

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Aldin Talve 134859IAPB

Ehitushangete otsimise veebirakendus

Bakalaureusetöö

Juhendaja: Gert Kanter
Doktorikraad

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: Aldin Talve

25.05.2021

Annotatsioon

Käesoleva lõputöö eesmärk on luua veebirakendus ehitushangete otsimise teenuse osutamiseks. Rakendus kuvab kasutaja jaoks relevantseid ehitusteatised ja laseb nende põhjal sooritada otsinguid, mis lihtsustavad sobiva hanke leidmist. Samuti on võimalik otsida hangetega seotud ehitisi ja sedakaudu otsida hankeid kindlast piirkonnast.

Rakenduses on võimalik märkida teatised ja ehitisi jälgitavateks, mis aitab nendega seotud teatiste kohta teavet saada. Kasutaja poolt otsitavad andmed on piiratud temaga seotud firmade profiilidega.

Projekt on tehtud koostöös Merzell Estonia OÜ-ga, kus autor selle tegemise ajal töötab ja sellega seoses peab teostus vastama ka teatud firmapoolsetele nõuetele. Kõik andmed saadetakse rakenduse serverisse firma kesksest serverist.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 6 peatükki, 23 joonist.

Abstract

Web application for searching construction tenders

The goal of this thesis is to accompany the creation of a web application for providing the service of searching building tenders. The application shows building notices relevant to the specific user and provides a search function in order to facilitate finding a tender of interest. It is also possible to search for related buildings in order to find tenders in a specified area.

The user can mark interesting notices and buildings as favorites in order to facilitate finding them and receiving updates about them in the future. Each user is tied to a company which has one or more profiles limiting what type of notices the user can search for and see.

There are 3 main views for searching, allowing searches for tenders, buildings and tenders split into profiles. The building and notice views point out related entities of the other type.

The application is made in collaboration with Merzell Estonia OÜ, where the author is employed at the time and as such it has to comply with certain stipulations set by the company. The data is received from the central server of said company. User and data management is in large part realized through secured JSON endpoints.

The thesis is in Estonian and contains 25 pages of text, 6 chapters, 23 figures.

Lühendite ja mõistete sõnastik

AJAX	<i>Asynchronous JavaScript And XML</i> Asünkroonne päring serverisse üle Javascripti
CPV	<i>Common Procurement Vocabulary</i> , hangete klassifitseerimise kood [1]
CSS	<i>Cascading Style Sheets</i> , veebilehtede stiilimiseks kasutatav märgistuskeel
DTO	<i>data transfer object</i> , olem andmete kandmiseks eri protsesside vahel
HTML	<i>Hypertext Markup Language</i> , veebilehtede loomiseks kasutatav standartne märgistuskeel
IDE	<i>Integrated Development Environment</i> , integreeritud arenduskeskkond
JSON	<i>JavaScript Object Notation</i> , kompaktne andmevahetusvorming
LINQ	<i>Language Integrated Query</i> , .NET raamistiku komponent andmebaasipäringute tegemiseks
NUTS	<i>Nomenclature of Territorial Units for Statistics</i> , hierarhiline süsteem Euroopa majandusala jaotamiseks [2]
PHP	PHP: <i>Hypertext Preprocessor</i> , skriptimiskeel dünaamiliste veebilehtede loomiseks
SQL	<i>Structured Query Language</i> , andmebaasiga manipuleeriiseks kasutatav keel

Sisukord

Autorideklaratsioon	2
Annotatsioon.....	3
Abstract Web application for searching construction tenders	4
Lühendite ja mõistete sõnastik	5
Sisukord.....	6
Jooniste loetelu	8
1 Sissejuhatus	9
2 Töö ülevaade.....	10
2.1 Firma taustainfo	10
2.2 Ülesandepüstitus	10
2.3 Lahenduse nõuded	11
2.4 Kasutatavad vahendid.....	12
3 Planeerimine ja struktuur.....	14
3.1 Projekti ülesehitus.....	14
3.2 Andmebaasi skeemi koostamine	14
3.3 Vaadete planeerimine	15
3.4 Visuaalne ja UX kujundus	15
3.5 Turvalisus	17
4 Arendusprotsess.....	18
4.1 Andmebaasi üles seadmine.....	18
4.2 Andmebaasi andmete kandmine	18
4.3 Andmebaasipäringud	20
4.4 Kontrollerid	23
4.5 Sisselogimine ja autentimine	23
4.6 Kasutajate registreerimine	23
4.7 Sisuvaated.....	24
4.8 Lisafunktionaalsus	27
4.9 Kujundus.....	29

5 Tulemused	32
5.1 Realiseeritud lahendused	32
5.2 Ilmnenud puudused.....	32
5.3 Saadud kogemused	32
5.4 Võimalikud edasiarendused.....	33
6 Kokkuvõte	34
Kasutatud kirjandus	35
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	36

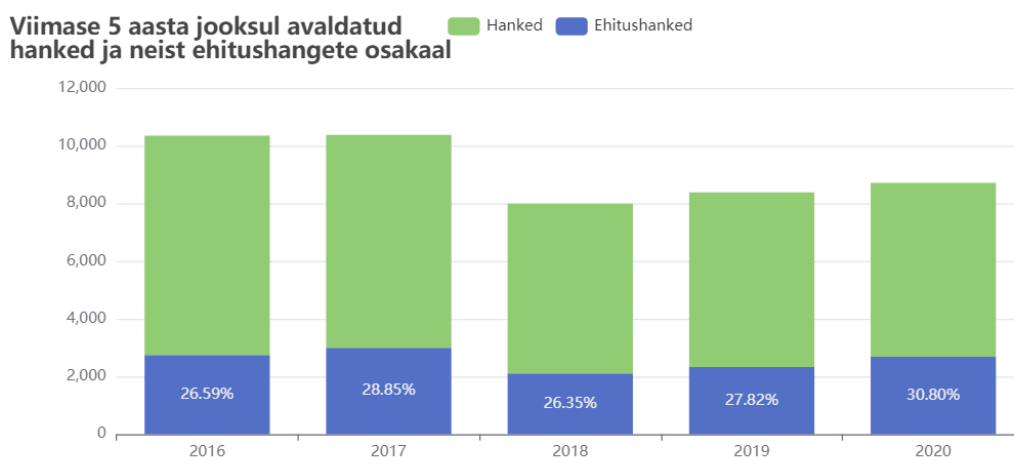
Jooniste loetelu

Joonis 1. Ehituhangete osakaal 2016-2020.....	9
Joonis 2. Olemi Company mudel rakenduses.....	12
Joonis 3. Esimeste sprintide ülevaade Trellos.	13
Joonia 4. Rakenduse arhitektuur.....	14
Joonis 5. Meiliteavituse kujundus.	16
Joonis 6. Andmebaasi skeem.....	18
Joonis 7. Osaline näide teatise kohta tulevast infost.	19
Joonis 8. Näide teatise sisselugemisel kasutatavatest annotatsioonidest.....	20
Joonis 9. Andmebaasipäring ja vaatemudeli koostamine teatise vaate eeskujul.	21
Joonis 10. Firma vahetus ja muudatuse salvestamine sessiooni.....	22
Joonis 11. Parameetrite arvust sõltuv päring.....	22
Joonis 12. Testitav versioon kasutaja loomise JSONist.....	24
Joonis 13. Kontrolleri poolne osa teatise lemmikuks lisamiseks.....	24
Joonis 14. AJAX päringu osad teatise lemmikuks lisamiseks.	25
Joonis 15. Teatiste tabeli vaade.	25
Joonis 16. Profiili järgi otsingu vaade.	26
Joonis 17. Ehitise vaade.	27
Joonis 18a. Esileht eesti keeles.....	28
Joonis 18b. Esileht inglise keeles.	28
Joonis 19. Modal asukoha näitamiseks kaardil.	29
Joonis 20a. Päise menüü kujundus <i>desktop</i> vaates.	30
Joonis 20b. Päise menüü kujundus telefonivaates.....	30
Joonis 21. Rakenduse stiiliraamat.....	31

1 Sissejuhatus

Antud lõputöö eesmärk on luua koostöös Merzell Estonia OÜ-ga ehituhangete otsimiseks rakendus, mille abil kasutaja saab eelfiltreeritud andmehulgast leida teda huvitavaid hankeid. Hange kujutab endat taotlust mingi toote või teenuse saamiseks ja laseb antud toote või teenuse pakkujal essitada enda poolne hind. Antud rakendus aitab teenuse pakkujal kergemini välja filtreerida ainult teda huvitavad ehitushanked, võimaldades nii hoida kokku aega ja kulutusi hangete otsimise pealt ja vähendades riski, et hea hange jääb märkamata.

Enamus hankeid tuleb süsteemi riigihangete registrist. Nende kodulehelt [3] avalikult kättesaadava statistika põhjal oli 2019. ja 2020. aastal riigihangete registriss vastavalt 8381 ja 8706 hanget, neist ehitushankeid vastavalt 2332 ja 2682 ehk suurusjärgus 30%. Hangete algne info ja täpsustused tulevad teatistega, mida võib ühe hanke kohta olla mitu. Teatis saab lisada ka olemasolevale hankele hankele uusi osasid, mis võivad muuta antud hankest huvitatud kontingenti. Seega on rakenduse põhifookus uute teatiste edastamisel kasutajale, mis annab reaajas ülevaate nii uute hangete kui ka vanade hangete muudatuste kohta. Joonisel 1 on toodud ehitushangete osakaal kõigist hangetest viimase 5 aasta lõikes.



Joonis 1. Ehituhangete osakaal 2016-2020.

2 Töö ülevaade

2.1 Firma taustainfo

Mercell on Taanis baseeruv rahvusvaheline firma, mis põhilielt pakub klientidele hangete haldamise teenust. Seega kliendi rollis on üldjuhul hangetel pakkumisi tegevad firmad, kuid Mercell võimaldab oma süsteemis ka hankeid korraldada [4]. Firma harud on paljudes Euroopa riikides, sealhulgas Eestis, Lätis ja Leedus. Baltikumi harud teevad omavahel tihedalt koostööd ja põhiline arendustiim asub Eestis.

Praeguse seisuga pakub Mercell siin regioonis hangete otsimise teenust üle meili ja töötajad annavad klientidele edasi hankeid nende profiili ja varasemate hangetel kandideerimiste põhjal. Kliendi makstav kuutasu sõltub tema tellitud profiili ulatusest CPV (*Common Procurement Vocabulary*) koodide lõikes.

2.2 Ülesandepüstitus

Loodava rakenduse eesmärk on pakkuda keskkonda, kus klient saaks ise oma profiili ulatuses ehitushangete otsinguid sooritada. Nii saaks seda pakkuda klientidele lisateenusena ja samas aitaks see vähendada Mercelli töötajate koormust, kui suuremad kliendid on huvitatud ise hangete otsimisest ja määravad endal selle ülesande firmasiseselt.

Mercell Estonia keskses serveris on ehitushanked segi muude hangetega, seega üle aastate on kogunenud väga suur andmehulk ja Mercelli töötajad teevad selle peal oma tööülesandeid täites tihti ressursimahukaid päringuid. Seega on mõistlik loodava rakenduse andmebaas panna eraldi serverisse, kus on seda ka kergem skaleerida olenevalt teenuse populaarsusest. Lisaks on see variant parem ka turvalisuse kaalutlusel.

Arvestades, et enamus andmeid on pärit riigihangete registrist ja neil on juba olemas põhjalik otsimisfunktsioon mis võimaldab otsida suure osa antud rakenduse raames plaaneeritud otsinguparameetrite järgi, peab rakendusel olema mingi lisaväärtus. Siinkohal saab tänu ainult ehitushangetele fokuseerumisele välja tuua ehitistepiirkonna ja

aadressi järgi teatiste otsimise võimaluse ja kasutaja profiilide põhjal ainult teda huvitada võivate teatiste kuvamine.

2.3 Lahenduse nõuded

Loodav rakendus peab lõppversioonis olema kasutatav üle Baltikumi ja seega arvestama kõigi Balti regioonide eripäradega. Näiteks Eesti hangete puhul tulevad hanke osad üldjuhul eri teatistega, aga mujal Baltikumis võib ühe teatisega tulla mitu osa. Samuti tuleneb sellest vajadus veebilehe tõlkimise järele. Kuna Mercell on rahvusvaheline firma ja suhtlus toimub suuresti inglise keeles, on mõistlik teha nii koodibaas kui lehe algne tõlge inglise keeles ja seega on ka kaasnevad illustratsioonid ja koodinäited valdavalt ingliskeelsed.

Andmebaas hakkab sisaldama ainult ehitushankeid, mis saadetakse rakenduse andmebaasi Mercell Estonia kesksest andmebaasist JSON (*JavaScript Object Notation*) faili kujul ja rakenduse liides loeb saadud andmed enda andmebaasi. Põhiolemite puhul peab säilitama seose keskse andmebaasi olemitega unikaalse andmevälja 'PesaId' kaudu. See aitab ka vältida andmete duplitseerimist ja on kasuks andmete uuendamisel.

Rakenduse kasutajal peab olema ligipääs ainult oma firma profiiliga seotud andmetele ja süsteem peaks võimaldama ühele kasutajale määrata mitu firmat, mida peaks saama päisest vabalt vahetada. Kasutajate registreerimise ja ligipääsu haldamisega peaks enamjaolt suutma tegeleda müügiga tegelev töötaja ise, kasutades Mercell Estonia kesksesse süsteemi arendatavat liidest.

Eeldatava andmebaasipakkuja tõttu peaks andmebaasi tegema võimalikult vähe päringuid iga vaate kohta ja võimalusel andmeid meeles hoidma. Sealsed teenustasud on põhiliselt päringute arvu, mitte andmemahu järgi. Samas võib liiga paljute andmete korraka saatmine aeglustada vaadete kuvamise kiirust ning teatud juhtudel võib päringute arvu liigne optimeerimine suurendada süsteemi keerukust ebamõistlikus ulatuses.

Tasustamise osas on kavas kasutada olemasolevat süsteemi, kus ligipääsetavate andmete maht sõltub kasutaja tellitud profiilidest. Täpsemad detailid pannakse paika koostöös müügiosakonnaga rakenduse arendamise hilisemas faasis.

2.4 Kasutatavad vahendid

Rakenduse loomisel kasutab autor Microsoft Visual Studiot, mis on firmas põhiliselt kasutuses olev IDE (*Integrated Development Environment*) ja C# keelt. Antud vahendid on ka igati sobivad projekti efektiivseks realiseerimiseks täies mahus. Visual Studiole on lisatud ReSharper laiendus, mille funktsioon on mitmes aspektis koodi kirjutamise produktiivsuse suurendamine.

Projekt baseerub ASP.NET Core [5] ja Entity Framework Core raamistikel. Merzell Estonia praegune keskne süsteem on üles ehitatud ASP.NET raamistiku varasema versiooni peale ja on seega mõneti analoogne loodava rakenduse ülesehitusega, mis hõlbustab kaastöölistel hiljem koodi mõistmist ja vajadusel redigeerimist.

Andmebaas põhineb Microsoft SQL (*Structured Query Language*) andmebaasihalduril ja genereeritakse Entity Framework Core abil *code-first* põhimõttel olemi mudelist (joonis 2). Serveri hostimise teenus on projekti valmimisel kavas väljast tellida.

```
public class Company
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    [JsonProperty("localId")]
    public int CompanyId { get; set; }

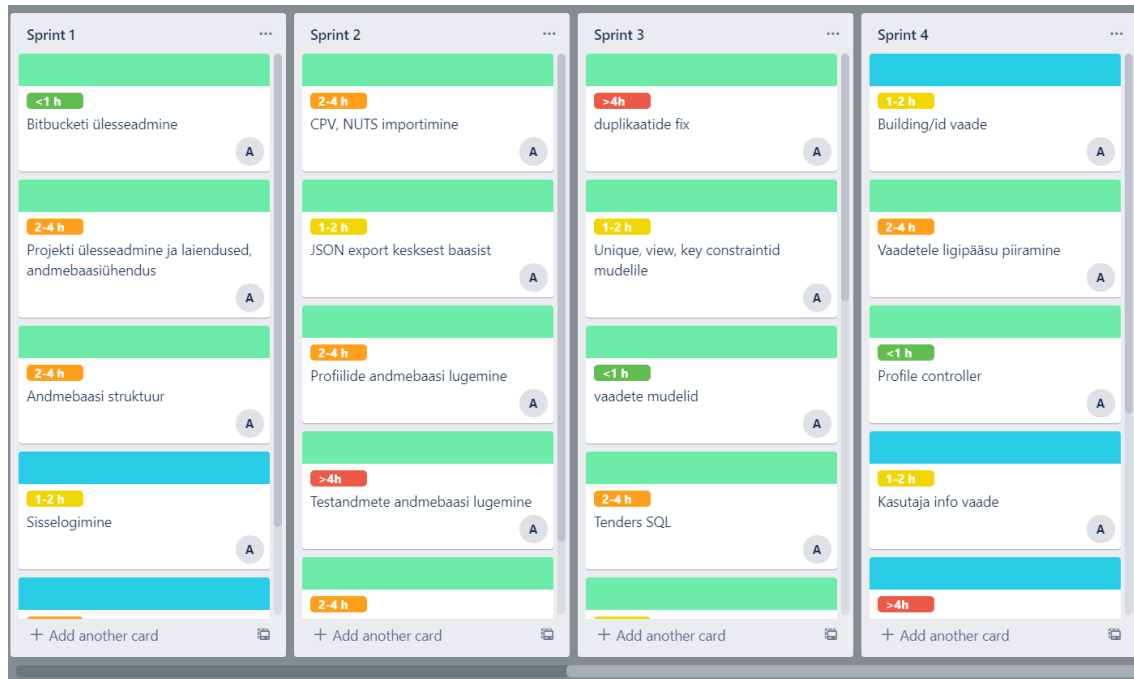
    [JsonProperty("companyId")]
    public int PesaId { get; set; }
    [JsonProperty("name")]
    public string Name { get; set; }
    [JsonProperty("country")]
    public string Country { get; set; }
    [JsonProperty("registrationNumber")]
    public string RegistrationNumber { get; set; }

    public ICollection<UserCompany> UserCompany { get; set; }
    public ICollection<CompanyProfile> CompanyProfile { get; set; }
    public ICollection<CompanyPackage> CompanyPackage { get; set; }
    public ICollection<RelatedParty> RelatedParty { get; set; }
}
```

Joonis 2. Olemi Company mudel rakenduses.

Front-end liidese loomisel on koos C#-ga kasutusel HTML (*Hypertext Markup Language*), JavaScript, JQuery [6] ja CSS (*Cascading Style Sheets*). Lisaks on kujundamisel kasutatud laiendusi Bootstrap [7] ja PopperJS [8].

Versioonihaldus käib läbi Git versioonihaldussüsteemi kasutades Bitbucketi [9] teenust, projekti planeerimine ühe nädala pikkute sprintide kaupa toimub Trello [10] keskkonnas (joonis 3).

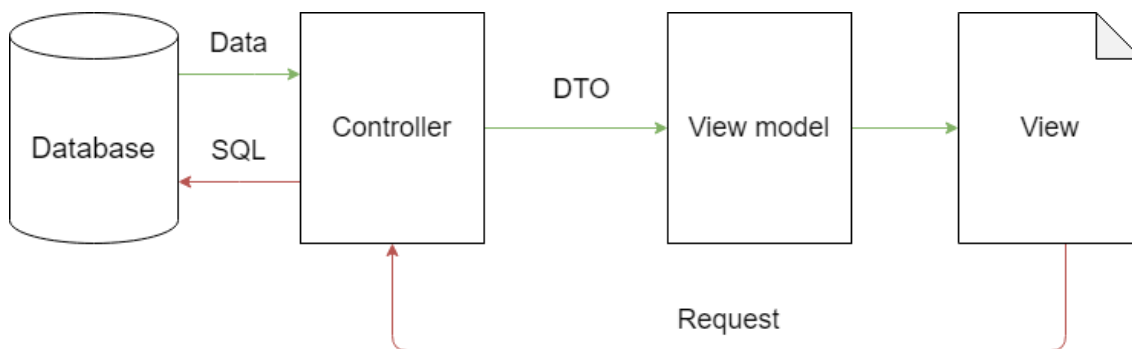


Joonis 3. Esimeste sprintide ülevaade Trellos.

3 Planeerimine ja struktuur

3.1 Projekti ülesehitus

Rakenduse ülesehitus järgib MVC (*Model–View–Controller*) arhitektuuri põhimõtteid, andmebaasiga suhtlevad vaid kontrollid ja nende päritud andmetest kootatakse vaatemudel. Lõplik vaade pannakse kokku vaatemudelist tulnud info põhjal. Andmete transpordiks eri kihtide vahel on kasutuses DTO (*Data Transfer Object*) muster. Joonisel 4 on toodud lihtsustatud ülevaade rakenduse arhitektuurist.



Joonia 4. Rakenduse arhitektuur.

Kasutajate haldamine toimub ASP.NET Core komponendi 'Identity' abil, mis võimaldab kasutajate registreerimist, autentimist ja muid seotud toiminguid.

3.2 Andmebaasi skeemi koostamine

Andmebaasi struktuur põhineb suures osas Merzell Estonia kesksel andmebaasil, koos mõningate muudatustega tulenevalt ehitushangetega mitte seotud väljade ja kliendile mittevajaliku info välja jätmisest. Põhiolemitele on ka lisatud atribuut 'PesaId', mis hoiab olemi unikaalset identifikaatorit Merzelli kesksest andmebaasis. Tabelid, mis sisaldavad staatilisi andmeid, nagu CPV ja NUTS (*Nomenclature of Territorial Units for Statistics*) koodid, saab kesksest andmebaasist otse importida.

3.3 Vaadete planeerimine

Rakenduse põhifunktsiooni täitmiseks oli algselt plaanitud 5 põhivaadet:

- Esileht, mis annab kiire ülevaate viimasest potentsiaalselt huvi pakkuvast informatsioonist
- Teatiste nimekiri mis järjestab kronoloogiliselt nimekirja aktiivsetest hanketeatistest, kus toodud nende kohta olulisem info ja laiendamisnappu vajutades antud ka ülevaade teatise osadest
- Ehitiste nimekiri, kus saab aadressi järgi otsida ehitisi ja mis annab põgusa ülevaade ehitise atribuutidest.
- Teatise vaade, kus kuvatakse kogu teave teatise kohta ja link teatisega seotud ehitistele
- Ehitise vaade, kus kuvatakse kogu teave ehitise kohta ja lingid temaga seotud teatistele.

Lisaks ülaltoodule lisandus hiljem nõudena profiilipõhine teatiste nimekirja vaade, kus kasutaja saab valida suvalise hulga oma firmaga seotud profiile ja saab uusimad teatised ainult valitud profiilide lõikes.

Plaaneeritud on ka põhifunktsionaalsusega otseselt mitteseotud vaated nagu kasutaja seadete vaade ja privaatsusvaade.

3.4 Visuaalne ja UX kujundus

Visuaalse järjepidevuse mõttes võiks rakenduse kujundus ja värviskeem üldjoontes vastata firma värviskeemile ja olema kliendile ära tuntavad. Seega oleks hea võtta eeskuju firma kodulehest ja meili teel saadetavate ülevaadete kujundustest. Joonisel 5 on toodud Figma disainirakenduses tehtud meiliteavituste prototüüp [11], millest on plaanis kujundamisel eeskuju võtta.









21.04.2020

Tere, Priit Turva! Siin on tänased profiilile vastavad hanked-, plaanid ja aruanded

Kuna tegu on automaatse teavitusega, mis tuleb noreply-aadressilt, palume sellele mitte vastata. Küsimuste või tõrgete korral võta ühendust oma konsultandiga. Kontaktinfo on toodud kirja lõpus.

Profiil

SS.0 IT teenused: nõuande-, tarkvaraarendus-, interneti- ja tugiteenused (Eesti)

NIMETUS	TÜÜP	REGIOON	LIIK
 Maardu Rimi välireklaamid Rimi Eesti Food AS · Tähtaeg: 24.04.2020 · 		Helsinki-Uusimaa, FI	Hange
Toetuste adimnistreerimise ja kontrolli süsteemi (TAKS) arendus- ja hooldustööde tellimine Põllumajanduse Registrate ja Informatsiooni Amet · Tähtaeg: 20.06.2020 · 		Eesti, EE	Hange
MS Dynamics tootepere hooldus ja arendus 2019-2020 Ettevõtluse Arenduse Sihtasutus · Tähtaeg: 27.07.2020 · 		Eesti, EE	Hange
Toetuste adimnistreerimise ja kontrolli süsteemi (TAKS) arendus- ja hooldustööde tellimine		Eesti, EE	Hange

Joonis 5. Meiliteavituse kujundus.

Päis peab sisaldama firma logo ja vastavalt nõuetele võimalust vahetada rakenduse keelt ja kasutaja aktiivset firmat. Viimaste muutmisel peaks leht end värskendama ja kuvama asjakohaselt uuendatud infot. Sisselogimisel vaikimisi valitud keel ja firma peaks olema kliendi registreerimisel määratav, vastasel juhul oleks need vastavalt inglise keel ja kronoloogiliselt esimene kliendiga seotud firma.

Rakendust peaks olema võimalik mugavalt kasutada ka mobiili ja tahvelarvuti peal, kuid põhirõhk on kasutatavusel arvutis. Seda johtuvalt asjaolust, et peamine kasutusjuht Mercelli klientide tausta arvestades on rakenduse kasutamine lauaarvutil.

3.5 Turvalisus

Klientide täpsemaid isikuandmeid pole turvakaalutlustel mõtet rakenduse serveris eraldi hoida. See tooks kaasa kõrgendatud turvariske ja tulenevalt rakenduse funktsionaalsusest ei ole see vajalik. Vajadusel saab tuge pakkuv töötaja need andmed kätte Merzell Estonia kesksest serverist, kus kõik selle rakenduse kasutajate vajalikud andmed on sisse kantud.

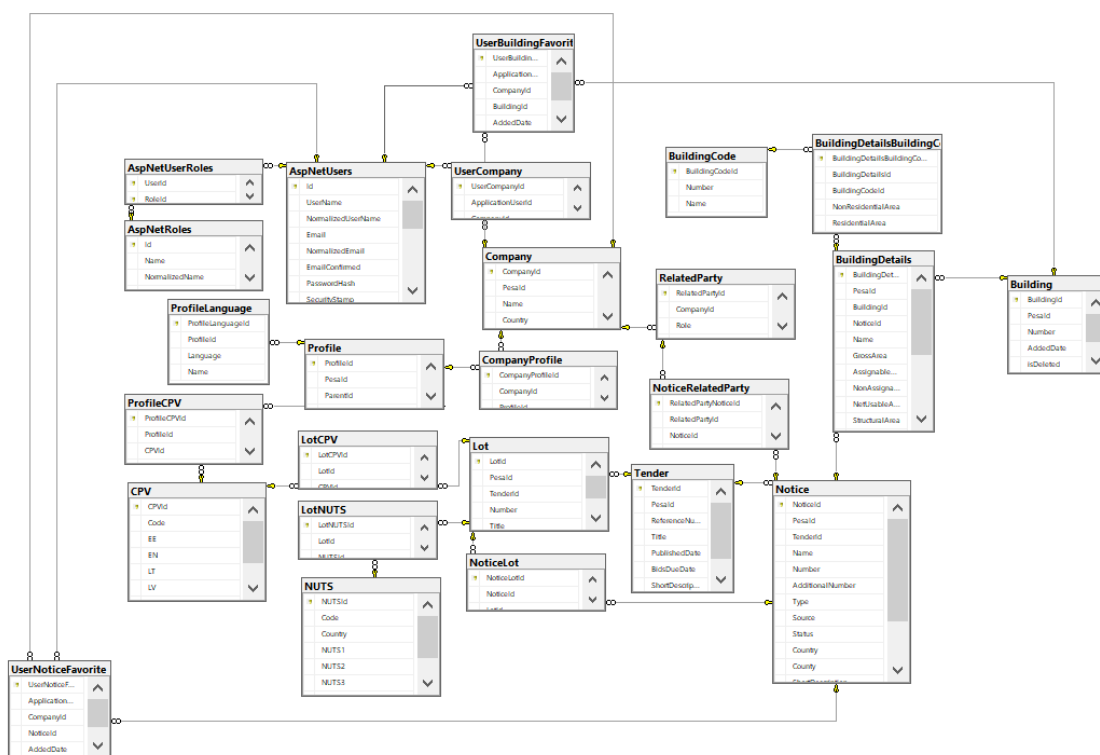
Firma kesksesse serverisse peaks suunduma võimalikult vähe päringuid, seega enamuse andmete jagamise rakenduse *endpointi* algatab keskne server ise. *Endpoint* ise peaks päringuid vastu võttes kontrollima IP aadressi kattuvust.

4 Arendusprotsess

4.1 Andmebaasi üles seadmine

Andmebaas genereeritakse olemeid kirjeldavate mudelite põhjal kasutades laiendust Entity Framework. Olemite konstrueerimisel võetakse aluseks keskse serveri olemid ja jäetakse välja rakenduse eesmärgiga mitte seonduvad osad. Andmebaasi konteksti failis on määratud ka unikaalsed väljad ja väljade kombinatsioonid andmete duplitseerimise vältimieks. Andmebaasi lõplik skeem on toodud joonisel 6.

Staatilised tabelid nagu CPV ja NUTS on imporditud Merzell Estonia serverist kasutades SQL Server Management Studiot. On oluline, et antud tabelite andmed kattuksid serverite vahel.



Joonis 6. Andmebaasi skeem

4.2 Andmebaasi andmete kandmine

Siinkohal on aluseks võetud JSON fail joonisel 7 toodud kujul. Andmete migreerimiseks kasutan JSON kaardistamist, mis tõlgib sissetulevad andmed JSON kujult automaatselt

olemi väljadeks Selleks on kasutatud Json.NET raamistikku ja mudeli väljad on vastavalt annoteeritud (joonis 8). Sissetuleva olemi identifikaatori välja väärtus kesksest andmebaasist kantakse vastavalt rakenduse nõuetele väljale 'PesaId'.

Rakendus hakkab hiljem regulaarselt saama koostatud *endpointi* uusi andmeid teatiste kohta, tõenäoliselt 1-2 korda päevas.

```
[
  {
    "noticeId": 14348,
    "name": "Koolihoone-ärihoone-elamu",
    "number": "1811002/10009",
    "additionalNumber": "113018970",
    "status": "Taotlus menetluses",
    "country": "EE",
    "county": "Harjumaa",
    "addedDate": "2020-06-17",
    "publishDate": "2020-06-15",
    "lastChangedDate": "2020-06-17",
    "type": 15,
    "source": 3,
    "threshold": 1,
    "noticeLot": [
      {
        "noticeId": 265118,
        "lotId": 307055,
        "lot": {
          "lotId": 307055,
          "number": "1",
          "title": "Koolihoone-ärihoone-elamu",
          "shortDescription": "Hoonevälise keldri kohal I ja II
            Korruse väljaehitamine",
          "lotCpv": [
            {
              "lotId": 307055,
              "cpvId": 1,
            },
            {
              "lotId": 307055,
              "cpvId": 2,
            },
            {
              "lotId": 307055,
              "cpvId": 3,
            }
          ]
        }
      }
    ]
  }
]
```

Joonis 7. Osaline näide teatise kohta tulevast infost.

```

public class Notice
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    [JsonIgnore]
    public int NoticeId { get; set; }
    [JsonProperty("noticeId")]
    public int PesaId { get; set; }

    public int? TenderId { get; set; }
    [JsonProperty("tender")]
    public Tender Tender { get; set; }

    [JsonProperty("name")]
    public string Name { get; set; }
    [JsonProperty("number")]
    public string Number { get; set; }
    [JsonProperty("additionalNumber")]
    public string AdditionalNumber { get; set; }

    [JsonProperty("type")]
    public int Type { get; set; }
    [JsonProperty("source")]
    public int Source { get; set; }
    [JsonProperty("status")]
    public string Status { get; set; }

    [JsonProperty("country")]
    public string Country { get; set; }
    [JsonProperty("county")]
    public string County { get; set; }
    [JsonProperty("shortDescription")]
    public string ShortDescription { get; set; }

    [JsonProperty("publishedDate")]
    public DateTime? PublishedDate { get; set; }
    [JsonProperty("addedDate")]
    public DateTime AddedDate { get; set; }
    [JsonProperty("lastChangedDate")]
    public DateTime? LastChangedDate { get; set; }

    [JsonProperty("noticeLot")]
    public ICollection<NoticeLot> NoticeLot { get; set; }
    [JsonProperty("relatedPartyNotice")]
    public ICollection<NoticeRelatedParty> NoticeRelatedParty { get;
        set; }
    [JsonProperty("buildingDetails")]
    public ICollection<BuildingDetails> BuildingDetails { get; set; }
}

```

Joonis 8.Näide teatise sisselugemisel kasutatavatest annotatsioonidest.

4.3 Andmebaasipäringud

Andmebaasipäringud on kirjutatud kasutades T-SQL (Transact-SQL) keelt, üritades järgida põhimõtet üks päring ühe vaate koostamiseks. Päritud andmed paigutatakse päringule vastavasse DTO-sse ja vajadusel jaotatakse abifunktsiooni kasutades eri

kihtidesse, peale mida viiakse need vaatemudelisse. Joonisel 9 on toodud teatise vaate jaoks vaatemudeli loomise kood rakenduse kontrolleriist.

Püsivamad andmed, nagu keele ja firma valik, salvestatakse pärast esimest päringut sessiooni muutujasse (joonis 10). Need andmed muutuvad väga harva ja seetõttu on peale sellekohaseid uuendusi välja ja sisse logimine marginaalse mõjuga rakenduse kasutamismugavusele.

Otsingute puhul on päringud vajadusel tehtud parameetrite arvu suhtes dünaamiliseks (joonis 11).

```
public NoticesNoticeVm GetNoticesNoticeVm(int id)
{
    var userId = _httpContext.HttpContext.User.
        FindFirst(ClaimTypes.NameIdentifier).Value;
    var selectedCompanyId = GetSelectedCompanyId(userId);
    var sql = $"
SELECT TOP(1) N.NoticeId, N.Name AS NoticeTitle, N.ShortDescription,
N.Source, N.PublishedDate, DATEADD(month, 6, N.PublishedDate) AS
BidsDueDate, STRING_AGG(L.LotId, '|') AS LotIdList,
STRING_AGG(CONVERT(nvarchar(max), ISNULL(L.Title, '')), '|') AS
LotTitleList,
STRING_AGG(CONVERT(nvarchar(max), ISNULL(L.ShortDescription, '')), '|')
AS LotDescriptionList, NF.IsFavoriteStr
FROM Notice AS N INNER JOIN NoticeLot AS NL ON N.NoticeId =
NL.NoticeId INNER JOIN Lot AS L ON NL.LotId = L.LotId
CROSS APPLY (VALUES (
    CASE WHEN EXISTS (
        SELECT TOP(1) UNF.NoticeId
        FROM (SELECT NoticeId FROM UserNoticeFavorite WHERE
ApplicationUserId = '{userId}' AND CompanyId = '{selectedCompanyId}')
AS UNF
        WHERE N.NoticeId = UNF.NoticeId)
        THEN 'true'
        ELSE 'false'
    END)) NF(IsFavoriteStr)
WHERE N.NoticeId = '{id}'
GROUP BY N.NoticeId, N.Name, N.ShortDescription, N.Source,
N.PublishedDate, NF.IsFavoriteStr
";
    var noticeRaw = _dbContext.NoticesNoticeRaw.
        FromSqlRaw(sql).FirstOrDefault();
    var notice = ConvertDTO.NoticeConvert(noticeRaw);
    var vm = new NoticesNoticeVm()
    {
        Notice = notice
    };
    return vm;
}
```

Joonis 9. Andmebaasipäring ja vaatemudeli koostamine teatise vaate eeskujul.

```

public IActionResult SwitchCompany(int id)
{
    Var userId = _httpContext.HttpContext.User.
        FindFirst(ClaimTypes.NameIdentifier).Value;
    if (_dbContext.UserCompany.Where(uc => uc.ApplicationUserId ==
        userId && uc.CompanyId == id).Count() > 0)
    {
        _httpContext.HttpContext.Session.SetString("SelectedCompany",
            id.ToString());
    }
    else return new JsonResult("Company not valid");
    return RedirectToAction("Index", "Home");
}

```

Joonis 10. Firma vahetus ja muudatuse salvestamine sessiooni.

```

SELECT X.ProfileId, X.ProfileTitle, N.NoticeId, N.Name AS NoticeTitle,
N.ShortDescription, N.Source, N.PublishedDate, DATEADD(month, 6,
N.PublishedDate) AS BidsDueDate, T.BidsDueDate, NF.IsFavorite,
STRING_AGG(L.LotId, '|') AS LotIdList,
STRING_AGG(CONVERT(nvarchar(max), ISNULL(L.Title, '')), '|') AS
LotTitleList,
STRING_AGG(CONVERT(nvarchar(max), ISNULL(L.ShortDescription, '')), '|')
AS LotDescriptionList
FROM Notice AS N INNER JOIN NoticeLot AS NL ON N.NoticeId =
NL.NoticeId INNER JOIN Lot AS L ON NL.LotId = L.LotId
RIGHT JOIN (
    SELECT NL.NoticeId, P.ProfileId, P.Number, PL.Name AS ProfileTitle
    FROM CompanyProfile AS CP
    INNER JOIN Profile AS P ON CP.ProfileId = P.ProfileId
    INNER JOIN ProfileLanguage AS PL ON P.ProfileId = PL.ProfileId AND
    PL.Language = '{selectedLanguage}'
    INNER JOIN ProfileCPV AS PCPV ON P.ProfileId = PCPV.ProfileId
    INNER JOIN LotCPV AS LCPV ON PCPV.CPVID = LCPV.CPVID
    INNER JOIN NoticeLot AS NL ON LCPV.LotId = NL.LotId
    WHERE CP.CompanyId = '{selectedCompanyId}'
) AS X ON X.NoticeId = N.NoticeId
CROSS APPLY (VALUES(
    CASE WHEN EXISTS(
        SELECT TOP(1) N.NoticeId
        FROM (
            SELECT NoticeId FROM UserNoticeFavorite
            WHERE ApplicationUserId = '{userId}'
            AND CompanyId = '{selectedCompanyId}') AS UNF
        WHERE N.NoticeId = UNF.NoticeId)
        THEN CAST(1 AS BIT)
        ELSE CAST(0 AS BIT)
    END)) NF(IsFavorite)
WHERE 1=1
{keyword != null ? $"
AND N.Name LIKE '%{keyword}%'
OR L.ShortDescription LIKE '%{keyword}%' : ""}
{searchFrom != null ? $"
AND N.PublishedDate >= '{searchFrom}' : ""}
{searchTo != null ? $"
AND N.PublishedDate <= '{searchTo}' : ""}
GROUP BY X.ProfileId, X.ProfileTitle, N.NoticeId, N.Name,
N.ShortDescription, N.Source, N.PublishedDate, NF.IsFavorite

```

Joonis 11. Parameetrite arvust sõltuv päring

4.4 Kontrollerid

Kontrollerid on jaotatud vastavalt oma funktsioonile avalehe, teatiste, ehitiste, profiili ja lokaalseks kontrolleriks. Täpsem jaotus on järgmine:

- Avalehe kontrolle (Home controller) tegeleb esilehele info pärmisega, praeguse seisuga pärib lemmikteatiseid ja -ehitised.
- Teatiste kontroller (Notices controller) tegeleb teatiste kohta info pärmisega ja lemmikteatiste lisamisega.
- Ehitiste kontroller (Buildings controller) tegeleb ehitiste kohta info pärmisega ja lemmikehitiste lisamisega.
- Profiili kontroller (Profile controller) tegeleb kasutaja andmete pärmise ja uuendamisega.
- Lokaalne kontroller (Local controller) tegeleb põhiserverist sissetulevate andmete töötlemisega (e.g. uute teatiste ja loodavate kasutajate info).

4.5 Sisselogimine ja autentimine

Kasutatav raamistik võimaldab kindlat lehte eri kujul kuvada olenevalt kas külastaja on sisse loginud või mitte. Mitteautenditud külastaja saadetakse sisselogimise lehele, kus saab meiliaadressi ja parooli kasutades sisse logida.

Kasutajal on kasutaja seadete lehel võimalik oma parooli muuta. Parooli unustamisel on sisse logimise vaates võimalik parool lähtestada. Mõlemad võimalused põhinevad ASP.NET Core Identity funktsionaalsusel.

4.6 Kasutajate registreerimine

Uue kasutaja registreerimine käib Mercelli töötaja poolt kesksest serverist saadetud JSON päringuga. Selle päringu koostamise loogika saab olema Mercelli keskses süsteemis, aga prototüübitud verioon on toodud joonisel 12. Päringuga peab kaasnema loodava kasutaja meiliaadress ja vähemalt üks temaga seotud firma. Samuti tuleb kaasa loodava kasutaja

firma informatsioon ja seotud profiilid, mis kantakse antud informatsiooni puudumisel rakenduse andmebaasi.

Peale kasutaja registreerimist saadetakse talle kasutaja aktiveerimiseks SendGrid teenust ja laiendust kasutades meil. Meilis on link konto aktiveerimiseks ja kasutajale määratud ajutine parool.

```
[
  {
    "email": "renn@rennike.ee",
    "occupation": "klient",
    "companyList": [
      {
        "companyId": 9998,
        "name": "Dumbecon OÜ",
        "registrationNumber": "12382137"
        "country": "EE"
      }
    ]
  }
]
```

Joonis 12. Testitav versioon kasutaja loomise JSONist

4.7 Sisuvaated

Realiseeritavad sisuvaated koosnevad kolmest tabeli vaatest (teatiste tabel, ehitiste tabel ja profiilipõhine teatiste tabel) ja kahest olemi vaatest (teatis ja ehtis). Igas vaates on ka nupp leitud hanke või ehitise lemmikuks märkimiseks ja juba lemmikuks lisatud olemitel on see nupp märgitud kollaselt. Lemmikuks lisamine toimub üle Javacripti asünkroonselt AJAX(*Asynchronous JavaScript And XML*) päringuga. Lisamise õnnestumisel annab rakendus kasutajale sellealase teate. Antud protsessi illustreerivad joonised 13 ja 14.

```
public string AddFavoriteNotice(int id)
{
    var userId = _httpContextAccessor.HttpContext.User
        .FindFirst(ClaimTypes.NameIdentifier).Value;
    var selectedCompanyId = getSelectedCompanyId(userId);

    var unf = new UserNoticeFavorite()
    {
        ApplicationUserId = userId,
        CompanyId = selectedCompanyId,
        NoticeId = id,
        AddedDate = DateTime.UtcNow
    };
    _dbContext.UserNoticeFav.Add(unf);
    _dbContext.SaveChanges();
    return $"Added notice {id} to favorites";
}
```

Joonis 13. Kontrolleri poolne osa teatise lemmikuks lisamiseks


```

}
function ajaxPost(dest, data, async = true) {
  const xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) {
      alert(this.responseText);
    }
  };
  xhttp.open("POST", dest, async);
  xhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
  xhttp.send(data);
  return this.responseText;
}

function addFavoriteNotice(id) {
  ajaxPost("/Notices/AddFavoriteNotice/" + id);
  favButton = gid("favBtn_" + id);
  favButton.classList.add("fav");
  favButton.setAttribute("onclick",
    "javascript:removeFavoriteNotice(" + id + ")");
}

```

Joonis 14. AJAX päringu osad teatise lemmikuks lisamiseks.

Teatiste tabeli vaates (joonis 15) on välja toodud teatise pealkiri, allikas, kuupäev ja rea laiendamisel teatisega seotud hankeosad. Ehitiste tabeli vaates on ehitise aadress ja viimase ehitise kohta tulnud teatise kuupäev.

Mörcell
Home Buildings Notices Daily EN ▾ Mercell Testing OÜ ▾ Hello aldin.talve@hotmail.com! Log out

Search

From

To

Notices

Title	Source	Published	Bids due	Lots
★ üksikelamu	EHR	07/09/2020	07/09/2020	⤴
üksikelamu2011271/10885			Ehitise püstitamine	
★ rahvamaja	EHR	07/08/2020	07/08/2020	⤴
rahvamaja2011581/06332			Ehitise osa asendamine samaväärsega	
★ üksikelamu	EHR	07/08/2020	07/08/2020	⤵
★ Ärihoone Ringtee 73	EHR	07/08/2020	07/08/2020	⤵

Joonis 15. Teatiste tabeli vaade.

Profiili järgi teatiste tabelis on iga kasutaja valitud profiili kohta eraldi teatiste tabel ja profiili nimele klikkides saab antud profiili tulemusi peita. Kuna antud vaade lisandus

nõuete hulka hiljem, ei jõudnud seda peale põhifunktsionaalsuse korda saamist aga eriti viimistleda (joonis 16).

Search

Keyword

From

To

Selected 3/5

Päikesepaneelid, tuulegeneraatorid, taastuvenergeetika

Hoonete üldehitus

Title	Published	Lots
★ Vikerkaare 2	07/07/2020	▼
★ Katlamaja	07/07/2020	▼
★ üksikelamu	06/04/2020	▼
★ üksikelamu	06/05/2020	▼
★ Abihoone	07/08/2020	▼
★ Üksikelamu	07/06/2020	▼
★ Elamu	07/08/2020	▼

Joonis 16. Profiili järgi otsingu vaade.

Teatise vaates on teatise üldinfo, selle osade loend ja kronoloogilises järjestuses loend varasemate sama hanke kohta olnud teatistest. Ehitise vaates on uusim ehitise kohta saadav informatsioon ja ajalugu eelnevatest versioonidest. Ajaloos olevatel versioonidel on ka nupp informatsiooni võrdlemiseks kõige uuema versiooniga. Lisaks on olemas nupp ehitise asukoha näitamiseks kaardil. Ehitise vaade on toodud joonisel 17.

Building details

Address: Lääne-Viru maakond, Haljala vald, Käsmu küla, Laane tee 19

Latest details: 07/08/2020 12:18:11 AM

Show on map

Gross area: 0

Structural area: 123.5

Net usable area: 160.5

Assignable area: 155.3

Non-assignable area: 0

Ground floors: 2

Underground floors: N/A

Construction start date: N/A

Estimated completion date: N/A

Related parties

Majaka Arhitektid OÜ - Designer

Eraisik - ROLE_18_en

Eraisik - Applicant

Eraisik - Builder

Related notices

Elamu ★

Published on: 07/08/2020

Joonis 17. Ehitise vaade.

4.8 Lisafunktionaalsus

Rakenduse sisu tõlkimine toimub läbi abifunktsiooni, mis võtab parameetriteks tõlgitava stringi placeholder väärtuse ja soovitava tõlke keele ning tagastab tõlkefailist stringi väärtuse. Tõlke fail ise on JSON kujul, mis võimaldab seda kergelt redigeerida, tõlgete nimi on listina ja seega laiendatav. Vastava keele puudumisel võetakse vaikimisi ingliskeelne väärtus. Edasiarendusena oleks loodud tõlkesüsteemile lihtne lisada ka liides eri keeltes tõlgete lisamiseks. Esilehe eesti- ja ingliskeelsed tõlked on toodud joonistel 18a ja 18b.

Keelt ja ka firmat saab vahetada rakenduse päises olevatest rippmenüüdest. Uue väärtuse valimine jätab selle meelde ja värskendab lehte. Uued väärtused salvestatakse sessiooni muutujasse.

Ehitise vaade võimaldab korrektse aadressi puhul ka näha ehitise asukohta kaardil (joonis 19).

Otsi

Otsi ehitisi

Otsi teatiseid

Jälgitavad teatised

Pealkiri	Allikas	Avaldatud	Tähtaeg
Ei leidnud vasteid			

Jälgitavad ehitised

Address	Viimane uuendus
Ei leidnud vasteid	

© 2021 - Kroon21 - [Privaatsus](#)

Joonis 18a. Esileht eesti keeles.

Search

Search buildings

Search notices

Favorite notices

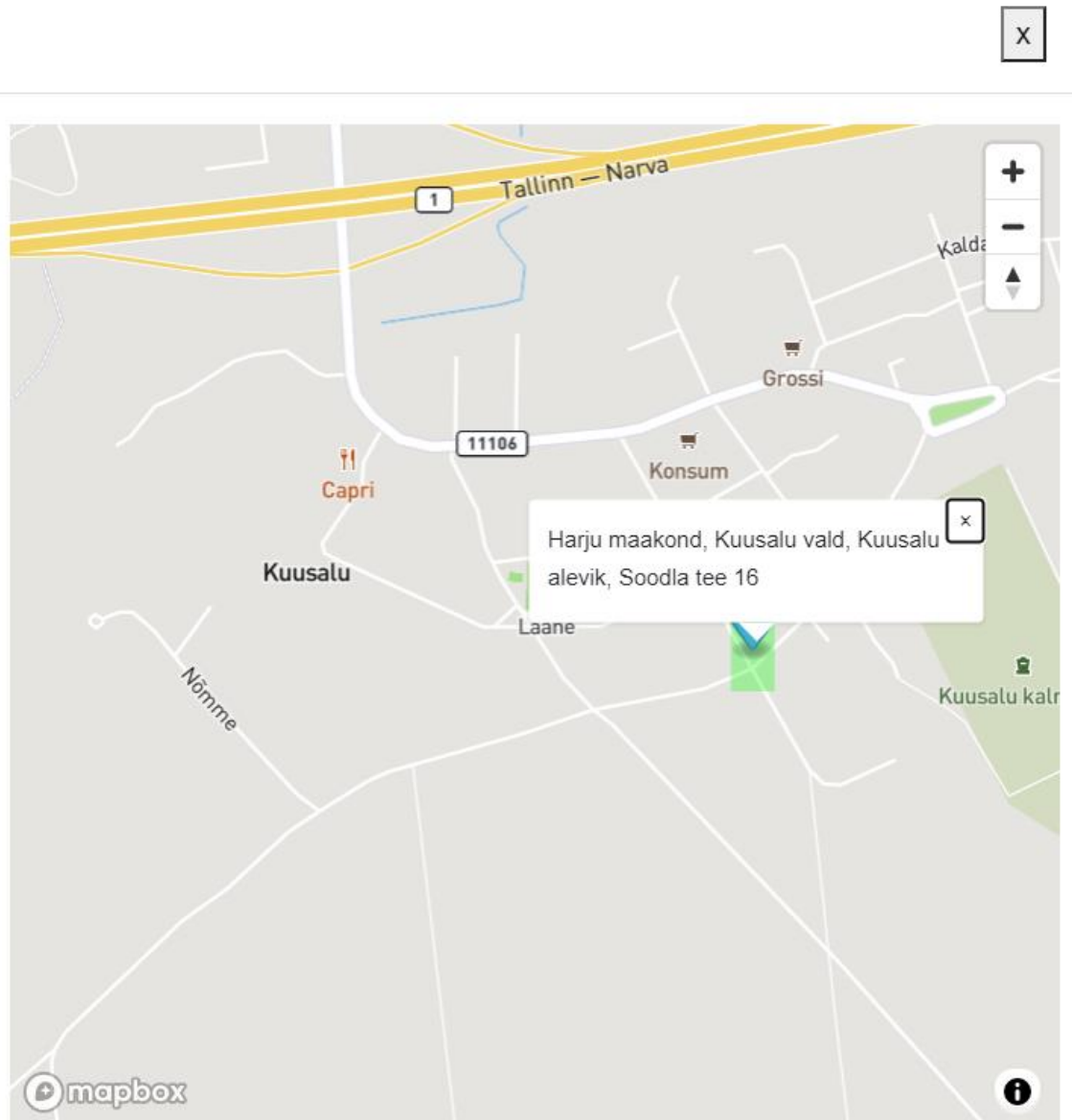
Title	Source	Published	Bids due
No results found			

Favorite buildings

Address	Last updated
No results found	

© 2021 - Kroon21 - [Privacy](#)

Joonis 18b. Esileht inglise keeles.



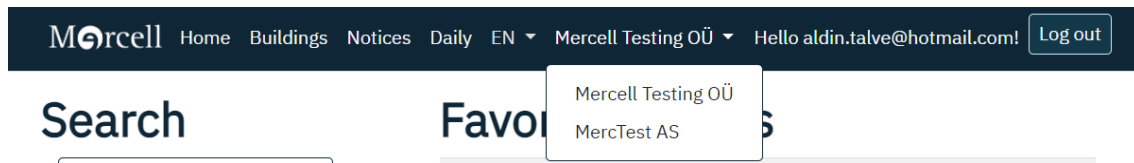
Joonis 19. Modal asukoha näitamiseks kaardil.

4.9 Kujundus

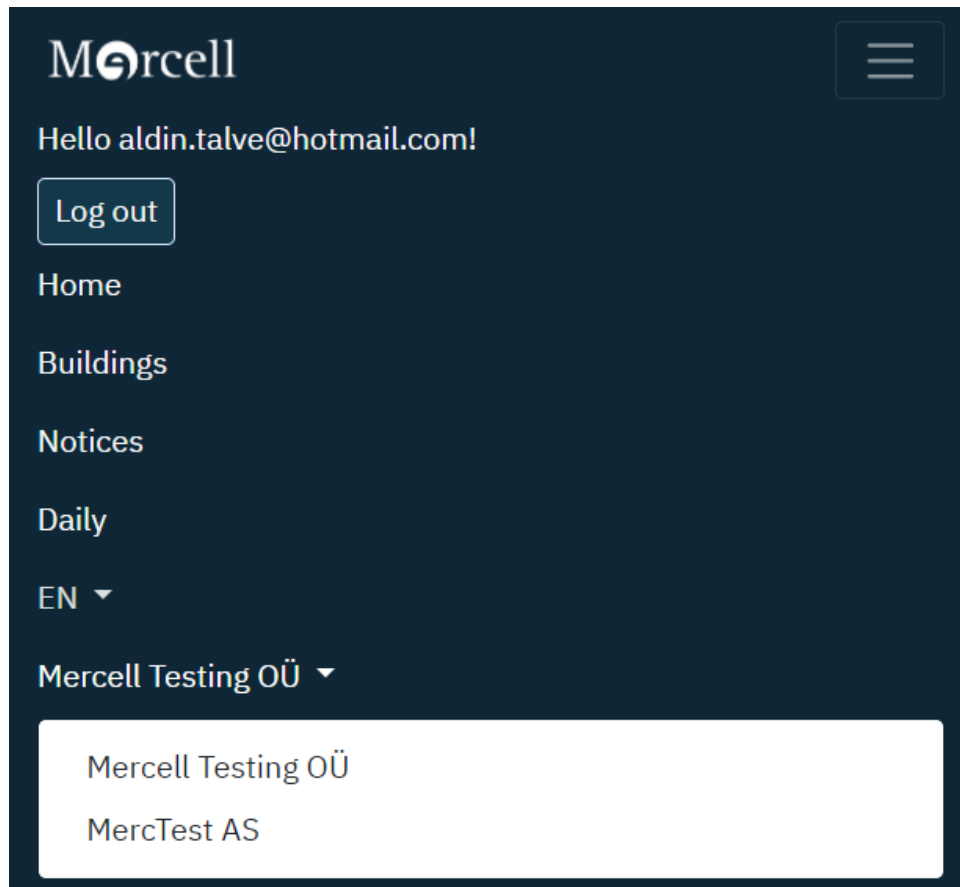
Kujunduse loomisel on võetud suuresti eeskujuna Merzell uudiskirja kujundusest [11]. Päises on toodud firma logo, keele valik ja firma valik. Nii päise kui ka vaadete kujundused reageerib ekraani suurusele ja on seeläbi kasutatav igas suuruses ekraaniga (joonised 20a ja 20b).

Kujunduse esimese prototüübi tegin Figma keskkonnas ning HTMLi, PHP-d (PHP: Hypertext Preprocessor) ja Javascripti kasutades *mockupi wamp64 (Windows, Apache*

Web Server, MySQL, PHP, 64-bit) abil. Seal realiseerisin ka rakenduse stiiliraamatu (joonis 21).



Joonis 20a. Päise menüü kujundus *desktop* vaates.

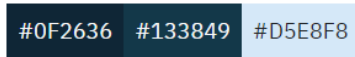


Joonis 20b. Päise menüü kujundus telefonivaates.

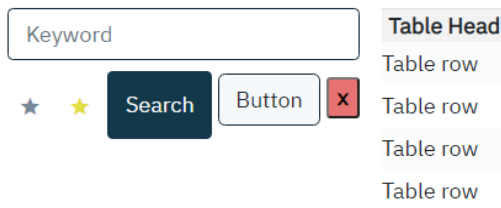
Header



used colors:



Content



used colors:



used fonts:

IBM Plex Sans(h1)

IBM Plex Sans(h3)

IBM Plex Sans(b)

IBM Plex Sans(p)

5 Tulemused

5.1 Realiseeritud lahendused

Käesolevas versioonis on realiseeritud rakenduse kogu põhifunktsionaalsus ehk hangete otsimine üle teatiste, seotud ehitiste ja kasutaja profiilide. Rakenduse *endpointi* saadetavate JSON päringutega saab nii kasutajaid luua kui ka teatise ja profiile lisada. Rakenduse päises on võimalik operatiivselt vahetada nii keelt kui firmat.

Kujundus vastab esialgsetele tingimustele ja on tugevalt inspireeritud eelnevalt joonisel 5 toodud mallist. Tänu kujunduse dünaamilisusele on rakendus normaalselt kasutatav ka mobiiltelefoni või tahvelarvuti ekraanil.

5.2 Ilmnenud puudused

Arendusel käigus tuli välja mõningaid puudujääke nii rakenduses kui ka sisse tulevates andmetes. Näiteks Mercelli keskse infosüsteemi andmed, kaasa arvatud projekti koostamisel kasutatud andmevalim on kohati mittetäielikud. Seda tuli andmete andmebaasi lugemisel eraldi arvestada ning ka viia sisse mõningad korrektuurid andmebaasi struktuuris. Kohati on aadressid sisse kantud ilma maja numbrita, mis raskendab oluliselt ehitiste kaardistamist rakenduses. Need andmed tuleks aga korrigeerida keskses serveris, kuni see ette võetakse peab rakenduse arendamisel toodud kitsaskohti arvesse võtma.

Esialgelt valitud projekti skoop oli selgelt liiga ambitsioonikas, plaanis oli näiteks ka ehitiste otsingul tulemuste kaardistamine mis jäi osalt ka ülal toodud probleemide tõttu teostamatuks. Samas lisandus arenduse käigus ka vajadus teha teatistele profiili kaupa sorteeritud otsimine.

5.3 Saadud kogemused

Kogu süsteemi otsast peale arendamine oli üpriski keerukas ja stressirohke, kuid nüüdseks peaaegu aastane töökogemus selles valdkonnas töötamisel tuli kindlasti kasuks. Palju sai ka kasutatud ülikoolis õpitut ja kirja pandut, seda eriti andmebaasipäringuid

koostades. Õppisin rakendust arendades põhjalikumalt tundma Entity Framework Core raamistikku ja MVC mudeli kasutamist.

Kindlasti aitaks see kogemus vajadusel sarnasel platformil uut projekti paremini ette valmistatuda ja hallata. Olles mitte varem agiilse arenduse põhimõtetega vaid põgusalt kokku puutunud, sain nüüd selle rakendamisega põhjaliku kogemuse.

5.4 Võimalikud edasiarendused

Käesolev töö on mõeldud otseseks baasiks või aluseks hilisemaks arenduseks Merzell Estonia OÜ poolt. Plaanis on näiteks lisada aktiivsete hangete kaardil kuvamise funktsioon ja algoritmil põhinev, kasutaja-spetsiifiline soovitusüsteem. Kahjuks ei oleks nende komponentide rakendusse lisamine olnud ajakava arvestades realistlik.

On olnud mõtte ehitada sarnane süsteem ka ülejäänud hangete jaoks. See idee jääks aga kaugesse tulevikku, arvestades et esmalt tuleks käesoleva süsteemi edasiarendus lõpetada ning arvestades arendusel kogetud raskusi oleks enne kasulik ka Merzelli keskse andmebaas ajakohastada.

6 Kokkuvõte

Kokkuvõtteks saab öelda, et rakenduse põhifunktsionaalsus, milleks on hangete otsimise funktsioon, sai teostatud ja kõik põhivaated on töökorras. Tänu kasutatud MVC ja DTO arendusmustritele on kood edasi arendamiseks selgelt struktureeritud, seega on valminud rakendus heaks aluseks mille pealt ka edasist arendust jätkata. Lisaks sai enamjaolt paika ka rakenduse üldine kujundus ja küljendus.

Seega võib töö tulemusega üldjoontes rahule jääda, kuigi ei jõudnud nii kaugele kui oli algul lootus jõuda. Paljud ette tulnud takistused tunduvad tänu saadud kogemustele nüüdseks triviaalsed ja sain kasuliku kogemuse ka tulevikus taolise projekti otsast peale arendamiseks.

Kasutatud kirjandus

- [1] „europa.eu - Common procurement vocabulary,“ [Võrgumaterjal]. Available: https://ec.europa.eu/growth/single-market/public-procurement/digital/common-vocabulary_en. [Kasutatud 08 05 2021].
- [2] „europa.eu - Background - NUTS,“ [Võrgumaterjal]. Available: <https://ec.europa.eu/eurostat/web/nuts/background>. [Kasutatud 08 05 2021].
- [3] „Riigihangete register,“ [Võrgumaterjal]. Available: <https://riigihanked.riik.ee/rhr-web/#/>. [Kasutatud 15 05 2021].
- [4] „Merzell - Lisa hange,“ [Võrgumaterjal]. Available: <https://www.merzell.com/et-ee/37399370/lisa-hange.aspx>. [Kasutatud 15 05 2021].
- [5] „Microsoft Docs - ASP.NET Core Documentation,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0>. [Kasutatud 25 05 2021].
- [6] „jQuery,“ [Võrgumaterjal]. Available: <https://jquery.com/>.
- [7] „Bootstrap,“ [Võrgumaterjal]. Available: <https://getbootstrap.com/>.
- [8] „PopperJS,“ [Võrgumaterjal]. Available: <https://popper.js.org/>.
- [9] „Bitbucket,“ [Võrgumaterjal]. Available: <https://www.bitbucket.org/>.
- [10] „Trello,“ [Võrgumaterjal]. Available: <https://www.trello.com/>.
- [11] „Figma - Merzell uudiskiri,“ [Võrgumaterjal]. Available: <https://www.figma.com/proto/tCYGjFpSBdX4qrnRS5M23R/Merzell-uudiskiri>. [Kasutatud 16 05 2021].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Aldin Talve

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Ehitushangete otsimise veebirakendus“, mille juhendaja on Gert Kanter
 - 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.

25.05.2021

¹ 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 loomingulise 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.