Net Group Kandideerimisülesanne

# Sugupuu, 2020

Tristan Krass

# [Frontend kood](https://github.com/tristankrass/GenyFront)

# [Backend kood](https://github.com/tristankrass/Genyology)

# Kasutatud tehnoloogiad:

* .NET Core
* React
* MSSQL
* Azure
* Firebase

# Ärireeglid

1. Ei saa olla kaks ema, kaks isa(peab olema üks isa, üks ema)
2. Laps ei tohi olla vanem kui tema ema või isa.
3. Iga suhtega on seotud kaks isikut.
4. Iga suhtega on seotud õks või mitu perekonda.
5. Isiku õdedel ja vendadel on ühised vanemad.
6. Iga perekonnaga peab olema seotud vähemalt üks isik.

## Validatsioonid

1. Laps ei saa olla sündinud tulevikus.

2. Laps ei saa olla sündinud varem, kui tema vanemad

3. Inimese ees- ja perenimi peab olema vähemalt üks täht pikk ja ei tohi olla null.

4. Inimene peab olema seotud vähemalt ühe perekonnaga, kuid võib olla seotud rohkem kui ühe perekonnaga.

5. Iga suhtega seotud inimesel on üks kindel roll(ema, isa, poeg, jne).

6. Igal suhtel on kindel suhte tüüp. (ema-poeg, isa-poeg, jne).

# Andmemudel

# 

# Lahenduseks kulunud aeg

Oma aja paremaks planeerimiseks kasutan Wakatime, ning backendi lahenduse jaoks kulunud aeg on leitav [siit](https://wakatime.com/@tristankrass/projects/pxqlqgsmoc?start=2020-03-25&end=2020-03-31). Frondi jaoks kulunud aeg on leitav [siit](https://wakatime.com/@tristankrass/projects/jvxhvctjmo?start=2020-03-25&end=2020-03-31). Muidugi pole arvestatud siin andmemudeli jaoks kulunud aega ning aega, mis kulus arvutist eemal mõeldes lahendusele. Tulenevalt asjaolust, et hetkel olen võtnud 48 EAP-d selle semestri raames, siis lahenduse implementeerimiseks palju aega ei jäänud. Suurem osa lahendusest kirjutasin kolmekümne esimesel märtsil.

# Probleemid, mida ei jõudnud lahendada

* Ühe isiku lisamise/muutmise vaade.
  1. Hetkel jäi ajast puudu, kuid vajalik kontroller on selleks backendis olemas.
* Mitmes laps peres on antud isik (isikut võimalik valida loetelust)?
  1. Seda on võimalik lahendada järgmiselt: Teha päring, mis loeb kokku, isikute arvu, kellel on sama perekonnanimi kui sellel isikul.
  2. Nüüd tuleb võrrelda isiku sünniaega teiste isikute sünniajaga. Kõige varem sündinud isik on sugupuu tipus jne.
* Kellel on kõige rohkem eelkäijaid? Kes on tema eelkäijad?
  1. Sarnaselt eelneva lähenemisega on võimalik lahedada ka antud probleem.
* Kes on kõige noorem onu või tädi?
  1. Iga suhtega on seotud kaks inimest. Igal suhtega on seotud kaks rolli. Teha päring, mis leiab üles kõik isikud, kes on seotud otsitava isikuga ja kelle üheks rolliks on vastavalt tädi või onu.
  2. Järgneval tuleb välja valida isik, kelle sünnikuupäev on kõige hilisem.
* Kasutaja saab valida ühe isiku ning joonistatakse välja tema sugupuu. Esialgselt valitud isik on selgelt visuaalselt eristatav.
  1. Probleem on seotud eelkõige visuaalse poolsega ehk Front endiga. Sellest tulenevalt kasutaksin olemasolevat libraryt. Esimesel pilgul jäi silma [vis-network](https://github.com/visjs/vis-network).

# Mida saab rakenduses paremini teha?

## Backend

* Hetkel on kõik päringud kontrolleris. Hoopis parem oleks kui kontroller ei teaks midagi meie andmekihist. Üks võimalik varian oleks Repository mustrit või DAO mustrit.
* Hetkel pole kontrollerid kaitstud. Iga inimene, kes soovib saab isikutele ligi. Minimaalselt tuleks lisada igale päringule näiteks globaalne filter, mis kontrollib, et kasutaja saab ainult endaga seotud isikud tagasi. Rest API jaoks on vaja genereerida JWT tokenid.

## FrontEnd

* Hetkel vajab UI/UX kindlasti tööd. Tuleks teha näiteks Figmas võiks eelnevalt visuaalsed vaated valmis teha.
* Koodistruktuur tuleks ümber muuta. Tuleb otsustada, kas kasutada state manageerimiseks reduxi või Reacti enda pakutud võimalusi.