutaja paneel), külgriba (komponendid). Need on defineeritud kaustas *layout*, mis omakorda asub kaustas *app*. Viimane tähistab peamist kasutaja vaadet. Sisse logimise leht on eraldi vaade *account*, mis on mõistlik, sest see on täiesti eraldiseisev. Igal vaatel on ka oma „baas“. Näiteks failis *app-component-base.ts* tuleb loetleda kõik teenused, mida baasil vaja on. Vaade jagunebki komponentideks, millel on teenustega suhtlemise komponent ning sellega kaasas käib .html fail, mis on sisuliselt vormingu ja kasutajaga andmevahetuse kontrollimiseks.

Kogu veebiserveriga suhtlemise poolt juhitakse failist *service-proxies.ts*. Sinna ilmuvad nswaggeriga automaatselt genereeritud teenused.

### 6.3 Testimine

Antud projekti üks eesmärk on olla mobiilsetel seadmetel hästi kasutatav. Seetõttu kasutati arenduse käigus pea eksklusiivselt Chrome veebilehitseja pakutud tööriista – mobiilivaadet. Rakendust testiti ka Edge brauseril.

Serveri testimiseks kasutati Postmani programmi, kuid üldiselt arendati kliendi poolt serveriga paralleelselt, nii et suurem osa testimisest toimus täieliku süsteemiga ning arvutis jooksva Microsoft SQL serveri pihta.

## **7 Hinnang tehtud rakendusele**

Enamik kohustuslikest funktsionaalsustest said tehtud ning prototüüpi saab inimestele tagasiside saamiseks näidata. Kasutajakogemus ei ole hetkel veel parim ning mõned olulised omadused arendatakse lähiajal – näiteks kategooriate lisamine ning ürituste kategooriate järgi filtreerimine.

Raamistike ja baasprojekti valik oli väga õnnestunud ning need täitsid eesmärgi hästi. Baasprojekti kasutamine aitas saavutada ka hea kasutatavuse mobiili peal. Ürituste vaadet on vaja edasi arendada kasutusmugavuse suurendamiseks, kuid vaated kohanduvad eri ekraani suurustega hästi.

### **7.1 Uudsus**

Veebirakendus Leia on küsitlusele tuginedes oodatud ja vajalik, sest olemasolevad lahendused ei täida kasutajate vajadusi ja ootuseid - või on kasutusvaldkond liiga kitsas nagu MeetUp puhul.

Baasprojekti kasutamine oli autorile uus ja huvitav kogemus. Projekti on kaasatud uuemaid tehnoloogiaid, näiteks SignalR teek, mis kindlasti ka edaspidi kasutust leiab. On hea praktika tutvuda olemasolevate projektidega ja erinevate raamistikega, sest see avardab silmaringi ning annab võimaluse avastada, mida keegi teine peab parimateks praktikateks.

### **7.2 Võimalused edasiarenduseks**

Kõrvale jättes MVP-ks vajalikud omadused, on mitmeid võimalusi rakenduse edasi arendamiseks. Esimese sammuna lisatakse kasutajate hindamise süsteem, et halvasti käituvad kasutajad – näiteks keegi, kes mitu korda üritusele pärast registreerimist kohale ei tule – jääksid üritustest edaspidi kõrvale ning platvorm oleks kasutajatele turvalisem. Lisatakse ka reklaamide näitamise võimalus ning müüdav reklaampind.

Eriti pikas perspektiivis, kui platvormi peaks saatma edu, oleks sobiv mõelda ka tasulistele teenustele, näiteks võiks tasulise kontoga saada korraldada teistele piletiga üritusi.

Samuti on väga head alused teistesse riikidesse laienemiseks. Suuremal määral on juba toetatud erinevate keelte kasutus, kui kasutusele võtta ka erinevad *tenant*-id, saaks teha ka riigipõhiseid rakendusi, mis vastaks just kohalike ootustele. Valitud tehnoloogia lubab kasutada erinevaid klientrakendusi veelgi suuremaks oludega kohandamiseks.

## 8 Kokkuvõte

Antud diplomitöö eesmärk oli luua prototüüp veebirakendusele Leia, mis aitab igavlejal leida üritusi, millel osaleda, ning ettevõtmisteks inimesi otsival korraldajal leida kaaslaseid, kellega koos neid asju teha. Prototüüp on suuremalt jaolt valmis, kuid vajab arendusi enne vähimaks elujõuliseks tooteks saamist, misjärel oleks sobiv seda ka avalikult jagada.

Töö esimene osa keskendus nõuete defineerimisele, mis osaliselt tugines läbi viidud küsitlusele. Rakenduse idee meeldis paljudele ning võrdlemisi suur hulk vastanutest leidis, et antud rakenduses oles nende elus koht.

Edasi tutvustati põgusalt erinevaid tehnoloogiaid klient-server rakenduse loomiseks ja identifitseeriti sobivad lahendused. Prototüüp arendati ASP.NET-il ning Angular-il põhinevat baasprojekti (ASP.NET Boilerplate) kasutades, mis andis palju eeliseid, sealjuures ka suurem turvalisus ja parem kasutajakogemus.

Kliendi arendus osutus oodatust keerulisemaks, kuid oli väga õpetlik. Edasi võib juba oodata olulisi edasiarendusi kasutajakogemuse parendamiseks.

## Kasutatud kirjandus

- [1] L. C. Hawkey ja J. T. Cacioppo, „Loneliness matters: A theoretical and empirical review of consequences and mechanisms,” *Annals of behavioral medicine*, kd. 40, nr 2, pp. 218-227, 2010.
- [2] B. S. A. S. J. W. E. y. L. L. R. D. C. J. R. A. M. E. Primack, „ Social media use and perceived social isolation among young adults in the US,” *American journal of preventive medicine*, kd. 53, nr 1, pp. 1-8, 2017.
- [3] C. P. R. W. X. T. Y. X. L. H. C. S. H. R. C. Wang, „Immediate psychological responses and associated factors during the initial stage of the 2019 coronavirus disease (COVID-19) epidemic among the general population in China,” *International journal of environmental research and public health*, kd. 17, nr 5, p. 1729, 2020.
- [4] D. & B. R. Clegg, *Case method fast-track: a RAD approach*, Addison-Wesley Longman Publishing Co., Inc, 1994.
- [5] „SmarterASP.NET,” [Võrgumaterjal]. Available: [https://www.smarterasp.net/hosting\\_plans](https://www.smarterasp.net/hosting_plans). [Kasutatud 27 aprill 2021].
- [6] „Business of Apps,” [Võrgumaterjal]. Available: <https://www.businessofapps.com/data/tinder-statistics/>. [Kasutatud 27 aprill 2021].
- [7] „Light IT,” [Võrgumaterjal]. Available: <https://light-it.net/blog/choosing-a-technology-stack-for-web-application-development/>. [Kasutatud 27 aprill 2021].
- [8] J. E. J. G. M. a. H. E. S. Israel, „Separating data from function in a distributed file system,” nr No. CSL-78-5, 1978.
- [9] „Digital Thinker Help,” [Võrgumaterjal]. Available: <http://digitalthinkerhelp.com/what-is-client-server-network-example-advantages-disadvantages/>. [Kasutatud 27 aprill 2021].
- [10] „Geeks for Geeks,” [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/client-server-model/>. [Kasutatud 27 aprill 2021].
- [11] „Orangesoft,” [Võrgumaterjal]. Available: <https://orangesoft.co/blog/choosing-a-technology-stack-for-web-application-development>. [Kasutatud 27 aprill 2021].
- [12] „Section,” 4 Jaanuar 2021. [Võrgumaterjal]. Available: <https://www.section.io/engineering-education/choosing-between-django-flask-and-fastapi/>. [Kasutatud 8 mai 2021].
- [13] „Hackernoon,” [Võrgumaterjal]. Available: <https://hackernoon.com/php-laravel-vs-symfony-a-detailed-comparison-of-web-development-frameworks-sq493y3l>. [Kasutatud 5 mai 2021].
- [14] „altexsoft,” [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-node-js-web-app-development/>. [Kasutatud 6 mai 2021].

- [15] „PYPL - PopularitY of Programming Language,“ [Võrgumaterjal]. Available: <https://pypl.github.io/PYPL.html>. [Kasutatud 6 mai 2021].
- [16] „Medium,“ [Võrgumaterjal]. Available: <https://medium.com/javarevisited/10-popular-java-frameworks-for-web-applications-691c28f6c182>. [Kasutatud 6 mai 2021].
- [17] „W3Techs,“ [Võrgumaterjal]. Available: [https://w3techs.com/technologies/history\\_overview/programming\\_language](https://w3techs.com/technologies/history_overview/programming_language). [Kasutatud 6 mai 2021].
- [18] „Insights.StackOverflow,“ [Võrgumaterjal]. Available: <https://insights.stackoverflow.com/survey/2020#most-loved-dreaded-and-wanted>. [Kasutatud 7 mai 2021].
- [19] „Simform,“ [Võrgumaterjal]. Available: <https://www.simform.com/best-frontend-frameworks/>. [Kasutatud 27 aprill 2021].
- [20] „ASP.NET Boilerplate,“ [Võrgumaterjal]. Available: <https://aspnetboilerplate.com/Pages/Documents>. [Kasutatud 27 aprill 2021].
- [21] R. N. S. B. E. R. & N. S. B. Elmasri, Fundamentals of Database Systems, Addison-Wesley, 2000.
- [22] „Microsoft Docs,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/ef/core/providers/?tabs=dotnet-core-cli>. [Kasutatud 27 aprill 2021].
- [23] „Microsoft Docs,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-version-15?view=sql-server-ver15>. [Kasutatud 27 Aprill 2021].
- [24] „Statista,“ [Võrgumaterjal]. Available: <https://www.statista.com/statistics/306528/share-of-mobile-internet-traffic-in-global-regions/>. [Kasutatud 27 aprill 2021].
- [25] „Microsoft .NET What is .NET,“ [Võrgumaterjal]. Available: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>. [Kasutatud 28 aprill 2021].
- [26] „Microsoft .NET SignalR,“ [Võrgumaterjal]. Available: <https://dotnet.microsoft.com/apps/aspnet/signalr>. [Kasutatud 27 aprill 2021].
- [27] „less,“ [Võrgumaterjal]. Available: <https://lesscss.org/#>. [Kasutatud 27 aprill 2021].
- [28] D. P. Daniel Moncada, „npmjs.com,“ [Võrgumaterjal]. Available: <https://www.npmjs.com/package/@danielmoncada/angular-datetime-picker>. [Kasutatud 20 4 2021].
- [29] L. S. Sterling, The Art of Agent-Oriented Modeling, London: The MIT Press, 2009.

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Renata Muru

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Veebirakendus Leia eraisiku ürituste reklaamimiseks ja otsimiseks“, mille juhendaja on Meelis Antoi
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

15.05.2021

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Veebiteenuse API-d JSON kujul

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "leia API",
    "description": "leia",
    "contact": {
      "name": "leia",
      "url": "https://twitter.com/aspboilerplate",
      "email": ""
    },
    "license": {
      "name": "MIT License",
      "url":
"https://github.com/aspnetboilerplate/aspnetboilerplate/blob/dev/LICEN
SE"
    },
    "version": "v1"
  },
  "paths": {
    "/api/services/app/Account/IsTenantAvailable": {
      "post": {
        "tags": [
          "Account"
        ],
        "requestBody": {
          "content": {
            "application/json-patch+json": {
              "schema": {
                "$ref": "#/components/schemas/IsTenantAvailableInput"
              }
            },
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/IsTenantAvailableInput"
              }
            },
            "text/json": {
              "schema": {
                "$ref": "#/components/schemas/IsTenantAvailableInput"
              }
            }
          },
          "application/*+json": {
            "schema": {
              "$ref": "#/components/schemas/IsTenantAvailableInput"
            }
          }
        }
      },
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "text/plain": {
```



```

        "schema": {
          "$ref":
            "#/components/schemas/IsTenantAvailableOutput"
        }
      },
      "application/json": {
        "schema": {
          "$ref":
            "#/components/schemas/IsTenantAvailableOutput"
        }
      },
      "text/json": {
        "schema": {
          "$ref":
            "#/components/schemas/IsTenantAvailableOutput"
        }
      }
    }
  },
  "/api/services/app/Account/Register": {
    "post": {
      "tags": [
        "Account"
      ],
      "requestBody": {
        "content": {
          "application/json-patch+json": {
            "schema": {
              "$ref": "#/components/schemas/RegisterInput"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/RegisterInput"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/RegisterInput"
            }
          },
          "application/*+json": {
            "schema": {
              "$ref": "#/components/schemas/RegisterInput"
            }
          }
        }
      },
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "text/plain": {
              "schema": {
                "$ref": "#/components/schemas/RegisterOutput"
              }
            },
            "application/json": {

```

```

        "schema": {
          "$ref": "#/components/schemas/RegisterOutput"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/RegisterOutput"
        }
      }
    }
  }
},
"/api/services/app/Category/Get": {
  "get": {
    "tags": [
      "Category"
    ],
    "parameters": [
      {
        "name": "Id",
        "in": "query",
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "text/plain": {
            "schema": {
              "$ref": "#/components/schemas/CategoryDto"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/CategoryDto"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/CategoryDto"
            }
          }
        }
      }
    }
  }
},
"/api/services/app/Category/GetAll": {
  "get": {
    "tags": [
      "Category"
    ],
    "parameters": [
      {
        "name": "Sorting",

```

```

        "in": "query",
        "schema": {
            "type": "string",
            "nullable": true
        }
    },
    {
        "name": "SkipCount",
        "in": "query",
        "schema": {
            "maximum": 2147483647,
            "minimum": 0,
            "type": "integer",
            "format": "int32"
        }
    },
    {
        "name": "MaxResultCount",
        "in": "query",
        "schema": {
            "maximum": 2147483647,
            "minimum": 1,
            "type": "integer",
            "format": "int32"
        }
    }
],
"responses": {
    "200": {
        "description": "Success",
        "content": {
            "text/plain": {
                "schema": {
                    "$ref":
"/components/schemas/CategoryDtoPagedResultDto"
                }
            },
            "application/json": {
                "schema": {
                    "$ref":
"/components/schemas/CategoryDtoPagedResultDto"
                }
            },
            "text/json": {
                "schema": {
                    "$ref":
"/components/schemas/CategoryDtoPagedResultDto"
                }
            }
        }
    }
}
},
"/api/services/app/Category/Create": {
    "post": {
        "tags": [
            "Category"
        ],
        "requestBody": {
            "content": {

```

```

    "application/json-patch+json": {
      "schema": {
        "$ref": "#/components/schemas/CategoryDto"
      }
    },
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/CategoryDto"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/CategoryDto"
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/CategoryDto"
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/CategoryDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/CategoryDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/CategoryDto"
          }
        }
      }
    }
  }
},
"/api/services/app/Category/Update": {
  "put": {
    "tags": [
      "Category"
    ],
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {
            "$ref": "#/components/schemas/CategoryDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/CategoryDto"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "text/json": {
    "schema": {
      "$ref": "#/components/schemas/CategoryDto"
    }
  },
  "application/*+json": {
    "schema": {
      "$ref": "#/components/schemas/CategoryDto"
    }
  }
}
},
"responses": {
  "200": {
    "description": "Success",
    "content": {
      "text/plain": {
        "schema": {
          "$ref": "#/components/schemas/CategoryDto"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/CategoryDto"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/CategoryDto"
        }
      }
    }
  }
}
},
"/api/services/app/Category/Delete": {
  "delete": {
    "tags": [
      "Category"
    ],
    "parameters": [
      {
        "name": "Id",
        "in": "query",
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success"
      }
    }
  }
},
"/api/services/app/Configuration/ChangeUiTheme": {

```

```

"post": {
  "tags": [
    "Configuration"
  ],
  "requestBody": {
    "content": {
      "application/json-patch+json": {
        "schema": {
          "$ref": "#/components/schemas/ChangeUiThemeInput"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ChangeUiThemeInput"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/ChangeUiThemeInput"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/ChangeUiThemeInput"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    }
  }
},
"/api/services/app/Event/Create": {
  "post": {
    "tags": [
      "Event"
    ],
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {
            "$ref": "#/components/schemas/EventDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/EventDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/EventDto"
          }
        },
        "application/*+json": {
          "schema": {
            "$ref": "#/components/schemas/EventDto"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "responses": {
    "200": {
      "description": "Success",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/EventDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/EventDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/EventDto"
          }
        }
      }
    }
  }
},
"/api/services/app/Event/Get": {
  "get": {
    "tags": [
      "Event"
    ],
    "parameters": [
      {
        "name": "Id",
        "in": "query",
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "text/plain": {
            "schema": {
              "$ref": "#/components/schemas/EventDto"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/EventDto"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/EventDto"
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
}
},
"/api/services/app/Event/GetAll": {
  "get": {
    "tags": [
      "Event"
    ],
    "parameters": [
      {
        "name": "Sorting",
        "in": "query",
        "schema": {
          "type": "string",
          "nullable": true
        }
      },
      {
        "name": "SkipCount",
        "in": "query",
        "schema": {
          "maximum": 2147483647,
          "minimum": 0,
          "type": "integer",
          "format": "int32"
        }
      },
      {
        "name": "MaxResultCount",
        "in": "query",
        "schema": {
          "maximum": 2147483647,
          "minimum": 1,
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "text/plain": {
            "schema": {
              "$ref":
"#/components/schemas/EventDtoPagedResultDto"
            }
          },
          "application/json": {
            "schema": {
              "$ref":
"#/components/schemas/EventDtoPagedResultDto"
            }
          },
          "text/json": {
            "schema": {
              "$ref":
"#/components/schemas/EventDtoPagedResultDto"
            }
          }
        }
      }
    }
  }
}

```



```

        }
    }
}
},
"/api/services/app/Event/Update": {
    "put": {
        "tags": [
            "Event"
        ],
        "requestBody": {
            "content": {
                "application/json-patch+json": {
                    "schema": {
                        "$ref": "#/components/schemas/EventDto"
                    }
                },
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/EventDto"
                    }
                },
                "text/json": {
                    "schema": {
                        "$ref": "#/components/schemas/EventDto"
                    }
                },
                "application/*+json": {
                    "schema": {
                        "$ref": "#/components/schemas/EventDto"
                    }
                }
            }
        },
        "responses": {
            "200": {
                "description": "Success",
                "content": {
                    "text/plain": {
                        "schema": {
                            "$ref": "#/components/schemas/EventDto"
                        }
                    },
                    "application/json": {
                        "schema": {
                            "$ref": "#/components/schemas/EventDto"
                        }
                    },
                    "text/json": {
                        "schema": {
                            "$ref": "#/components/schemas/EventDto"
                        }
                    }
                }
            }
        }
    }
},
"/api/services/app/Event/Delete": {
    "delete": {

```

```

    "tags": [
      "Event"
    ],
    "parameters": [
      {
        "name": "Id",
        "in": "query",
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success"
      }
    }
  },
  "/api/services/app/Registration/Get": {
    "get": {
      "tags": [
        "Registration"
      ],
      "parameters": [
        {
          "name": "Id",
          "in": "query",
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "text/plain": {
              "schema": {
                "$ref": "#/components/schemas/RegistrationDto"
              }
            },
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/RegistrationDto"
              }
            },
            "text/json": {
              "schema": {
                "$ref": "#/components/schemas/RegistrationDto"
              }
            }
          }
        }
      }
    }
  },
  "/api/services/app/Registration/GetAll": {
    "get": {

```

```

"tags": [
  "Registration"
],
"parameters": [
  {
    "name": "Sorting",
    "in": "query",
    "schema": {
      "type": "string",
      "nullable": true
    }
  },
  {
    "name": "SkipCount",
    "in": "query",
    "schema": {
      "maximum": 2147483647,
      "minimum": 0,
      "type": "integer",
      "format": "int32"
    }
  },
  {
    "name": "MaxResultCount",
    "in": "query",
    "schema": {
      "maximum": 2147483647,
      "minimum": 1,
      "type": "integer",
      "format": "int32"
    }
  }
],
"responses": {
  "200": {
    "description": "Success",
    "content": {
      "text/plain": {
        "schema": {
          "$ref":
"#/components/schemas/RegistrationDtoPagedResultDto"
        }
      },
      "application/json": {
        "schema": {
          "$ref":
"#/components/schemas/RegistrationDtoPagedResultDto"
        }
      },
      "text/json": {
        "schema": {
          "$ref":
"#/components/schemas/RegistrationDtoPagedResultDto"
        }
      }
    }
  }
}
},
"/api/services/app/Registration/Create": {

```

```

"post": {
  "tags": [
    "Registration"
  ],
  "requestBody": {
    "content": {
      "application/json-patch+json": {
        "schema": {
          "$ref": "#/components/schemas/RegistrationDto"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/RegistrationDto"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/RegistrationDto"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/RegistrationDto"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/RegistrationDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/RegistrationDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/RegistrationDto"
          }
        }
      }
    }
  }
},
"/api/services/app/Registration/Update": {
  "put": {
    "tags": [
      "Registration"
    ],
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {

```

```

        "$ref": "#/components/schemas/RegistrationDto"
      }
    },
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/RegistrationDto"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/RegistrationDto"
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/RegistrationDto"
      }
    }
  }
},
"responses": {
  "200": {
    "description": "Success",
    "content": {
      "text/plain": {
        "schema": {
          "$ref": "#/components/schemas/RegistrationDto"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/RegistrationDto"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/RegistrationDto"
        }
      }
    }
  }
}
},
"/api/services/app/Registration/Delete": {
  "delete": {
    "tags": [
      "Registration"
    ],
    "parameters": [
      {
        "name": "Id",
        "in": "query",
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "200": {

```

```

        "description": "Success"
    }
}
},
"/api/services/app/Role/Create": {
    "post": {
        "tags": [
            "Role"
        ],
        "requestBody": {
            "content": {
                "application/json-patch+json": {
                    "schema": {
                        "$ref": "#/components/schemas/CreateRoleDto"
                    }
                },
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/CreateRoleDto"
                    }
                },
                "text/json": {
                    "schema": {
                        "$ref": "#/components/schemas/CreateRoleDto"
                    }
                },
                "application/*+json": {
                    "schema": {
                        "$ref": "#/components/schemas/CreateRoleDto"
                    }
                }
            }
        },
        "responses": {
            "200": {
                "description": "Success",
                "content": {
                    "text/plain": {
                        "schema": {
                            "$ref": "#/components/schemas/RoleDto"
                        }
                    },
                    "application/json": {
                        "schema": {
                            "$ref": "#/components/schemas/RoleDto"
                        }
                    },
                    "text/json": {
                        "schema": {
                            "$ref": "#/components/schemas/RoleDto"
                        }
                    }
                }
            }
        }
    }
},
"/api/services/app/Role/GetRoles": {
    "get": {
        "tags": [

```

```

    "Role"
  ],
  "parameters": [
    {
      "name": "Permission",
      "in": "query",
      "schema": {
        "type": "string",
        "nullable": true
      }
    }
  ],
  "responses": {
    "200": {
      "description": "Success",
      "content": {
        "text/plain": {
          "schema": {
            "$ref":
"#/components/schemas/RoleListDtoListResultDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref":
"#/components/schemas/RoleListDtoListResultDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref":
"#/components/schemas/RoleListDtoListResultDto"
          }
        }
      }
    }
  }
},
"/api/services/app/Role/Update": {
  "put": {
    "tags": [
      "Role"
    ],
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {
            "$ref": "#/components/schemas/RoleDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/RoleDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/RoleDto"
          }
        }
      }
    }
  },

```

```

        "application/*+json": {
            "schema": {
                "$ref": "#/components/schemas/RoleDto"
            }
        }
    },
    "responses": {
        "200": {
            "description": "Success",
            "content": {
                "text/plain": {
                    "schema": {
                        "$ref": "#/components/schemas/RoleDto"
                    }
                },
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/RoleDto"
                    }
                },
                "text/json": {
                    "schema": {
                        "$ref": "#/components/schemas/RoleDto"
                    }
                }
            }
        }
    }
},
"/api/services/app/Role/Delete": {
    "delete": {
        "tags": [
            "Role"
        ],
        "parameters": [
            {
                "name": "Id",
                "in": "query",
                "schema": {
                    "type": "integer",
                    "format": "int32"
                }
            }
        ],
        "responses": {
            "200": {
                "description": "Success"
            }
        }
    }
},
"/api/services/app/Role/GetAllPermissions": {
    "get": {
        "tags": [
            "Role"
        ],
        "responses": {
            "200": {
                "description": "Success",

```



```

        "content": {
          "text/plain": {
            "schema": {
              "$ref":
"/components/schemas/PermissionDtoListResultDto"
            }
          },
          "application/json": {
            "schema": {
              "$ref":
"/components/schemas/PermissionDtoListResultDto"
            }
          },
          "text/json": {
            "schema": {
              "$ref":
"/components/schemas/PermissionDtoListResultDto"
            }
          }
        }
      }
    },
    "/api/services/app/Role/GetRoleForEdit": {
      "get": {
        "tags": [
          "Role"
        ],
        "parameters": [
          {
            "name": "Id",
            "in": "query",
            "schema": {
              "type": "integer",
              "format": "int32"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Success",
            "content": {
              "text/plain": {
                "schema": {
                  "$ref": "#/components/schemas/GetRoleForEditOutput"
                }
              },
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/GetRoleForEditOutput"
                }
              },
              "text/json": {
                "schema": {
                  "$ref": "#/components/schemas/GetRoleForEditOutput"
                }
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "/api/services/app/Role/Get": {
    "get": {
      "tags": [
        "Role"
      ],
      "parameters": [
        {
          "name": "Id",
          "in": "query",
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "text/plain": {
              "schema": {
                "$ref": "#/components/schemas/RoleDto"
              }
            },
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/RoleDto"
              }
            },
            "text/json": {
              "schema": {
                "$ref": "#/components/schemas/RoleDto"
              }
            }
          }
        }
      }
    }
  },
  "/api/services/app/Role/GetAll": {
    "get": {
      "tags": [
        "Role"
      ],
      "parameters": [
        {
          "name": "Keyword",
          "in": "query",
          "schema": {
            "type": "string",
            "nullable": true
          }
        },
        {
          "name": "SkipCount",
          "in": "query",
          "schema": {
            "maximum": 2147483647,
            "minimum": 0,

```

```

        "type": "integer",
        "format": "int32"
    }
},
{
    "name": "MaxResultCount",
    "in": "query",
    "schema": {
        "maximum": 2147483647,
        "minimum": 1,
        "type": "integer",
        "format": "int32"
    }
}
],
"responses": {
    "200": {
        "description": "Success",
        "content": {
            "text/plain": {
                "schema": {
                    "$ref": "#/components/schemas/RoleDtoPagedResultDto"
                }
            },
            "application/json": {
                "schema": {
                    "$ref": "#/components/schemas/RoleDtoPagedResultDto"
                }
            },
            "text/json": {
                "schema": {
                    "$ref": "#/components/schemas/RoleDtoPagedResultDto"
                }
            }
        }
    }
}
},
"/api/services/app/Session/GetCurrentLoginInformations": {
    "get": {
        "tags": [
            "Session"
        ],
        "responses": {
            "200": {
                "description": "Success",
                "content": {
                    "text/plain": {
                        "schema": {
                            "$ref":
"#/components/schemas/GetCurrentLoginInformationsOutput"
                        }
                    },
                    "application/json": {
                        "schema": {
                            "$ref":
"#/components/schemas/GetCurrentLoginInformationsOutput"
                        }
                    },
                    "text/json": {

```

```

                "schema": {
                    "$ref":
"/components/schemas/GetCurrentLoginInformationsOutput"
                }
            }
        }
    },
"/api/services/app/Tenant/Create": {
    "post": {
        "tags": [
            "Tenant"
        ],
        "requestBody": {
            "content": {
                "application/json-patch+json": {
                    "schema": {
                        "$ref": "#/components/schemas/CreateTenantDto"
                    }
                },
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/CreateTenantDto"
                    }
                },
                "text/json": {
                    "schema": {
                        "$ref": "#/components/schemas/CreateTenantDto"
                    }
                },
                "application/*+json": {
                    "schema": {
                        "$ref": "#/components/schemas/CreateTenantDto"
                    }
                }
            }
        },
        "responses": {
            "200": {
                "description": "Success",
                "content": {
                    "text/plain": {
                        "schema": {
                            "$ref": "#/components/schemas/TenantDto"
                        }
                    },
                    "application/json": {
                        "schema": {
                            "$ref": "#/components/schemas/TenantDto"
                        }
                    },
                    "text/json": {
                        "schema": {
                            "$ref": "#/components/schemas/TenantDto"
                        }
                    }
                }
            }
        }
    }
}
}

```

```

    }
  },
  "/api/services/app/Tenant/Delete": {
    "delete": {
      "tags": [
        "Tenant"
      ],
      "parameters": [
        {
          "name": "Id",
          "in": "query",
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success"
        }
      }
    }
  },
  "/api/services/app/Tenant/Get": {
    "get": {
      "tags": [
        "Tenant"
      ],
      "parameters": [
        {
          "name": "Id",
          "in": "query",
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "text/plain": {
              "schema": {
                "$ref": "#/components/schemas/TenantDto"
              }
            },
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/TenantDto"
              }
            },
            "text/json": {
              "schema": {
                "$ref": "#/components/schemas/TenantDto"
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "/api/services/app/Tenant/GetAll": {
    "get": {
      "tags": [
        "Tenant"
      ],
      "parameters": [
        {
          "name": "Keyword",
          "in": "query",
          "schema": {
            "type": "string",
            "nullable": true
          }
        },
        {
          "name": "IsActive",
          "in": "query",
          "schema": {
            "type": "boolean",
            "nullable": true
          }
        },
        {
          "name": "SkipCount",
          "in": "query",
          "schema": {
            "maximum": 2147483647,
            "minimum": 0,
            "type": "integer",
            "format": "int32"
          }
        },
        {
          "name": "MaxResultCount",
          "in": "query",
          "schema": {
            "maximum": 2147483647,
            "minimum": 1,
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "text/plain": {
              "schema": {
                "$ref":
"#/components/schemas/TenantDtoPagedResultDto"
              }
            },
            "application/json": {
              "schema": {
                "$ref":
"#/components/schemas/TenantDtoPagedResultDto"
              }
            }
          }
        }
      }
    }
  }
}

```

```

        "text/json": {
            "schema": {
                "$ref":
"/components/schemas/TenantDtoPagedResultDto"
            }
        }
    },
    "/api/services/app/Tenant/Update": {
        "put": {
            "tags": [
                "Tenant"
            ],
            "requestBody": {
                "content": {
                    "application/json-patch+json": {
                        "schema": {
                            "$ref": "#/components/schemas/TenantDto"
                        }
                    },
                    "application/json": {
                        "schema": {
                            "$ref": "#/components/schemas/TenantDto"
                        }
                    },
                    "text/json": {
                        "schema": {
                            "$ref": "#/components/schemas/TenantDto"
                        }
                    },
                    "application/*+json": {
                        "schema": {
                            "$ref": "#/components/schemas/TenantDto"
                        }
                    }
                }
            },
            "responses": {
                "200": {
                    "description": "Success",
                    "content": {
                        "text/plain": {
                            "schema": {
                                "$ref": "#/components/schemas/TenantDto"
                            }
                        },
                        "application/json": {
                            "schema": {
                                "$ref": "#/components/schemas/TenantDto"
                            }
                        },
                        "text/json": {
                            "schema": {
                                "$ref": "#/components/schemas/TenantDto"
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
  },
  "/api/TokenAuth/Authenticate": {
    "post": {
      "tags": [
        "TokenAuth"
      ],
      "requestBody": {
        "content": {
          "application/json-patch+json": {
            "schema": {
              "$ref": "#/components/schemas/AuthenticateModel"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/AuthenticateModel"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/AuthenticateModel"
            }
          },
          "application/*+json": {
            "schema": {
              "$ref": "#/components/schemas/AuthenticateModel"
            }
          }
        }
      },
      "responses": {
        "200": {
          "description": "Success",
          "content": {
            "text/plain": {
              "schema": {
                "$ref":
"#/components/schemas/AuthenticateResultModel"
              }
            },
            "application/json": {
              "schema": {
                "$ref":
"#/components/schemas/AuthenticateResultModel"
              }
            },
            "text/json": {
              "schema": {
                "$ref":
"#/components/schemas/AuthenticateResultModel"
              }
            }
          }
        }
      }
    }
  },
  "/api/TokenAuth/GetExternalAuthenticationProviders": {
    "get": {

```



```

    "tags": [
      "TokenAuth"
    ],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "text/plain": {
            "schema": {
              "type": "array",
              "items": {
                "$ref":
"#/components/schemas/ExternalLoginProviderInfoModel"
              }
            }
          },
          "application/json": {
            "schema": {
              "type": "array",
              "items": {
                "$ref":
"#/components/schemas/ExternalLoginProviderInfoModel"
              }
            }
          },
          "text/json": {
            "schema": {
              "type": "array",
              "items": {
                "$ref":
"#/components/schemas/ExternalLoginProviderInfoModel"
              }
            }
          }
        }
      }
    }
  },
  "/api/TokenAuth/ExternalAuthenticate": {
    "post": {
      "tags": [
        "TokenAuth"
      ],
      "requestBody": {
        "content": {
          "application/json-patch+json": {
            "schema": {
              "$ref":
"#/components/schemas/ExternalAuthenticateModel"
            }
          },
          "application/json": {
            "schema": {
              "$ref":
"#/components/schemas/ExternalAuthenticateModel"
            }
          },
          "text/json": {
            "schema": {

```

```

        "$ref":
"#/components/schemas/ExternalAuthenticateModel"
    }
  },
  "application/*+json": {
    "schema": {
      "$ref":
"#/components/schemas/ExternalAuthenticateModel"
    }
  }
},
"responses": {
  "200": {
    "description": "Success",
    "content": {
      "text/plain": {
        "schema": {
          "$ref":
"#/components/schemas/ExternalAuthenticateResultModel"
        }
      },
      "application/json": {
        "schema": {
          "$ref":
"#/components/schemas/ExternalAuthenticateResultModel"
        }
      },
      "text/json": {
        "schema": {
          "$ref":
"#/components/schemas/ExternalAuthenticateResultModel"
        }
      }
    }
  }
}
},
"/api/services/app/User/Create": {
  "post": {
    "tags": [
      "User"
    ],
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {
            "$ref": "#/components/schemas/CreateUserDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/CreateUserDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/CreateUserDto"
          }
        }
      }
    }
  }
},

```

```

        "application/*+json": {
            "schema": {
                "$ref": "#/components/schemas/CreateUserDto"
            }
        }
    },
    "responses": {
        "200": {
            "description": "Success",
            "content": {
                "text/plain": {
                    "schema": {
                        "$ref": "#/components/schemas/UserDto"
                    }
                },
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/UserDto"
                    }
                },
                "text/json": {
                    "schema": {
                        "$ref": "#/components/schemas/UserDto"
                    }
                }
            }
        }
    }
},
"/api/services/app/User/Update": {
    "put": {
        "tags": [
            "User"
        ],
        "requestBody": {
            "content": {
                "application/json-patch+json": {
                    "schema": {
                        "$ref": "#/components/schemas/UserDto"
                    }
                },
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/UserDto"
                    }
                },
                "text/json": {
                    "schema": {
                        "$ref": "#/components/schemas/UserDto"
                    }
                },
                "application/*+json": {
                    "schema": {
                        "$ref": "#/components/schemas/UserDto"
                    }
                }
            }
        }
    },
    "responses": {

```

```

    "200": {
      "description": "Success",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/UserDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/UserDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/UserDto"
          }
        }
      }
    }
  },
  "/api/services/app/User/Delete": {
    "delete": {
      "tags": [
        "User"
      ],
      "parameters": [
        {
          "name": "Id",
          "in": "query",
          "schema": {
            "type": "integer",
            "format": "int64"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "Success"
        }
      }
    }
  },
  "/api/services/app/User/Activate": {
    "post": {
      "tags": [
        "User"
      ],
      "requestBody": {
        "content": {
          "application/json-patch+json": {
            "schema": {
              "$ref": "#/components/schemas/Int64EntityDto"
            }
          },
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/Int64EntityDto"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/Int64EntityDto"
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/Int64EntityDto"
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    }
  }
},
"/api/services/app/User/DeActivate": {
  "post": {
    "tags": [
      "User"
    ],
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {
            "$ref": "#/components/schemas/Int64EntityDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/Int64EntityDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/Int64EntityDto"
          }
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/Int64EntityDto"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    }
  }
},
"/api/services/app/User/GetRoles": {
  "get": {
    "tags": [
      "User"
    ],

```

```

"responses": {
  "200": {
    "description": "Success",
    "content": {
      "text/plain": {
        "schema": {
          "$ref": "#/components/schemas/RoleDtoListResultDto"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/RoleDtoListResultDto"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/RoleDtoListResultDto"
        }
      }
    }
  }
},
"/api/services/app/User/ChangeLanguage": {
  "post": {
    "tags": [
      "User"
    ],
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {
            "$ref": "#/components/schemas/ChangeUserLanguageDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ChangeUserLanguageDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ChangeUserLanguageDto"
          }
        },
        "application/*+json": {
          "schema": {
            "$ref": "#/components/schemas/ChangeUserLanguageDto"
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      }
    }
  }
},
"/api/services/app/User/ChangePassword": {

```

```

"post": {
  "tags": [
    "User"
  ],
  "requestBody": {
    "content": {
      "application/json-patch+json": {
        "schema": {
          "$ref": "#/components/schemas/ChangePasswordDto"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ChangePasswordDto"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/ChangePasswordDto"
        }
      },
      "application/*+json": {
        "schema": {
          "$ref": "#/components/schemas/ChangePasswordDto"
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success",
      "content": {
        "text/plain": {
          "schema": {
            "type": "boolean"
          }
        },
        "application/json": {
          "schema": {
            "type": "boolean"
          }
        },
        "text/json": {
          "schema": {
            "type": "boolean"
          }
        }
      }
    }
  }
},
"/api/services/app/User/ResetPassword": {
  "post": {
    "tags": [
      "User"
    ],
    "requestBody": {
      "content": {
        "application/json-patch+json": {
          "schema": {

```

```

        "$ref": "#/components/schemas/ResetPasswordDto"
    }
},
"application/json": {
    "schema": {
        "$ref": "#/components/schemas/ResetPasswordDto"
    }
},
"text/json": {
    "schema": {
        "$ref": "#/components/schemas/ResetPasswordDto"
    }
},
"application/*+json": {
    "schema": {
        "$ref": "#/components/schemas/ResetPasswordDto"
    }
}
},
"responses": {
    "200": {
        "description": "Success",
        "content": {
            "text/plain": {
                "schema": {
                    "type": "boolean"
                }
            },
            "application/json": {
                "schema": {
                    "type": "boolean"
                }
            },
            "text/json": {
                "schema": {
                    "type": "boolean"
                }
            }
        }
    }
}
},
"/api/services/app/User/Get": {
    "get": {
        "tags": [
            "User"
        ],
        "parameters": [
            {
                "name": "Id",
                "in": "query",
                "schema": {
                    "type": "integer",
                    "format": "int64"
                }
            }
        ],
        "responses": {
            "200": {

```



```

    "description": "Success",
    "content": {
      "text/plain": {
        "schema": {
          "$ref": "#/components/schemas/UserDto"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/UserDto"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/UserDto"
        }
      }
    }
  }
},
"/api/services/app/User/GetAll": {
  "get": {
    "tags": [
      "User"
    ],
    "parameters": [
      {
        "name": "Keyword",
        "in": "query",
        "schema": {
          "type": "string",
          "nullable": true
        }
      },
      {
        "name": "IsActive",
        "in": "query",
        "schema": {
          "type": "boolean",
          "nullable": true
        }
      },
      {
        "name": "SkipCount",
        "in": "query",
        "schema": {
          "maximum": 2147483647,
          "minimum": 0,
          "type": "integer",
          "format": "int32"
        }
      },
      {
        "name": "MaxResultCount",
        "in": "query",
        "schema": {
          "maximum": 2147483647,
          "minimum": 1,
          "type": "integer",

```

```

        "format": "int32"
      }
    }
  ],
  "responses": {
    "200": {
      "description": "Success",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/UserDtoPagedResultDto"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/UserDtoPagedResultDto"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/UserDtoPagedResultDto"
          }
        }
      }
    }
  }
}
},
"components": {
  "schemas": {
    "IsTenantAvailableInput": {
      "required": [
        "tenancyName"
      ],
      "type": "object",
      "properties": {
        "tenancyName": {
          "maxLength": 64,
          "minLength": 0,
          "type": "string"
        }
      }
    },
    "additionalProperties": false
  },
  "TenantAvailabilityState": {
    "enum": [
      1,
      2,
      3
    ],
    "type": "integer",
    "format": "int32"
  },
  "IsTenantAvailableOutput": {
    "type": "object",
    "properties": {
      "state": {
        "$ref": "#/components/schemas/TenantAvailabilityState"
      },
      "tenantId": {

```

```

        "type": "integer",
        "format": "int32",
        "nullable": true
    }
},
"additionalProperties": false
},
"RegisterInput": {
    "required": [
        "emailAddress",
        "name",
        "password",
        "surname",
        "userName"
    ],
    "type": "object",
    "properties": {
        "name": {
            "maxLength": 64,
            "minLength": 0,
            "type": "string"
        },
        "surname": {
            "maxLength": 64,
            "minLength": 0,
            "type": "string"
        },
        "userName": {
            "maxLength": 256,
            "minLength": 0,
            "type": "string"
        },
        "emailAddress": {
            "maxLength": 256,
            "minLength": 0,
            "type": "string",
            "format": "email"
        },
        "password": {
            "maxLength": 32,
            "minLength": 0,
            "type": "string"
        },
        "captchaResponse": {
            "type": "string",
            "nullable": true
        },
        "description": {
            "maxLength": 4096,
            "type": "string",
            "nullable": true
        },
        "profilePicture": {
            "maxLength": 16000,
            "type": "string",
            "format": "byte",
            "nullable": true
        }
    }
},
"additionalProperties": false
},

```

```

"RegisterOutput": {
  "type": "object",
  "properties": {
    "canLogin": {
      "type": "boolean"
    }
  },
  "additionalProperties": false
},
"CategoryDto": {
  "type": "object",
  "properties": {
    "name": {
      "maxLength": 256,
      "minLength": 1,
      "type": "string",
      "nullable": true
    },
    "description": {
      "maxLength": 4096,
      "type": "string",
      "nullable": true
    },
    "id": {
      "type": "integer",
      "format": "int32"
    }
  },
  "additionalProperties": false
},
"CategoryDtoPagedResultDto": {
  "type": "object",
  "properties": {
    "totalCount": {
      "type": "integer",
      "format": "int32"
    },
    "items": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/CategoryDto"
      },
      "nullable": true
    }
  },
  "additionalProperties": false
},
"ChangeUiThemeInput": {
  "required": [
    "theme"
  ],
  "type": "object",
  "properties": {
    "theme": {
      "maxLength": 32,
      "minLength": 0,
      "type": "string"
    }
  },
  "additionalProperties": false
},

```

```

"EventDto": {
  "type": "object",
  "properties": {
    "categoryId": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "name": {
      "maxLength": 256,
      "minLength": 1,
      "type": "string",
      "nullable": true
    },
    "description": {
      "maxLength": 4096,
      "type": "string",
      "nullable": true
    },
    "location": {
      "maxLength": 256,
      "type": "string",
      "nullable": true
    },
    "phone": {
      "maxLength": 256,
      "type": "string",
      "nullable": true
    },
    "email": {
      "maxLength": 4096,
      "type": "string",
      "format": "email",
      "nullable": true
    },
    "notes": {
      "maxLength": 4096,
      "type": "string",
      "nullable": true
    },
    "limit": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "creationTime": {
      "type": "string",
      "format": "date-time"
    },
    "eventTime": {
      "type": "string",
      "format": "date-time"
    },
    "creatorUserId": {
      "type": "integer",
      "format": "int64",
      "nullable": true
    },
    "id": {
      "type": "integer",
      "format": "int32"
    }
  }
}

```

```

    }
  },
  "additionalProperties": false
},
"EventDtoPagedResultDto": {
  "type": "object",
  "properties": {
    "totalCount": {
      "type": "integer",
      "format": "int32"
    },
    "items": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/EventDto"
      }
    },
    "nullable": true
  }
},
"additionalProperties": false
},
"RegistrationDto": {
  "type": "object",
  "properties": {
    "eventId": {
      "type": "integer",
      "format": "int32"
    },
    "approval": {
      "type": "boolean"
    },
    "approvedTime": {
      "type": "string",
      "format": "date-time"
    },
    "creationTime": {
      "type": "string",
      "format": "date-time"
    },
    "creatorUserId": {
      "type": "integer",
      "format": "int64",
      "nullable": true
    },
    "id": {
      "type": "integer",
      "format": "int32"
    }
  }
},
"additionalProperties": false
},
"RegistrationDtoPagedResultDto": {
  "type": "object",
  "properties": {
    "totalCount": {
      "type": "integer",
      "format": "int32"
    },
    "items": {
      "type": "array",
      "items": {

```

```

        "$ref": "#/components/schemas/RegistrationDto"
      },
      "nullable": true
    }
  },
  "additionalProperties": false
},
"CreateRoleDto": {
  "required": [
    "displayName",
    "name"
  ],
  "type": "object",
  "properties": {
    "name": {
      "maxLength": 32,
      "minLength": 0,
      "type": "string"
    },
    "displayName": {
      "maxLength": 64,
      "minLength": 0,
      "type": "string"
    },
    "normalizedName": {
      "type": "string",
      "nullable": true
    },
    "description": {
      "maxLength": 5000,
      "minLength": 0,
      "type": "string",
      "nullable": true
    },
    "grantedPermissions": {
      "type": "array",
      "items": {
        "type": "string"
      },
      "nullable": true
    }
  },
  "additionalProperties": false
},
"RoleDto": {
  "required": [
    "displayName",
    "name"
  ],
  "type": "object",
  "properties": {
    "name": {
      "maxLength": 32,
      "minLength": 0,
      "type": "string"
    },
    "displayName": {
      "maxLength": 64,
      "minLength": 0,
      "type": "string"
    },
  },

```

```

        "normalizedName": {
            "type": "string",
            "nullable": true
        },
        "description": {
            "maxLength": 5000,
            "minLength": 0,
            "type": "string",
            "nullable": true
        },
        "grantedPermissions": {
            "type": "array",
            "items": {
                "type": "string"
            },
            "nullable": true
        },
        "id": {
            "type": "integer",
            "format": "int32"
        }
    },
    "additionalProperties": false
},
"RoleListDto": {
    "type": "object",
    "properties": {
        "name": {
            "type": "string",
            "nullable": true
        },
        "displayName": {
            "type": "string",
            "nullable": true
        },
        "isStatic": {
            "type": "boolean"
        },
        "isDefault": {
            "type": "boolean"
        },
        "creationTime": {
            "type": "string",
            "format": "date-time"
        },
        "id": {
            "type": "integer",
            "format": "int32"
        }
    },
    "additionalProperties": false
},
"RoleListDtoListResultDto": {
    "type": "object",
    "properties": {
        "items": {
            "type": "array",
            "items": {
                "$ref": "#/components/schemas/RoleListDto"
            },
            "nullable": true
        }
    }
}

```



```

    }
  },
  "additionalProperties": false
},
"PermissionDto": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "nullable": true
    },
    "displayName": {
      "type": "string",
      "nullable": true
    },
    "description": {
      "type": "string",
      "nullable": true
    },
    "id": {
      "type": "integer",
      "format": "int64"
    }
  },
  "additionalProperties": false
},
"PermissionDtoListResultDto": {
  "type": "object",
  "properties": {
    "items": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/PermissionDto"
      },
      "nullable": true
    }
  },
  "additionalProperties": false
},
"RoleEditDto": {
  "required": [
    "displayName",
    "name"
  ],
  "type": "object",
  "properties": {
    "name": {
      "maxLength": 32,
      "minLength": 0,
      "type": "string"
    },
    "displayName": {
      "maxLength": 64,
      "minLength": 0,
      "type": "string"
    },
    "description": {
      "maxLength": 5000,
      "minLength": 0,
      "type": "string",
      "nullable": true
    }
  }
}

```

```

    },
    "isStatic": {
      "type": "boolean"
    },
    "id": {
      "type": "integer",
      "format": "int32"
    }
  },
  "additionalProperties": false
},
"FlatPermissionDto": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "nullable": true
    },
    "displayName": {
      "type": "string",
      "nullable": true
    },
    "description": {
      "type": "string",
      "nullable": true
    }
  },
  "additionalProperties": false
},
"GetRoleForEditOutput": {
  "type": "object",
  "properties": {
    "role": {
      "$ref": "#/components/schemas/RoleEditDto"
    },
    "permissions": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/FlatPermissionDto"
      },
      "nullable": true
    },
    "grantedPermissionNames": {
      "type": "array",
      "items": {
        "type": "string"
      },
      "nullable": true
    }
  },
  "additionalProperties": false
},
"RoleDtoPagedResultDto": {
  "type": "object",
  "properties": {
    "totalCount": {
      "type": "integer",
      "format": "int32"
    },
    "items": {
      "type": "array",

```

```

        "items": {
          "$ref": "#/components/schemas/RoleDto"
        },
        "nullable": true
      }
    },
    "additionalProperties": false
  },
  "ApplicationInfoDto": {
    "type": "object",
    "properties": {
      "version": {
        "type": "string",
        "nullable": true
      },
      "releaseDate": {
        "type": "string",
        "format": "date-time"
      },
      "features": {
        "type": "object",
        "additionalProperties": {
          "type": "boolean"
        },
        "nullable": true
      }
    },
    "additionalProperties": false
  },
  "UserLoginInfoDto": {
    "type": "object",
    "properties": {
      "name": {
        "type": "string",
        "nullable": true
      },
      "surname": {
        "type": "string",
        "nullable": true
      },
      "userName": {
        "type": "string",
        "nullable": true
      },
      "emailAddress": {
        "type": "string",
        "nullable": true
      },
      "id": {
        "type": "integer",
        "format": "int64"
      }
    },
    "additionalProperties": false
  },
  "TenantLoginInfoDto": {
    "type": "object",
    "properties": {
      "tenancyName": {
        "type": "string",
        "nullable": true
      }
    }
  }
}

```

```

    },
    "name": {
      "type": "string",
      "nullable": true
    },
    "id": {
      "type": "integer",
      "format": "int32"
    }
  },
  "additionalProperties": false
},
"GetCurrentLoginInformationsOutput": {
  "type": "object",
  "properties": {
    "application": {
      "$ref": "#/components/schemas/ApplicationInfoDto"
    },
    "user": {
      "$ref": "#/components/schemas/UserLoginInfoDto"
    },
    "tenant": {
      "$ref": "#/components/schemas/TenantLoginInfoDto"
    }
  },
  "additionalProperties": false
},
"CreateTenantDto": {
  "required": [
    "adminEmailAddress",
    "name",
    "tenancyName"
  ],
  "type": "object",
  "properties": {
    "tenancyName": {
      "maxLength": 64,
      "minLength": 0,
      "pattern": "^[a-zA-Z][a-zA-Z0-9_-]{1,}$",
      "type": "string"
    },
    "name": {
      "maxLength": 128,
      "minLength": 0,
      "type": "string"
    },
    "adminEmailAddress": {
      "maxLength": 256,
      "minLength": 0,
      "type": "string"
    },
    "connectionString": {
      "maxLength": 1024,
      "minLength": 0,
      "type": "string",
      "nullable": true
    },
    "isActive": {
      "type": "boolean"
    }
  }
},

```

```

    "additionalProperties": false
  },
  "TenantDto": {
    "required": [
      "name",
      "tenancyName"
    ],
    "type": "object",
    "properties": {
      "tenancyName": {
        "maxLength": 64,
        "minLength": 0,
        "pattern": "^[a-zA-Z][a-zA-Z0-9_-]{1,}$",
        "type": "string"
      },
      "name": {
        "maxLength": 128,
        "minLength": 0,
        "type": "string"
      },
      "isActive": {
        "type": "boolean"
      },
      "id": {
        "type": "integer",
        "format": "int32"
      }
    }
  },
  "additionalProperties": false
},
"TenantDtoPagedResultDto": {
  "type": "object",
  "properties": {
    "totalCount": {
      "type": "integer",
      "format": "int32"
    },
    "items": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/TenantDto"
      },
      "nullable": true
    }
  }
},
"additionalProperties": false
},
"AuthenticateModel": {
  "required": [
    "password",
    "userNameOrEmailAddress"
  ],
  "type": "object",
  "properties": {
    "userNameOrEmailAddress": {
      "maxLength": 256,
      "minLength": 0,
      "type": "string"
    },
    "password": {
      "maxLength": 32,

```

```

        "minLength": 0,
        "type": "string"
    },
    "rememberClient": {
        "type": "boolean"
    }
},
"additionalProperties": false
},
"AuthenticateResultModel": {
    "type": "object",
    "properties": {
        "accessToken": {
            "type": "string",
            "nullable": true
        },
        "encryptedAccessToken": {
            "type": "string",
            "nullable": true
        },
        "expireInSeconds": {
            "type": "integer",
            "format": "int32"
        },
        "userId": {
            "type": "integer",
            "format": "int64"
        }
    },
    "additionalProperties": false
},
"ExternalLoginProviderInfoModel": {
    "type": "object",
    "properties": {
        "name": {
            "type": "string",
            "nullable": true
        },
        "clientId": {
            "type": "string",
            "nullable": true
        }
    },
    "additionalProperties": false
},
"ExternalAuthenticateModel": {
    "required": [
        "authProvider",
        "providerAccessCode",
        "providerKey"
    ],
    "type": "object",
    "properties": {
        "authProvider": {
            "maxLength": 128,
            "minLength": 0,
            "type": "string"
        },
        "providerKey": {
            "maxLength": 256,
            "minLength": 0,

```

```

        "type": "string"
    },
    "providerAccessCode": {
        "type": "string"
    }
},
"additionalProperties": false
},
"ExternalAuthenticateResultModel": {
    "type": "object",
    "properties": {
        "accessToken": {
            "type": "string",
            "nullable": true
        },
        "encryptedAccessToken": {
            "type": "string",
            "nullable": true
        },
        "expireInSeconds": {
            "type": "integer",
            "format": "int32"
        },
        "waitingForActivation": {
            "type": "boolean"
        }
    },
    "additionalProperties": false
},
"CreateUserDto": {
    "required": [
        "emailAddress",
        "name",
        "password",
        "surname",
        "userName"
    ],
    "type": "object",
    "properties": {
        "userName": {
            "maxLength": 256,
            "minLength": 0,
            "type": "string"
        },
        "name": {
            "maxLength": 64,
            "minLength": 0,
            "type": "string"
        },
        "surname": {
            "maxLength": 64,
            "minLength": 0,
            "type": "string"
        },
        "emailAddress": {
            "maxLength": 256,
            "minLength": 0,
            "type": "string",
            "format": "email"
        },
        "isActive": {

```

```

        "type": "boolean"
    },
    "roleNames": {
        "type": "array",
        "items": {
            "type": "string"
        },
        "nullable": true
    },
    "password": {
        "maxLength": 32,
        "minLength": 0,
        "type": "string"
    },
    "description": {
        "maxLength": 4096,
        "type": "string",
        "nullable": true
    },
    "profilePicture": {
        "maxLength": 16000,
        "type": "string",
        "format": "byte",
        "nullable": true
    }
},
"additionalProperties": false
},
"UserDto": {
    "required": [
        "emailAddress",
        "name",
        "surname",
        "userName"
    ],
    "type": "object",
    "properties": {
        "userName": {
            "maxLength": 256,
            "minLength": 0,
            "type": "string"
        },
        "name": {
            "maxLength": 64,
            "minLength": 0,
            "type": "string"
        },
        "surname": {
            "maxLength": 64,
            "minLength": 0,
            "type": "string"
        },
        "emailAddress": {
            "maxLength": 256,
            "minLength": 0,
            "type": "string",
            "format": "email"
        },
        "isActive": {
            "type": "boolean"
        }
    },
},

```



```

    "fullName": {
      "type": "string",
      "nullable": true
    },
    "lastLoginTime": {
      "type": "string",
      "format": "date-time",
      "nullable": true
    },
    "creationTime": {
      "type": "string",
      "format": "date-time"
    },
    "description": {
      "maxLength": 4096,
      "type": "string",
      "nullable": true
    },
    "profilePicture": {
      "maxLength": 16000,
      "type": "string",
      "format": "byte",
      "nullable": true
    },
    "roleNames": {
      "type": "array",
      "items": {
        "type": "string"
      },
      "nullable": true
    },
    "id": {
      "type": "integer",
      "format": "int64"
    }
  },
  "additionalProperties": false
},
"Int64EntityDto": {
  "type": "object",
  "properties": {
    "id": {
      "type": "integer",
      "format": "int64"
    }
  },
  "additionalProperties": false
},
"RoleDtoListResultDto": {
  "type": "object",
  "properties": {
    "items": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/RoleDto"
      },
      "nullable": true
    }
  },
  "additionalProperties": false
},

```

```

"ChangeUserLanguageDto": {
  "required": [
    "languageName"
  ],
  "type": "object",
  "properties": {
    "languageName": {
      "type": "string"
    }
  },
  "additionalProperties": false
},
"ChangePasswordDto": {
  "required": [
    "currentPassword",
    "newPassword"
  ],
  "type": "object",
  "properties": {
    "currentPassword": {
      "type": "string"
    },
    "newPassword": {
      "type": "string"
    }
  },
  "additionalProperties": false
},
"ResetPasswordDto": {
  "required": [
    "adminPassword",
    "newPassword",
    "userId"
  ],
  "type": "object",
  "properties": {
    "adminPassword": {
      "type": "string"
    },
    "userId": {
      "type": "integer",
      "format": "int64"
    },
    "newPassword": {
      "type": "string"
    }
  },
  "additionalProperties": false
},
"UserDtoPagedResultDto": {
  "type": "object",
  "properties": {
    "totalCount": {
      "type": "integer",
      "format": "int32"
    },
    "items": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/UserDto"
      }
    }
  }
}

```

```
        "nullable": true
    }
},
"additionalProperties": false
}
},
"securitySchemes": {
    "bearerAuth": {
        "type": "apiKey",
        "description": "JWT Authorization header using the Bearer
scheme. Example: \"Authorization: Bearer {token}\"",
        "name": "Authorization",
        "in": "header"
    }
}
}
}
```