

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Mihkel Koks 210780IADB

Õpigruppide leidmist toetava veebirakenduse arendus

Bakalaureusetöö

Juhendaja: Märt Kalmo
MSc

Tallinn 2024

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: Mihkel Koks

04.01.2024

Annotatsioon

Lõputöö "Õpigruppide leidmist toetava veebirakenduse arendus" käsitleb probleemi, kuidas luua õpilastele virtuaalselt paremaid tingimusi grupitöödeks ja koos õppimiseks. Koolid kasutavad mitmesuguseid suhtluskanaleid ja õppetööd toetavaid rakendusi, kuid need on enamasti individuaalsele tööle suunatud ja sageli segavad keskendumist. Lõputöö eesmärk on välja töötada rakendus, mis keskendub just õpigruppide moodustamisele ja koostööle, vähendades segavaid elemente ning pakkudes efektiivseid suhtlusvõimalusi.

Uuritakse olemasolevaid lahendusi, analüüsitakse nende puudujääke ja kavandatakse rakendus, mis vastaks õpilaste vajadustele ja õppimisele. Rakendus on kavandatud selliselt, et see aitaks õpilastel leida sobivaid õpigruppe, pakuks mugavat ja segamatut suhtluskeskkonda ning võimaldaks meeldetuletusi ja teavituste haldamist. Eesmärgiks on luua lihtne, kasutajasõbralik ja keskendunud õpikeskkond, mis toetab hariduslikke eesmärke.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 26 leheküljel, 7 peatükki, 20 joonist.

Abstract

Development of a Web Application that Supports Finding Study Groups

The thesis "Development of a Web Application that Supports Finding Study Groups" addresses the problem of creating better virtual conditions for group work and collaborative learning for students. Schools use a variety of communication channels and learning support applications, but these are mostly directed towards individual work and often distract from concentration. The aim of the thesis is to develop an application that focuses on forming study groups and collaboration, reducing distracting elements, and providing effective communication options.

Existing solutions are examined, their shortcomings analyzed, and a new application is designed to meet the needs of students and learning. The application is designed in such a way that it helps students find suitable study groups, offers a convenient and undistracted communication environment, and allows for the management of reminders and notifications. The goal is to create a simple, user-friendly, and focused learning environment that supports educational objectives.

The thesis is in Estonian and contains 26 pages of text, 7 chapters, 20 figures.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> ehk rakendusliides.
BLL	<i>Business Logic Layer</i> ehk äriloogika kiht.
CLI	<i>Application Programming Interface</i> ehk rakendusliides.
DAL	<i>Data Access Layer</i> ehk andmete ligipääsukiht.
DOM	<i>Document Object Model</i> ehk Dokumendi objekti mudel.
HTML	<i>HyperText Markup Language</i> ehk märgistuskeel.
HTTP	<i>Hypertext Transfer Protocol</i> ehk hüperteksti edastamise protokoll.
MVC	<i>Model-View-Controller</i> ehk mudel-vaade-kontroller.
REST	<i>Representational State Transfer</i> ehk Esindusliku oleku ülekanne.
SPA	<i>Singel-Page Application</i> ehk üheleherakendus.
URL	<i>Uniform Resource Locator</i> ehk ühtne ressursiviitaja.

Sisukord

1 Sissejuhatus	9
1.1 Taust ja aktuaalsus	9
1.2 Probleemi kirjeldus	10
1.3 Eesmärk	10
1.4 Metoodika	10
2 Olemasolevate lahenduste uurimine ja rakenduse visioon.....	11
2.1 Rakenduse Visioon.....	11
2.2 Olemasolevate lahenduste uurimine.....	12
3 Lahenduse analüüs	15
3.1 Lahenduse esialgselt seatavad nõuded	15
3.1.1 Funktsionaalsed nõuded.....	15
3.1.2 Mittefunktsionaalsed nõuded	15
4 Tehnoloogiate analüüs ja valik	16
4.1 Programeerimiskeele valik	16
4.2 Tarkvara arenduse raamistiku valik	17
4.3 Kasutajaliidese raamistiku valik	17
4.4 Reaalajas suhtluse teegi valik	18
4.5 Andmebaasisüsteemi valik	19
5 Teostus	21
5.1 Serveripoolse rakenduse arendus	21
5.1.1 ASP.NET Core Projekti loomine	21
5.1.2 Arhitektuur ja struktuuri paika panek	22
5.1.3 REST liides	24
5.1.4 SignalR rakendamine serveri poolel.....	26
5.1.5 Turvalisus.....	27
5.2 Kliendipoolse rakenduse arendus.....	27
5.2.1 Rakenduse disain	28
5.2.2 Funktsioonide loomine	28
5.2.3 SignalR rakendamine kliendipoolses rakenduses.....	30

6 Tulemused	32
7 Kokkuvõte	34
Kasutatud kirjandus	35
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	37
Lisa 2 – Olemasolevate rakenduste otsimiseks kasutatud fraasid	38
Lisa 3 – Otsingus leitud olemasolevad rakendused	39

Jooniste loetelu

Joonis 1 Andmebaasi olemi suhte diagramm.	20
Joonis 2 Projekti loomine.	21
Joonis 3 Näide NuGet paketihaldurist Rideris.	22
Joonis 4 Serveripoolse rakenduse arhitektuur.	23
Joonis 5 Serveripoolne rakenduse struktuur.	24
Joonis 6 GET Päring.	25
Joonis 7 PUT päring.	25
Joonis 8 POST päring.	26
Joonis 9 DELETE päring.	26
Joonis 10 SignalR lisamine teenuste konteinerisse.	26
Joonis 11 SignalR marsruudi määramine ja veebisoklite konfigureerimis näide.	26
Joonis 12 SignalR meetodi näidis, et saata sõnum kõigile grupis.	27
Joonis 13 rakenduse üldise disaini näide.	28
Joonis 14 Eesrakenduse struktuur.	29
Joonis 15 UseEffect näide.	30
Joonis 16 SignalR kliendipoolel ühenduse loomise näide.	30
Joonis 17 SignalR komponendist kuvamise näide.	31
Joonis 18 Näide suhtluse akenast.	32
Joonis 19 Näide kursuste valikust.	33
Joonis 20 Näide õpigruppide haldamise aknast.	33

1 Sissejuhatus

1.1 Taust ja aktuaalsus

Uuringute kohaselt leiavad osad õpilased, et lisaks individuaalsetele õpingutele on neile kasulik ka grupis õppimine. Selgus, et 78% vastanutest osalesid õppejõudude julgustusel ja kaaslaste mõjul vähemalt ühes õpigrupi sessioonis. Samuti oli toodud välja, et õpilased, kes osalesid õpingu protsessis koostöös teiste õpilastega, saavutasid paremad tulemused. Veel selgub uuringust, et õpilased, kes õppisid grupis ja kasutasid tõenduspõhiseid õppemeetodeid, näiteks küsimuste esitamine üksteisele, kursusematerjali läbi arutamine ja üksteise küsitlemine, märkasid, et see süvendas õppimist ja parandas materjali meeldejätmist [1].

Kooli poolt on õpilastele soetatud erinevaid rakendusi õppetöö parandamiseks, kuid need on pigem individuaalsele tööle suunatud. Hetkel kasutavad õpilased erinevaid suhtluskanaleid, sealhulgas sotsiaalmeedia platvormi nagu Facebook, samuti e-posti, suhtlusplatvormi nagu Discord ja Microsoft Teams ning koostöövahendeid nagu Google Drive ja Zoom. Võimalusel saadakse kokku ja leitakse võimalus näha üksteist ekraaniväliselt, mis on minevikku vaadates olnud raskendatud, kui peaks tekkima eriolukordi, mis takistavad sellisel viisil kohtumist.

Olemasolevatel rakendustel on olemas mitmeid vajalike funktsioone ja mõnel isegi üleliigselt, mis hakkavad segama keskendumist ja vähendavad produktiivsust kuni 40%, mida jällegi tõestavad uuringud [2]. Nimetatud rakendustel on vastavalt nende tegelikule eesmärgile mõned puudused, mis mõjutavad nende kasutatavust grupitöödeks ja nii tekib vajadus veel teise rakenduse järgi.

1.2 Probleemi kirjeldus

Koolil puudub oma esmane rakendus, mis toetab õpilaste koostööd ja koosõppimist, andes neile täiendava võimaluse haridusprotsessis osalemiseks ja õppimiseks vajalike sidemete loomiseks. Kuigi olemasolevad suhtluskanalid võivad pakkuda suhtlemisvõimalusi, puuduvad või on neil üleliigsed ja õppimisele keskendumist segavad spetsiifilised funktsioonid ja elemendid, mis viivad õpilase tähelepanu mujale ja segavad koostööd ning üldist keskendumist eesmärgile. Oleks vaja leida või luua rakendus, mis vähendab segavaid elemente oma platvormil ja tooks esile just vajaminevad. Sotsiaalmeedia, mida õpilane kasutab vabal ajal, võiks jääda eraldatuks õpinguteks kasutatavatest rakendustest, et hoida õppetöö välised asjad eraldatuna koolielust.

1.3 Eesmärk

Eesmärgiks on leida või välja töötada rakendus, mis lahendaks probleemi, kus õpilastele oleks virtuaalselt võimalik luua paremad tingimused grupitöödeks ja koos õppimiseks. Rakendus, mida on vaja, peab olema piisavate funktsioonidega, mis aitavad kaasa õpingutele.

Rakenduse puudumisel, tuleks vajalik rakendus arendada ja leida parimad moodused selle tegemiseks.

1.4 Metoodika

Antud lõputöö tehakse arendusuuringuna, kus analüüsitakse probleemi. Leidmaks sobivat lahendust uuritakse, milliseid sarnaseid lahendusi on juba olemas. Analüüsitakse, mida on nende puhul tehtud ja mida on võimalik rakendada selle rakenduse puhul.

Viimaks kirjeldatakse rakenduse valmimisprotsessi ja hinnatakse, kas see rakendus lahendab probleemi.

2 Olemasolevate lahenduste uurimine ja rakenduse visioon

Hetkel kasutusel olevad rakendused, mis on eelnevalt mainitud, ei rahulda soovi väga hea keskkonna järgi, kus saaks õppida teiste õpilastega koos, ilma et oleks segavaid elemente. Erinevad üleliigsed ja ebaolulised teavitused ning postitused. Mõnel juhul puudub rakendusel esmatähtsad funktsioonid, et üldse luua hea kontakt teiste õpilastega. Üleliigsed funktsioonid segavad keskendumist ning ei sobi akadeemilisse keskkonda, kus peaks saama keskenduda vaid olulistele ülesannetele. Teisest küljest, on puudu esmased olulised suhtlust toetavad elemendid, nagu võimalus sujuvaks kirj vahetuseks.

2.1 Rakenduse Visioon

Selle rakenduse eesmärk on toetada õpilasi nende haridustee konnal, pakkudes neile paindlikke ja interaktiivseid vahendeid õpigruppide moodustamiseks ning efektiivseks koostööks. Rakendus peaks olema lihtne ja keskenduma õppimisele, mitte kasutajat kõrvaliste tegevustega ahvatlema.

Rakendus peab võimaldama õpigruppide loomist ja nendega liitumist, loomiseks piisaks sellest, et üks liige loob grupi ja saab edastada selle aadressi teistele liikmetele, ning selle kaudu oleks neil võimalus liituda. Selline lähenemine aga ei sobiks, kuna eesmärk on, et õpilastel jääks ka ligipääs viimastele vestlustele, mille saavutamiseks on otstarbekas luua isikulised kasutajad, mis hoiavad rohkemat informatsiooni, ehk võimaldaks salvestada isikuandmeid. Isikuandmed aitaks tekitada uusi võimalusi sorteerida õpilasi ja annaks neile sorteerimisel mingeid parameetreid, mille järgi neid valida gruppidesse. Lisaks on kasutajate süsteem hea, võimaldamaks otsida inimesi, kui ei ole juba eelnevalt teada, kellega sooviks koostööd teha. Sellest tulenevalt peaks olema süsteem, kus kasutaja luua, ning hiljem vajadusel ka muuta andmeid sellel kasutajal.

Õpigruppide kuvamisel peaks olema näha informatsioon, mis võimaldaks õpilasel saada aru, kas antud grupp on sobiv temale ja kas tal on üldse võimalik veel selle grupiga liituda. Peaks olema võimalik saada eraldi infot erinevatest õpigruppidest ja täpsustav info iga õpigrupi kohta. Selle informatsiooniga on õpilasel hea saada ülevaade, ning see kiirendab

nende grupi otsingut. Igal grupil võiks olla kohe näha selle grupi nimi, mis ainetele see võiks keskenduda, kes on selles grupis juba olemas ja palju on veel võimalik sellesse gruppi õpilasi mahutada.

Kuna tegemist on rakendusega, kus tihti on töö koos mitme inimesega ja on vaja planeerida kohtumisi, siis oleks vaja seada meeldetuletusi. Võiks olla võimalik kuvada rakenduses endas, mingil valitud hetkel, mis kasutaja on seadnud, et sellel kellaajal ja kuupäeval peaks tekkima aken, mis annab vajaliku info. Kasutajatel on ka vaja näha muid teavitusi erinevatest toimingutest, selleks tuleks luua ka koht, kus saaks kõiki muutusi näha, kuid see peaks olema mitte visuaalselt häiriv, et õppimise ajal ei oleks segavaid elemente.

Väga olulisel kohal on võimalus suhelda üksteisega. Suhtlus peaks olema aga reaalselt, et see oleks sujuv ja kiire ning aitaks nii kaasa, et õpilaste töös ei tekiks pause. Esmalt võiks olla vähemalt kirjavahetusega suhtlus, millele lisada ka hiljem võimalus video- või häälkõneks. Kirjavahetuseks peaks esmalt olema oma keskkond, kus saab vahetada kirju reaalselt, tagamaks sujuva ja mugava vestluse. Vestluse aknas oleks hea näha, mis grupis vestlus toimub ja kes on selles grupis, et oleks arusaadav, kus parajasti kirjavahetus toimub ja kes on inimesed, kellele kiri jõuab. Iga kirja juures peab olema näha, kes kirja kirjutas ja millal. See aitaks pidada järge ning kui tekib vajadus midagi tagantjärele vaadata, on parem otsida kirjavahetuses toetudes ajanäitajale kirjade juures.

2.2 Olemasolevate lahenduste uurimine

Selleks, et leida, kas visioonist tulenev rakendus on olemas või kas sellega sarnane on juba olemas, tuleks kõigepealt mõelda, kuidas teostada otsing. Otsides erinevaid rakendusi, mis oleks seotud grupiviisilise õppega, on lähenetud lahendustele erinevalt ja üritatud lahendada sarnaseid, kuid mitte päris neid murekohti, mis on tekkinud selle lõputöö käigus. Probleme, mida on näinud teiste rakenduste loojad, on olnud erinevaid. Peamiselt saab näha lahendusi, kus saab pigem võõrastega liituda ühises jututoas, millel oli palju lisasid, millest mõned oleksid isegi häirivad, mis eiravad eesmärki vältida kõrvalisi toiminguid. Lisaks on suur probleem, kus paljud funktsioonid olid limiteeritud või täielikult tasulised.

Leidmaks eksisteerivaid rakendusi, mis sarnanevad autori visiooniga tuleb kasutada Google otsingumootorit [3]. Otsingus kasutati ingliskeelseid fraase, mis on ka välja toodud lisades (Lisa 1).

Uurimise tulemusena jäi skooopi 11 erinevat rakendust (Lisa 2), millest valiti välja 3. Kuvatavad rakendused on sellised, mis omasid vähemalt mõnda vajaminevat elementi, mis oli visioonis väljatoodud. Lisaks on oluline, et ei oleks suurem osa peamiselt vajaminevaid funktsioone piiritletud maksva versiooni taha.

Otsingu tulemustest saadud rakendustest valiti välja need, mis tabasid visiooni kõige paremini, ehk omasid võimalikult palju vajaminevaid funktsioone ilma, et need oleks suuremas osas vaid tasulise versiooniga saadaval ja ei olnud liialdatud funktsioonidega, mis võiksid häirida keskendumist. Selliseid rakendusi oli 3, mis suutsid täita seatud nõuet piisavalt hästi.

1. ThinkBinder - ThinkBinder on väga huvitav ja lahenduselt päris sarnane käesoleva lõputöö autori visioonile. Antud rakendus võimaldab suhelda kirjavahetuse teel, mis on ka reaajas ning on palju huvitavaid lisasid, mida võiks kaaluda tulevikus lisada esialgsesse visioonist tulenevasse rakendusesse. Gruppide valim oli ka näha, et oli loodud, kuid kahjuks sellest palju infot ei saanud. Märgete jätmis võimalus on hea lisa, mis jääb silma ning teiseks oli tahvli olemasolu, kus saaks jooniseid luua. Tahvli osa oli küll pigem veidikene liiga lihtne ja sellel ei pruugi antud versioonis olla piisavalt funktsioone, kuid ideena on ta päris hea [4].

Puudulikkuse osas on näiteks see, et kirjavahetuse aken on liiga väike ja kõik on üritatud suruda kokku ühte kohta, autori visiooni kohaselt võiksid asjad olla rohkem eraldi, kuigi siin on mõttekoht, kas kõik kohe nähtaval on ehk mugavam, või siis pigem oleks parem kui on eraldi. Saaks kaaluda võimalust lasta kasutajal asju liigutada, kuid selline funktsioon jääks arendamiseks tulevikku.

2. Aircend – Aircend on teine väga lähedane rakendus visioonile, kus on samuti väga heal tasemel funktsioone. Sellel rakendusel jääb silma kirjade saatmis võimalus, mis nägi välja põhimõtteliselt täpselt selline, nagu oli autori visioonis seda ettekujutanud. Lisaks jääb silma hea süsteem, meeldetuletuste loomiseks, mis

oleks kindlasti vajalik ka autori rakenduses. Viimaks oli loodud huvitav failide hoiustamis süsteem, kus sai neid hoida ja edastada ning ka vajadusel edastada teistele läbi sõnumite. Aircsend on nüüdseks ka avatud lähtekoodiga.

Negatiivsest küljest oli näha, et varasemalt oli seal ka tasuline versioon, mille taha oli peidetud grupikanali loomine, mis on kindlasti takistus ja vastuolus visiooniga [5].

3. Slack – Slack osutus samuti valituks oma heade osadega, mis kattuvad visiooniga. Slackil on üldine kontseptsioon, mis toetaks akadeemilist tegevust, kuigi öeldud, et on loodud rohkem tööalasteks tegevusteks, mis ei erineks hetkel väga akadeemilistest tegevustest. Head osad olid Slackil kanalite loomine, mis küll jääb tasulise versiooni taha, kuid see, kuidas see töötas ja milliseid võimalusi andis, oli siiski märgatav ja saab võtta eeskuju ka loodavas rakenduses. Samuti oli neil hea liikmete gruppi lisamise võimalus, mida oleks võimalik lisada rakendusse hiljem. Slacki peamine probleem on, et palju vajalike elemente on peidus tasulise versiooni taga, mille tõttu ei ole ta piisavalt hea lahendus [6].

3 Lahenduse analüüs

Käesolevas alapeatükis analüüsitakse, millised nõuded peaks saama täidetud, et leida probleemile minimaalne lahendus.

3.1 Lahenduse esialgselt seatavad nõuded

Lahenduse funktsionaalsete ja mittefunktsionaalsete nõuete seadmisel võetakse aluseks töös püstitatud probleem ja pakutud lahenduse visioon. Nõuete määramisel tuleks arvestada ajalise piiranguga, millest tulenevalt saab valmistada rakenduse, mille alusel saab selgusele, kas on leitud lahendus probleemile. Nõuded määravad antud töö skoobi.

3.1.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded määravad, mida süsteem peab suutma pakkuda kasutajale.

- Rakenduses peab olema võimalus kirjavahetuseks grupis.
- Rakendus peab suutma salvestada kirjavahetused, et neid hiljem jätkata.
- Rakendus peab võimaldama näha erinevaid õppeaineid.
- Rakenduses peab olema võimalik kuvada nimekiri õpilastest ja neid otsida.
- Rakenduses peab olema võimalik valida erinevate gruppide vahel, kus kasutaja on lisatud.
- Rakenduses peab olema võimalik luua meeldetuletusi.

3.1.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded määravad süsteemi kvaliteedi ja töökindluse.

Mittefunktsionaalsed nõuded peavad kirjeldama, kuidas takvara peab neid funktsioone täitma.

- Toetus vähemalt Eesti ja Inglise keelele.
- Rakenduses peab olema võimalik kõike kasutada tasuta.
- Rakenduses peab olema lihtne navigeerida.

4 Tehnoloogiate analüüs ja valik

Antud töö tulemusena peab valmima veebirakendus, mis vastab probleemi analüüsis seatud nõuetele. Selleks, et selline rakendus luua, tuleks mõelda läbi milliste vahenditega saavutada oodatud tulemus, kus tuleb arvestada nõudeid, mis on varasemalt püstitatud. Rakenduse arendamiseks tuleks võtta vaatlusesse kaasaegsed ja kestva toega tehnoloogiaid, millel on olemas piisavalt varasemaid kasutajaid. Tehnoloogia valikul, tuleb ka silmas pidada, et tehnoloogia ei oleks tasuline. Selles peatükis tuuakse välja, millised rakendused olid kaalumisel ja millistel põhjustel osutusid valituks.

4.1 Programeerimiskeele valik

Tehes valiku keelte osas, peab arvesse võtma mitmeid asjaolusid. Esmalt, tuleks mõelda sellepeale, millised on rakenduse nõuded ja funktsionaalsus. Sellele järgnevalt on oluline keele kogukond ja toe olemasolu, et oleks lihtsam leida abistavaid materjale. Kindlasti on ka oluline vaadata, milliste raamistikega on kokku sobivus ja viimaks on oluline aru saada, millised on arendaja, ehk töö autori võimed ja oskused. Nendest lähtudes, peaks olema valitav keel staatiliselt tüübitud, kus muutuja tüüp on teada kompileerimise ajal. See võimaldab avastada vigu varem ja muudab koodi mõistetavamaks arendajale. Sellest lähtuvalt jäävad kõrvale keeled, mis on dünaamiliselt tüübitud, kus tüübi kontroll toimub kompileerimisel. Keeled, mis seega ei sobi on PHP, Python ja Javascript.

Valikusse jäävad keeled nagu: Typescript, C# ja Java. Sellest valikust on pigem sobilikumad C# või Java, Typescript tuleb rohkem kasutusse kliendipoolses arenduses. Nii java kui C# puhul on tegemist objektorienteeritud keeltega, tugdevalt tüübitud ja suure kasutajaskonna ning laia tööristade valikuga. Kuna mõlemad keeled sobivad antud töö tegemiseks väga hästi, siis tuleks lähtuda arendaja ehk töö autori oskustest. Kuna arendaja on pädevam ja rohkem kasutanud C#, siis valik langeb selle kasuks [7].

4.2 Tarkvara arenduse raamistiku valik

Rakenduse juurde käivale raamistikule pole erilisi nõudeid, millest selguks sobiv raamistik. Rakendus peab vastu võtma päringuid ja neid edastama, suhtlema andmebaasiga ja väljastama vajalikud andmed. Raamistik võiks olla võimeline toetama REST (*Representational State Transfer*) arhitektuuri.

Tulenevalt programmeerimiskeele valikust, milleks osutus C#, on selle rakenduse juures sobilikum ASP.NET Core, mis võimaldab luua dünaamilisi veebilehti, veebiteenuseid ja API-sid. ASP.NET Core sisaldab enamust vajalikest funktsioonidest ja sobib ideaalselt töö tegemiseks. Sellega tuleb kaasa MVC (*model-view-controller*), mis annab MVC arhitektuuri ja lisab mugavusi ja funktsionaalsust töösse [8].

ASP.NET Core kasuks räägib veel ka teenuste konfigureerimine ja keskkonnaseadete haldamine. Veel on kindlasti oluline turvalisus, kus antud raamistikul on olemas JWT autentimiseks [9].

4.3 Kasutajaliidese raamistiku valik

Eesrakenduse tehnoloogia valikul pole mingeid erilisi nõudeid. Jälgides, mis on populaarsemad eesrakendusraamistikud, siis on töö tegemiseks parimad valikud React, Vue ja Angular. Kõik kolm on aktiivselt uuenduses ja väljastatakse uusi versioone, kuid kindlasti ka hooldatakse vanemaid versioone. Kõigile kolmel raamistikul on ka suur kogukond ning toimub pidev areng. Valiku teeb ka raskemaks asjaolu, et kõik kolm jagavad palju sarnasusi. Valikusse jäävatel raamistikudel pole antud töö osas märgatavaid puudujääke ega eeliseid üksteise ees, mis teeksid valiku kergemaks [10].

Valikutegemisel tuleks vaadata rakenduse tegija ehk autori oskusi ja kogemusi, ning sellest lähtuvalt langes valik React raamistikule.

Reactis on võimalik kirjutada nii Javascriptis kui ka Typescriptis. Valik langeb siiski Typescripti peale, mis tagab tüüpide (primitiivide, liideste, generics jne) toetamisel suurema turvalisuse. See tuleb kasuks, üritades vältida koodis vigu või teostades hooldust, neid varem märgata. React raamistik aitab ka hoida kinni reeglitest ja hoida rakenduses struktuuri ja loetavat koodi, et lõpptulemus oleks võimalikult hea. Rakenduses kasutatakse HTML-il (*HyperText markup language*) põhinevaid malle (*templates*), mille

haldamisega tegelevad funktsionaalsed komponendid, mis on tänapäeval levinud alternatiiv klassipõhistele komponentidele, pakkudes sarnast funktsionaalsust. Komponentides kasutatakse teenust (*Service*), mis eraldab andmete haldamis loogika komponendi enda loogikast. See vastab teenusklassidele, kus rakenduse loogika on koondatud eraldi klassidesse. Selline lähenemine võimaldab koodi paremat organiseerimist ja taaskasutust. Kõik saab siduda kokku omavahel moodulitesse (*modules*) [11], [12].

4.4 Reaalajas suhtluse teegi valik

Rakenduse arendamisel, mis keskendub õpigruppide loomisele, on oluline reaalajas suhtluse võimaldamine. Suhtlus peab olema reaalajas, et tagada sujuv vestlus kasutajate vahel õpigrupis, mis on oluline, et tuleks parim tulemus nende eesmärgi saavutamiseks. Peamised tehnoloogilised valikud selle saavutamiseks on SignalR, WebSocket.io ja WebSocket. Igal nimetatud tehnoloogial on oma eripärad ja eelised reaalajas suhtluseks. Siiski, pärast nende analüüsimist ja arvestades rakenduse nõudeid, osutus valikuks SignalR.

SignalR ühildub hästi C#-ga, mis on antud rakenduse serveripoolse arenduse keel. Kuna SignalR on samuti Microsofti toode, on selle integratsioon olemasolevasse tehnoloogiatega hulka sujuv, mis vähendab potentsiaalseid ühilduvusprobleeme.

Erinevalt WebSocketist ja WebSocket.io-st, pakub SignalR abstraktsiooni erinevate transpordimehhanismide üle, nagu WebSockets, Server-Sent Events ja Long Polling. See valib automaatselt parima transpordimeetodi, lähtudes kliendi ja serveri võimalustest, mis lihtsustab arendusprotsessi.

SignalR toetab reaalajas andmete edastamist ühendatud klientidele, mis on selle rakenduse jaoks oluline. See funktsioon võimaldab õppegruppidele kohese suhtluse ja ajakohase infovahetuse, parandades kasutajate suhtlust ja koostööd.

SignalR toetab rakenduse skaleerimist, mis on oluline, kuna antud rakendus on mõeldud suurele kasutajate arvule [13], [14], [15].

4.5 Andmebaasisüsteemi valik

Valides andmebaasi peaks vaatama, millised on tänapäeval eelistatuid kogukonnas. Valikus peaksid kindlasti olema relatsioonilised andmebaasid, mis on tänapäeval eelistatuid [16], [17].

Töö skooopi jäävas andmebaasis on 13 tabelit (joonis 1):

AspUser – Kontod, mis omavad vajaliku kasutajate infot, mis on sisse ehitatud juba.

AspUserRole – Tabel hoidmaks rolle, mis võib kasutajal olla.

Studygroups – Tabel kus hoitakse informatsiooni õpigruppide kohta.

UserStudyGroups – tabel, kus on iga kasutaja enda õpigruppide info.

Courses – Tabel kus on info kõigi kursuste kohta.

UserCourses – Tabel kus on info kasutaja enda kursuste kohta.

GroupMessages – Tabel kus on grupisõnumite info.

GroupJoinRequests – Tabel kus on näha grupiga liitumise infot.

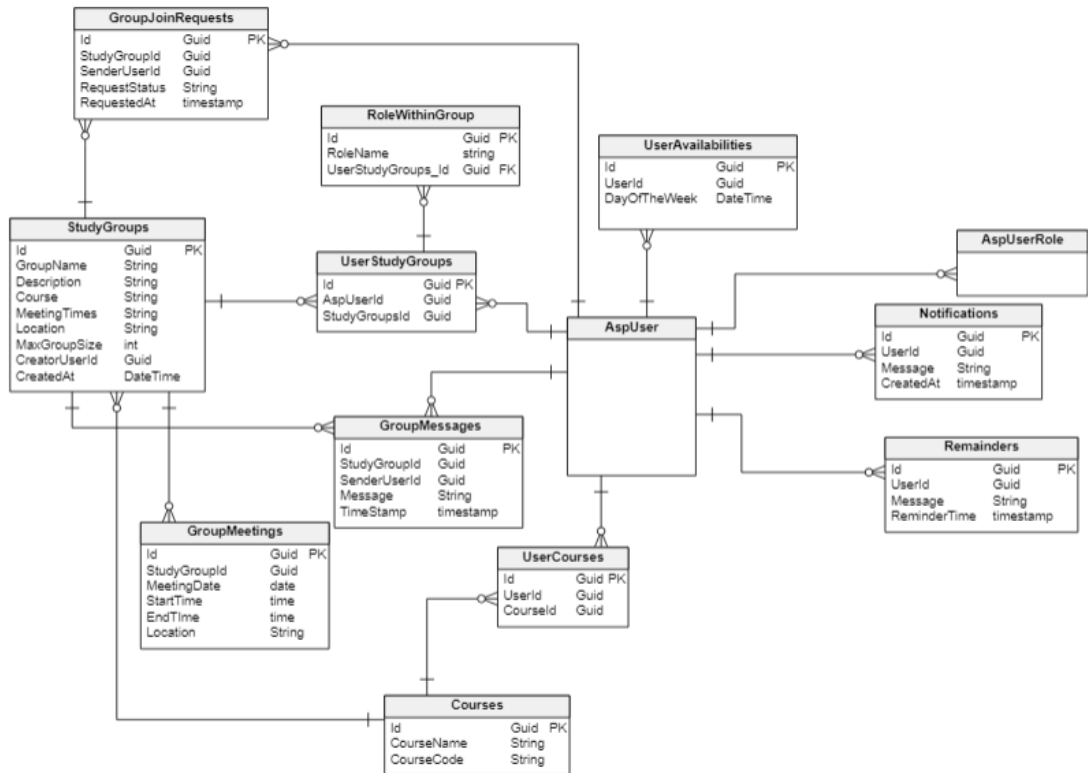
UserAvailabilities – Tabel, kus on näha kasutajate vabad ajad erinevateks tegevusteks.

Notifications – Tabel, kus on näha erinevad teavitused.

Reaminders – Tabel, kus on näha meeldetuletused.

GroupMeetings – Tabel, kus on informatsioon grupi kohtumiste kohta.

RoleWithinGroup – Tabel, kus on informatsioon rollide kohta gruppide sees.



Joonis 1 Andmebaasi olemi suhte diagramm.

Realisioonilistest andmebaasidest on tuntumad järgmised: SQLite, MSSQL, Oracle, MySQL, PostgreSQL. Vaadates rakendust, mis töö käigus valmib, siis sobiks väga hästi kas MySQL või PostgreSQL [6]. MySQL on lihtne ja kiire andmebaas võrreldes PostgreSQL-iga. PostgreSQL-i kasuks räägib aga paindlikus ja rohkemad funktsionaalsused. Arvestades autori oskusi ja kogemust mõlemaga, langeb valik PostgreSQL peale, lisaks võib paindlikus tulla kasuks tulevaste uuenduste ja lisade integreerimisel.

5 Teostus

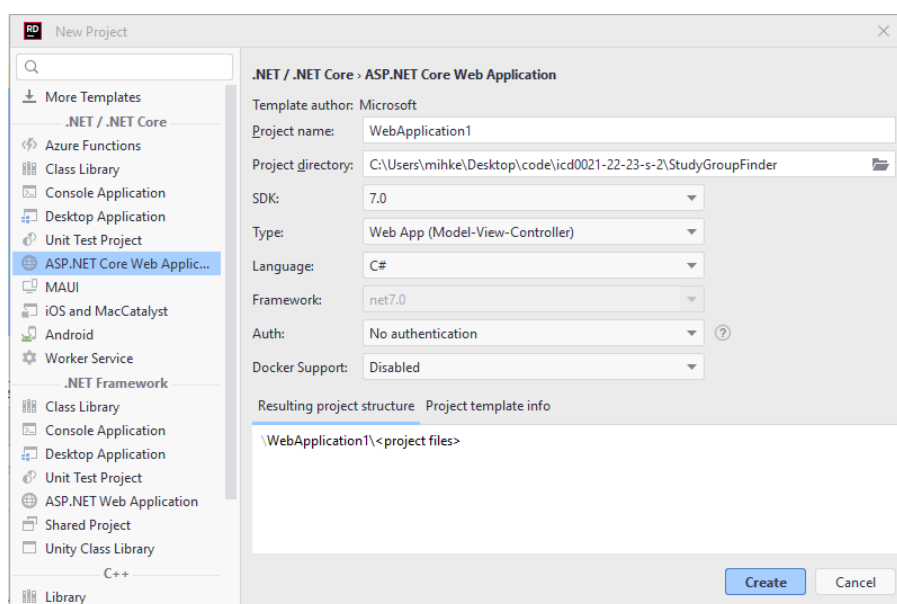
Selles osas kirjeldatakse rakenduse teostust ja arendusprotsessi. Peatükk on jaotatud kaheks: Serveripoolne rakenduse arendus ja kliendipoolne rakenduse arendus.

5.1 Serveripoolse rakenduse arendus

Serveripoolses rakenduse arenduses on peamine rõhk HTTP (*Hypertext Transfer Protocol*) päringute vastuvõtmisel ja vastuste tagasi saatmisel, lisaks on serveri poolel suhtlus andmebaasiga.

5.1.1 ASP.NET Core Projekti loomine

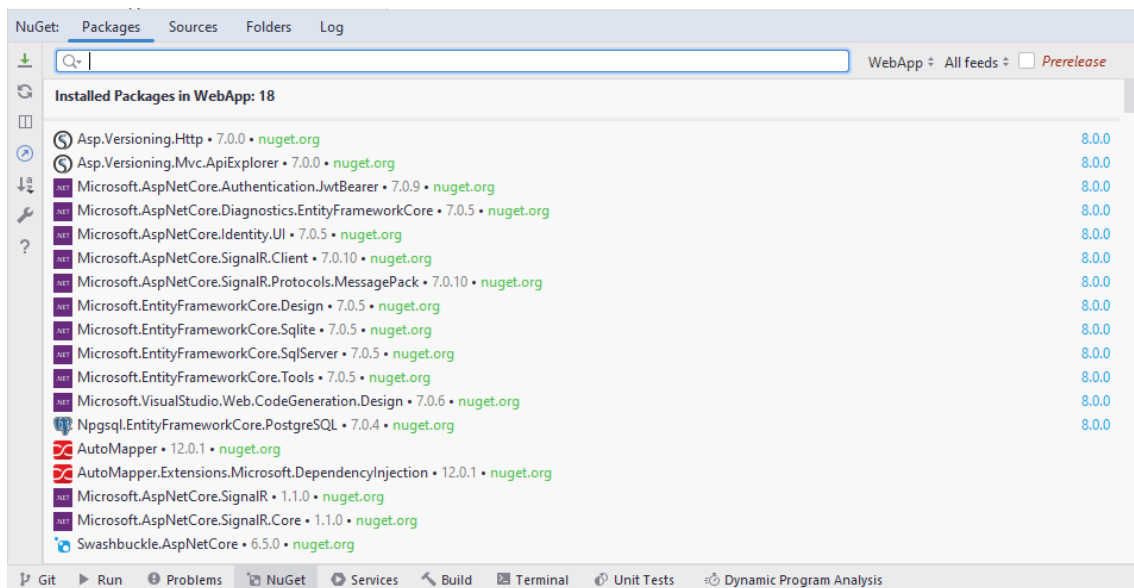
Rakenduse arendamisel C# ja ASP.NET Core'i keskkonnas JetBrains Rideriga alustatakse projekti loomist, keskendudes arendaja vajadustele ja valitud tehnoloogiatele. Rider genereerib vajaliku C# ja ASP.NET Core'i algstruktuuri. Rider pakub intuitiivset kasutajaliidest, kus on võimalik valida erinevate projektimallide vahel, sealhulgas veebirakenduste ja API-de arendamiseks sobivad mallid.



Joonis 2 Projekti loomine.

Valitud on ASP.NET Core Web Application projekt, mille puhul saab täpsustada konkreetse ASP.NET Core versiooni, milleks on 7.0 ja muid seadeid, nagu näiteks projekti nimi ja asukoht. Seejärel genereerib Rider automaatselt projekti struktuuri, mis sisaldab kõiki vajalikke sõltuvusi ja konfiguratsioone, et arendaja saaks kohe alustada rakenduse arendamist.

NuGet'i paketihaldur, mis on integreeritud Riderisse, võimaldab lihtsat sõltuvuste haldamist ja projekti konfiguratsiooni. See võimaldab arendajal lisada uusi teekide ja raamistike, et laiendada rakenduse funktsionaalsust või parandada selle jõudlust.



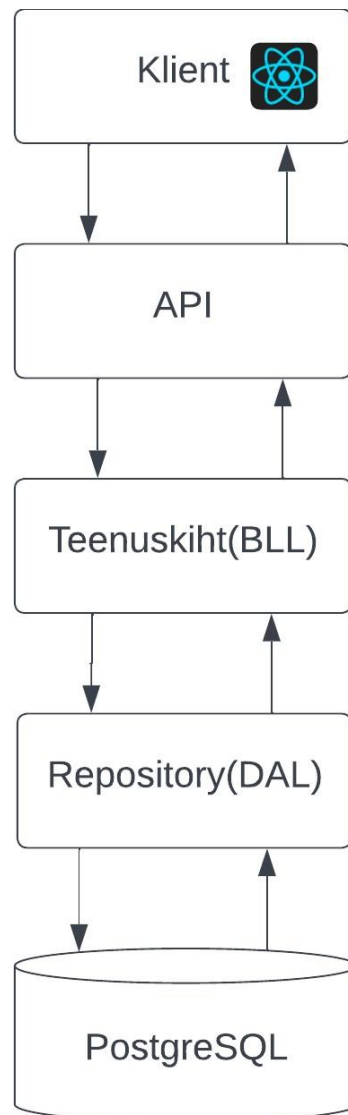
Joonis 3 Näide NuGet paketihaldurist Rideris.

Rideri IDE pakub ka mitmeid lisafunktsioone, nagu tõrkeotsing, koodi analüüs ja refaktoreerimine, mis teevad arendusprotsessi sujuvamaks ja efektiivsemaks. Rakenduse arendamine C# ja ASP.NET Core'iga JetBrains Rideris on seega lihtne ja paindlik, võimaldades arendajal kiiresti alustada ja vastavalt projektivajadustele kohendada, pakkudes samal ajal kõrgekvaliteedilist ja tõhusat arenduskogemust.

5.1.2 Arhitektuur ja struktuuri paika panek

Kvaliteetse rakenduse loomisel, oleks hea kasutada mitmekihilist arhitektuuri, selle kaudu on ka lihtsam eristada, milline osa vastutab mingi osa eest rakenduses ja hiljem on lihtsam leida vigu ja hooldada koodi [18].

Antud rakenduse puhul on serveripoolse rakenduse koodibaas jaotatud kolme peamisse kihti (Joonis 2):



Joonis 4 Serveripoolse rakenduse arhitektuur.

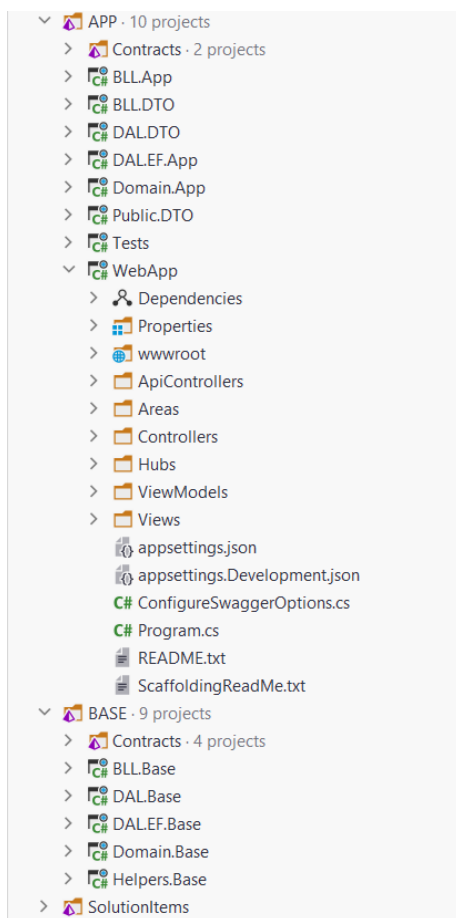
1. Api
2. Teenuskiht(BLL)
3. Repository(DAL)

Selle lähenemisega täidetakse mitmeid olulisi punkte:

- Võtta vastu päringuid ja saata välja vastused;

- Hoiustada andmed andmebaasi ja sealt need väljastada;
- Andmetöötlus

ASP.NET Core aitab suurepäraselt saavutada soovitud tulemust, kus kõikide kihtide vaheline liiklus on esindatud andmemudelite poolt. Kihid suhtlevad vaid kihtidega, mis on enne või pärast neid. Suhtluseks on raamistikus kasutatud sõltuvuste süstimis (ing.k dependency injection) meetodit.



Joonis 5 Serveripoolne rakenduse struktuur.

5.1.3 REST liides

Edasi kirjeldatakse rakenduse liidese kihti, mille ülesanne on hallata päringuid ning edastada vastuseid. See kiht on suhtluse keskpunktiks rakenduste vahel, kasutades REST arhitektuuri põhimõtteid, mis võimaldavad ressurssidega suhtlust standardiseeritud viisil.

Läbi antud liidese kihi, on võimalik infot lisada kasutades GET meetodit, samuti võimaldab see muuta olemasolevat sisu PUT meetodi abil, lisada uusi kirjeid POST meetodi kaudu ja eemaldada ressursse DELETE meetodiga. Järgnevalt on toodud ka mõned näide päringutest:

```
public async Task<ActionResult<IEnumerable<CoursesDTO>>> GetCourses()
{
    var data = await _bll.CoursesService.AllAsync();

    var res = data
        .Select(e => _mapper.Map(e!))
        .ToList();

    return res;
}
```

Joonis 6 GET Päring.

```
public async Task<IActionResult> PutCourses(Guid id, CoursesDTO courses)
{
    if (id != courses.Id)
    {
        return BadRequest();
    }

    var bllCourses = _mapper.Map(courses);
    if (bllCourses != null)
    {
        _bll.CoursesService.Update(bllCourses);
    }
    await _bll.SaveChangesAsync();

    return NoContent();
}
```

Joonis 7 PUT päring.

```

public async Task<ActionResult<CoursesDTO>> PostCourses(CoursesDTO courses)
{
    courses.Id = Guid.NewGuid();
    var bllCourses = _mapper.Map(courses);
    if (bllCourses != null) _bll.CoursesService.Add(bllCourses);
    await _bll.SaveChangesAsync();

    return CreatedAtAction("GetCourses", new { id = courses.Id },
        courses);
}

```

Joonis 8 POST päring.

```

public async Task<IActionResult> DeleteCourses(Guid id)
{
    var courses = await _bll.CoursesService.RemoveAsync(id);
    if (courses == null)
    {
        return NotFound();
    }
    await _bll.SaveChangesAsync();

    return NoContent();
}

```

Joonis 9 DELETE päring.

5.1.4 SignalR rakendamine serveri poolel

SignalR rakendamiseks serveri poolel, tuleb lisada teenused rakenduse teenuste konteinerisse, et kasutada SignalR ASP.NET Core süsteemis.

```

builder.Services.AddSignalR(c =>
{
    c.EnableDetailedErrors = true;
    c.ClientTimeoutInterval = TimeSpan.FromSeconds(30);
    c.KeepAliveInterval = TimeSpan.FromSeconds(15);
});

```

Joonis 10 SignalR lisamine teenuste konteinerisse.

Pärast seda määrati SignalR marsruut, mida saab klient kasutada serveriga ühenduse loomiseks. Lisaks konfigureeritakse veebisoklid.

```

app.MapHub<MyHub>("/myhub");
app.UseWebSockets();

```

Joonis 11 SignalR marsruudi määramine ja veebisoklite konfigureerimis näide.

Seejärel seati paika vajalikud meetodid, et saata ja vastu võtta infot, seoses meeldetuletuste ja realajas toimuvate vestlustega.

```
public async Task SendMessageToAll(string message)
{
    var userName = "Mihkel";
    var timeStamp = DateTime.UtcNow;

    var formattedMessage = $"{userName} at {timeStamp:g} : {message}";
    await Clients.All.SendAsync("ReceiveMessage", formattedMessage);
}
```

Joonis 12 SignalR meetodi näidis, et saata sõnum kõigile grupis.

5.1.5 Turvalisus

Rakenduse turvalisuse osas oleks targem kasutada juba olemasolevaid võimalusi, mida saab pakkuda näiteks ASP.NET. ASP.NET-il on olemas näiteks ASP.Net Core Identity, mis võimaldab turvalisemat registreerimist, sisselogimist ja haldust. Lisaks sellele on ka autoriseerimis ja autentimis valikud, millest autoriseerimist kaustatakse ka selles töös ja autoriseerimine võib olla tulevikuplaanides. Autoriseerimist läheb vaja, et määrata, kes saab mingeid toiminguid teha ja omab ligipääsu teatud kohtadesse. See saab olema vägagi oluline antud rakenduses, kuna tegemist on siiski isiklike vestluste ja informatsiooniga, mis peaks olema näha vaid asjaosalistele.

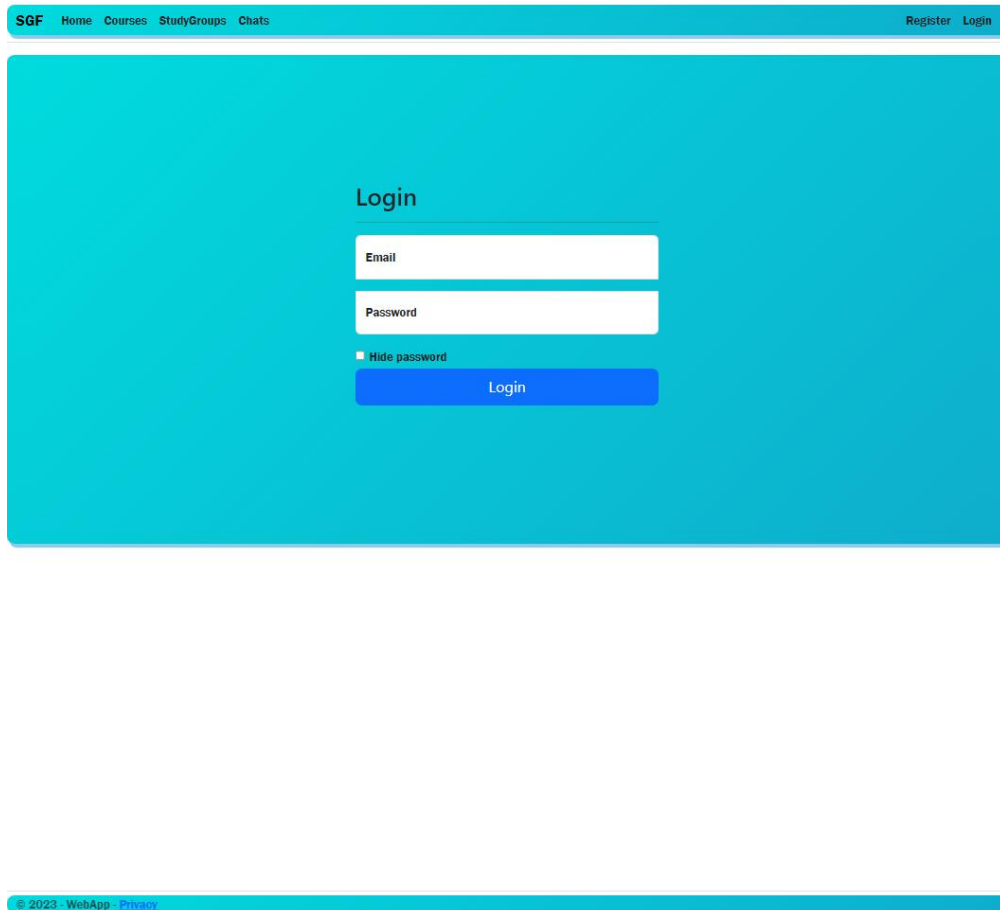
5.2 Kliendipoolse rakenduse arendus

Kliendipoolse rakenduse loomisel kasutati Reacti veebirakenduste raamistik, millega loodi üheleherakendus (*SPA*). React SPA-s laetakse veebileht vaid ükskord ja järgnev suhtlus ei põhjusta lehe uuesti laadimist. React uuendab selle asemel hoopis Dokumendi Objekti mudelit (*DOM*), pöördudes vajadusel andmete järele serverisse, ilma lehte värskendamata. Selline lähenemine võimaldab sujuvat kasutajakogemust.

Luues kliendipoolset rakendust tuleb avada CLI (*Command Line Interface*), kus saaks mugavalt alustada loomist, haldamist ja hooldamist. Kasutades Käsku „npx create-react-app rakenduse nimi“ asendades „rakenduse nimi“ tegeliku nimega, saab uue projekti koos algse struktuuriga. Kui hiljem soovitakse lisada uusi pakette saab kasutada käsku „npm install paketi_nimi“ asendades „paketi_nimi“ realse paketi nimega.

5.2.1 Rakenduse disain

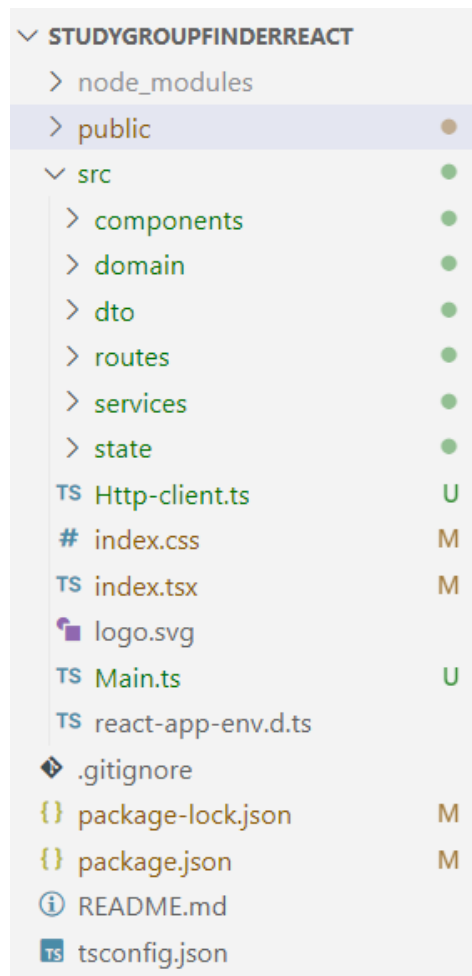
Töös on mitmes kohas mainitud, et rakenduses ei tohiks olla kasutajat segavaid elemente, seega peaks olema ka disain tagasihoidlik. Sellest lähtuvalt peab olema minimaalne kujundus, kus oleks vaid hädavajalik ja pole eriliselt silmapaistvaid ebaolulisi visuaalseid elemente. Rakendus on mõeldud kasutamiseks suurtel ekraanidel ja pole eraldi loodud paremaid tingimusi väiksematele ekraanidele.



Joonis 13 rakenduse üldise disaini näide.

5.2.2 Funktsioonide loomine

Erinevad vaated on jaotatud oma kaustadesse vastavalt nende liigitustele. Järgnevalt on näha eesrakenduse koodibaasi struktuur (Joonis 4).



Joonis 14 Eesrakenduse struktuur.

Struktuuri ehitades tuleb luua kaustad ja teada, mida on nendes kasutades vaja. Näiteks komponentides tuleb luua failid lõppudega .tsx ja .css, kus ühes on vajalik info, mida kuvada ja teises väilmust mõjutavad osad. Selline lähenemine annab võimaluse läbimõelda ja paremini aru saada, kuidas kogu eesrakendus peaks toimima.

Reacti rakendus koosneb komponentidest, kontekstist (*context*) ja konksudest (*hooks*), mis võimaldavad andmete ja funktsionaalsuste edastamist komponentide hierarhias. Reacti konksud, nagu „useState“ ja „useEffect“, võimaldavad kõrgema järgu komponentide (*higher-order components*) funktsionaalsuste kasutamist. Antud töös kasutatakse konkse „useEffect“ koos tingimustega näiteks andmete eellaadimisel, mis käivitab andmete laadimise enne komponendi renderdamist.

```

const { jwtResponse, setJwtResponse } = useContext(JwtContext);
const courseService = new CoursesService(setJwtResponse!);

useEffect(() => {
  if (jwtResponse) {
    courseService.getAll(jwtResponse).then(
      response => {
        console.log(response);
        if (response){
          setData(response);
        } else {
          setData([]);
        }
      }
    );
  }
}, []);

```

Joonis 15 UseEffect näide.

5.2.3 SignalR rakendamine kliendipoolses rakenduses

Kliendipoolses rakenduses, loodi vajalikud komponendid ja teenused, mille kaudu saab suhelda serveriga, et sinna saata ja sealt saada vajalik informatsioon, mida kuvada kasutajatele, kellele oli informatsioon mõeldud.

Teenuse loomiseks on oluline luua ühendus serveriga, selleks tuleb kasutada SignalR kliendi teeki, mis võimaldaks hallata ühendust serveriga. Ühenduse loomiseks, on vaja asünkroonset operatsiooni, mille käigus teostatakse ühenduse loomine ja selle jälgimine.

```

public startConnection = async () => {
  if (this.connection.state === HubConnectionState.Disconnected &&
!this.isStarting) {
    this.isStarting = true;
    try {
      await this.connection.start();
      console.log('SignalR Connected');
    } catch (error) {
      console.error('Error connecting to SignalR:', error);
    } finally {
      this.isStarting = false;
    }
  }
}

```

Joonis 16 SignalR kliendipoolel ühenduse loomise näide.

Sellele järgnevalt saadakse vajalikud meetodid kätte serveri poolelt, et saaks kuvada infot kasutajatele, milleks tuleb luua eraldi komponent, kus võetakse informatsioon, ja määratakse, kuidas seda kuvada.

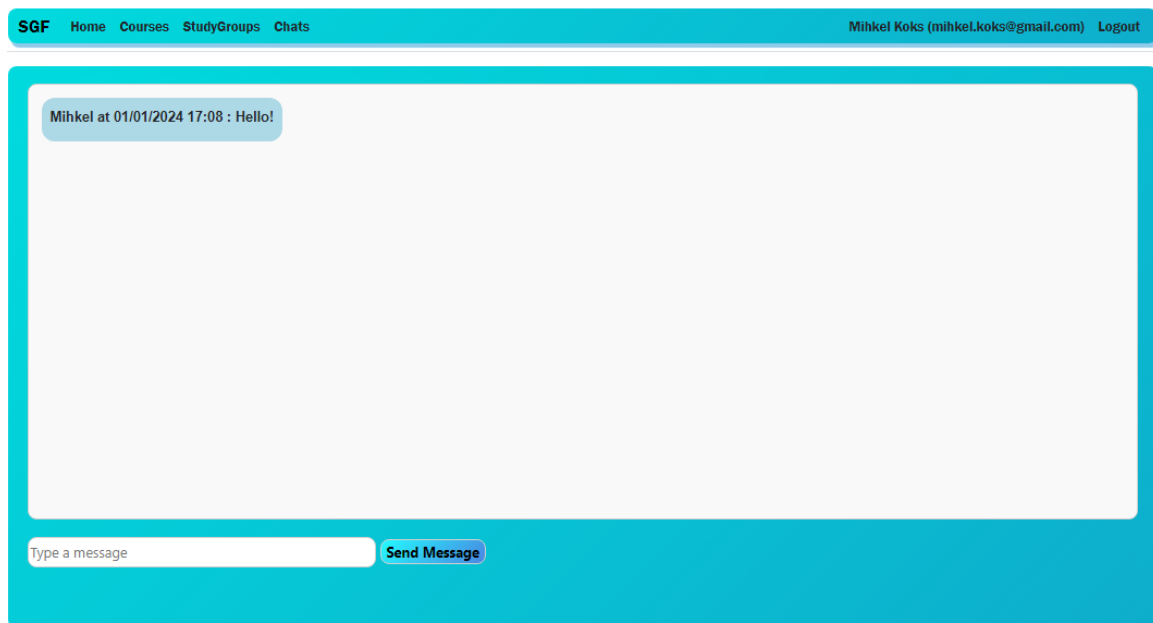
```
return (  
  <div className='mainBox'>  
    <div className="chatbox">  
      {messages.map((message, index) => (  
        <div key={index} className={`messagebubble`}>  
          <p>{message.text}</p>  
        </div>  
      ))}  
    </div>  
    <input  
      type="text"  
      value={inputMessage}  
      onChange={handleInputChange}  
      placeholder="Type a message"  
      onKeyDown={handleKeyPress}  
      className='messageInput'  
    />  
    <button onClick={handleSendMessage} className='sendMessage'>Send  
Message</button>  
  </div>  
);
```

Joonis 17 SignalR komponendist kuvamise näide.

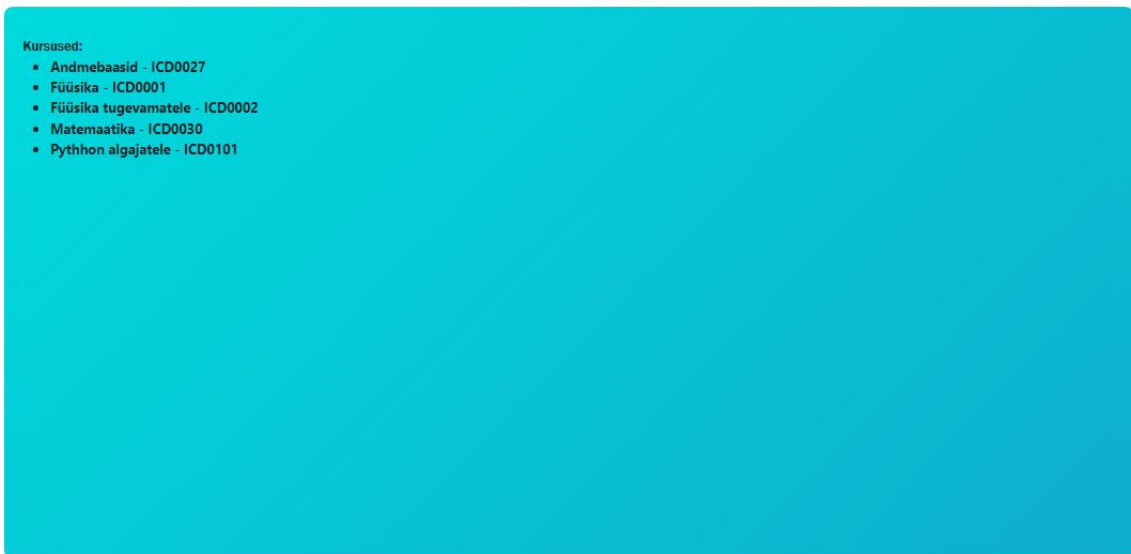
6 Tulemused

Antud lõputöö eesmärgiks oli leida lahendus probleemile, milleks oli autori visioonile vastava rakenduse puudumine. Probleemi lahendamiseks sai analüüsitud olemasolevaid lahendusi, tuues välja nende tugevusi ja nõrkusi. Probleem sai lõputöö käigus lahenduse, mida on võimalik tulevikus täiendada.

Teostuse faasis loodud lahendus, rahuldab autori visiooni, rakendus on tehtud hetkel vaid veebirakendusena ja ei toetatud mobiilil, kuid alguses piisab ka sellest. Hetkel on rakenduses võimalik suhelda kirjavahetuse teel realajas, mis oli rakenduse juures arendajale üks keerulisemaid ülesandeid. Lisaks on võimalik näha erinevaid kursuseid, mis on valikus kasutajatele, mida saab muuta vaid admin õigustega isik. Kõigil kasutajatel on võimalik luua õpigruppe ja lisada teisi enda gruppi ning ka neid sealt eemalda. Veebis on võimalik ka valida keelte vahel, mis toetaks ka välisõpilaste kaasamist gruppidesse.



Joonis 18 Näide suhtluse akenast.



Joonis 19 Näide kursuste valikust.



Joonis 20 Näide õpigruppide haldamise aknast.

7 Kokkuvõte

Bakalaureusetöö „Õpigruppide leidmist toetava veebirakenduse arendus“ keskendub õpilaste koostööd ja koosõppimist toetava veebipõhise lahenduse otsimisele ja selle puudumisel sellise rakenduse loomisele.

Töö alguses anti ülevaade teema aktuaalsusest, millele järgnes probleemi tutvustus. Uuriti, millised hetkel kasutusel olevaid rakendusi on olemas, seati paika nõudmised ja muud vajalikud tingimused. Analüüsist selgus, et oleks vaja siiski luua rakendus, mis vastaks paremini autori visioonile ja alustati planeerimisega. Rakenduse loomise protsessis analüüsiti ka vajalikud tehnoloogiad ja tehti selgeks, miks sobis üks tehnoloogia paremini kui mõni teine.

Antud töö käigus valmis esmane rakendus, mis vastab analüüsis seatud nõuetele. Lisaks on ka selle käigus tekkinud uusi ideid ja lisasid, mida võimalusel tulevikus rakendada. Rakendus peaks aitama kasutajatel teha grupitöid ja grupis õppimist ilma, et oleks liigseid segavaid elemente, kuid oleks kõik esmavajalik olemas ühes kohas.

Kasutatud kirjandus

[1] - facultyfocus.com, „the benefits of study group“. [Võrgumaterjal]. Loetud aadressil: <https://www.facultyfocus.com/articles/course-design-ideas/what-students-can-learn-from-studying-together/> [Kasutatud 29. oktoober 2023].

[2] - zipdo.co, „Social Media Distraction Statistics: Slide Deck“. [Võrgumaterjal]. Loetud aadressil: <https://zipdo.co/statistics/social-media-distraction/> [Kasutatud 31. oktoober 2023].

[3] - Google Inc., „Google Search“, Google Inc., [Võrgumaterjal]. Loetud aadressil: <https://www.google.com/>. [Kasutatud 13. November 2023].

[4] - ednak.com [võrgumaterjal]. Loetud aadressil: <https://ednak.com/thinkbinder/> [kasutatud 01.12.2023]

[5] – www.airsend.io [võrgumaterjal]. Loetud aadressil: <https://www.airsend.io/> [kasutatud 01.12.2023]

[6] – <https://slack.com/> [võrgumaterjal]. Loetud aadressil: <https://slack.com/> [kasutatud 01.12.2023]

[7] - www.codeconquest.com „C# vs Java: Performance, Syntax, Advantages, and Disadvantages“ [võrgumaterjal]. Loetud aadressil: <https://www.codeconquest.com/blog/c-vs-java-performance-syntax-advantages-and-disadvantages/> [kasutatud 07.12. 2023]

[8] – theonetechnologies.com, „Advantages of Using ASP.NET for Web Development“ 01.12.2023 [võrgumaterjal]. Loetud aadressil: <https://theonetechnologies.com/blog/post/advantages-of-using-asp-net-for-web-development> [Kasutatud 01.12.2023]

[9] – www.loginradius.com „JWT Authentication — Best Practices and When to Use“ [võrgumaterjal]. Loetud aadressil: <https://www.loginradius.com/blog/engineering/guest-post/jwt-authentication-best-practices-and-when-to-use/> [kasutatud 08.12. 2023]

[10] - www.codeinwp.com „Angular vs React vs Vue: Which Framework to Choose“ [võrgumaterjal]. Loetud aadressil: <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/> [kasutatud 07.12. 2023]

[11] - www.emizentech.com „React with TypeScript or Javascript: Which is Better?“ [võrgumaterjal]. Loetud aadressil: <https://www.emizentech.com/blog/react-with-typescript-or-javascript.html> [kasutatud 07.12. 2023]

[12] - [www.geeksforgeeks.org „React Functional Components“](https://www.geeksforgeeks.org/reactjs-functional-components/) [võrgumaterjal] Loetud aadressil: <https://www.geeksforgeeks.org/reactjs-functional-components/> [kasutatud 08.12. 2023]

[13] - [ably.com „SignalR vs. WebSocket: key differences and which to use“](https://ably.com/topic/signalr-vs-websocket) [võrgumaterjal]. Loetud aadressil: <https://ably.com/topic/signalr-vs-websocket> [kasutatud 08.12. 2023]

[14] - [krishan-samarawickrama.medium.com „Building a Real-time Application with ASP.NET Core SignalR: A Comprehensive Guide“](https://krishan-samarawickrama.medium.com/building-a-real-time-application-with-asp-net-core-signalr-a-comprehensive-guide-874e975377c8) [võrgumaterjal]. Loetud Aadressil: <https://krishan-samarawickrama.medium.com/building-a-real-time-application-with-asp-net-core-signalr-a-comprehensive-guide-874e975377c8> [kasutatud 08.12. 2023]

[15] - [www.c-sharpcorner.com „Best Practices for Securing Your ASP.NET Core MVC Application“](https://www.c-sharpcorner.com/article/best-practices-for-securing-your-asp-net-core-mvc-application/) [võrgumaterjal]. Loetud aadressil <https://www.c-sharpcorner.com/article/best-practices-for-securing-your-asp-net-core-mvc-application/> [kasutatud 08.12. 2023]

[16] – [learnsql.com, „The Most Popular Databases in 2023“](https://learnsql.com/blog/most-popular-databases-2023/) 24.11.2022 [võrgumaterjal]. Loetud aadressil: <https://learnsql.com/blog/most-popular-databases-2023/> [Kasutatud 05.12. 2023]

[17] – [medium.com, „What Are The Best Databases for Web Applications In 2023“](https://medium.com/@sphinxshivraj/what-are-the-best-databases-for-web-applications-in-2023-40dc3bf9ad0f) 06.06.2023 [võrgumaterjal]. Loetud aadressil: <https://medium.com/@sphinxshivraj/what-are-the-best-databases-for-web-applications-in-2023-40dc3bf9ad0f> [Kasutatud 05.12. 2023]

[18] – [gyires.inf.unideb.hu, „Chapter 4. Layered Architecture for Web Applications“](https://gyires.inf.unideb.hu/GyBITT/08/ch04.html) [võrgumaterjal]. Loetud aadressil: <https://gyires.inf.unideb.hu/GyBITT/08/ch04.html> [kasutatud 05.12. 2023]

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Mihkel Koks

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Õpigruppide leidmist toetava veebirakenduse arendus“, mille juhendaja on Märt Kalmo
 - 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.

07.12.2023

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 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 – Olemasolevate rakenduste otsimiseks kasutatud fraasid

Inglise keeles:

- Group study online
- Online study app
- Group up to learn
- Study together
- Study with friends
- Study with other students
- Group online rooms
- Place to share notes and chat
- Team collaboration

Lisa 3 – Otsingus leitud olemasolevad rakendused

Veebirakendused:

ThinkBinder

Studytogether.com

Studystream.live

Studyverse.live

Csw.live

Mooclab.club

flock

substack

slack

airsend

GroupRoom