

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Martin Aleksander Kostjajev 194008IADB

Veebirakenduse arendus restoranist toidu tellimiseks

Bakalaureusetöö

Juhendaja: German Mumma

Magistrikraad

Tallinn 2022

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: Martin Aleksander Kostjajev

13.05.2022

Annotatsioon

Antud bakalaureusetöö eesmärk on luua restoranist toidu tellimiseks veebirakendus. Rakendus pakub kasutajale võimalust tellida restoranist toitu kodust lahkumata.

Veebirakenduse arendamisel on kasutatud *ReactJS* ja *NextJS* raamistikud, *JavaScript* programmeerimiskeelt koos *NestJS* raamistikuga ja *MongoDB* andmebaasi.

Töö esimeses osas analüüsib autor olemasolevaid alternatiivseid restoranidest toidu tellimisega seotud veebirakendusi. Teises osas toob autor välja eesmärkide saavutamiseks valitud tehnoloogiad. Kolmandas osas kirjeldab autor rakenduse funktsionaalsust ja teostamist. Neljandas osas analüüsib autor rakenduse kasutajatestimise tulemusi ning toob välja peamised ideed, mida võib tulevikus rakenduse funktsionaalsuses teha.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 6 peatükki, 15 joonist.

Abstract

Web Application Development for Ordering Food from a Restaurant

The goal of this bachelor's thesis is to create a web application for ordering food from a restaurant. Application allows user to order food from the restaurant without leaving home.

The web application was developed using ReactJS and NextJS frameworks, JavaScript programming language with NestJS framework and MongoDB database.

In the first part of the thesis, author analyses existing alternative web applications related to ordering food from restaurants. In the second part, author outlines the technologies chosen to achieve goals and the reasons for the choices made. In the third part, author describes functionality and development of the application. In the fourth part, author analyses the results of user testing of the application and outlines the main ideas that can be done in the functionality in the future.

The thesis is in Estonian and contains 28 pages of text, 6 chapters, 15 figures.

Lühendite ja mõistete sõnastik

<i>SPA</i>	<i>Single-page application</i> . Dünaamiline veebirakendus, millel puudub vajadus lehe uuesti laadimiseks
<i>HTML</i>	<i>Hypertext Markup Language</i> . Veebilehtede märgistuskeele standard
<i>Clean Architecture</i>	Tarkvara disaini filosoofia, mis eraldab disaini elemendid ringitasanditeks
<i>CSS</i>	<i>Cascading Style Sheets</i> . Märgistuskeeles kirjutatud HTML dokumendikuvamise stiili kirjeldav keel
<i>API</i>	<i>Application Programming Interface</i> . Rakendusliides
<i>DAO</i>	<i>Data Access Object</i> . Objekt, mille abil saab ligi andmetele
<i>DAL</i>	<i>Data Access Layer</i> . Andmetele kättesaadavuse kiht
<i>DTO</i>	<i>Data Transfer Object</i> . Objekt, mis kannab andmeid protsesside vahel
<i>BLL</i>	<i>Business Logic Layer</i> . Äriloogika kiht
<i>DBMS</i>	<i>Database Management Systems</i> . Tarkvarasüsteemid, mida kasutatakse andmete salvestamiseks, toomiseks ja päringute käitamiseks

Sisukord

1 Sissejuhatus.....	9
2 Olemasolevad rakendused restoranidest toidu tellimiseks	12
2.1 Sushi Guru	12
2.2 O'Sushi	13
2.3 Tokyo55.....	13
2.4 Olemasolevate rakenduste analüüs	13
3 Tehnoloogiad	15
3.1 Veebirakendus	15
3.2 NextJS.....	16
3.3 Node.js.....	17
3.4 NestJS	17
3.5 MongoDB	18
4 Veebirakenduse arendus restoranist toidu tellimiseks	19
4.1 Tagarakenduse üldloogika	19
4.2 Eesrakenduse üldloogika	20
4.3 Rakenduse funktsionaalsus	23
4.3.1 Sisselogimine	23
4.3.2 Registreerimine.....	24
4.3.3 Toodete filtreerimine	25
4.3.4 Toote leht.....	26
4.3.5 Mobiilirakenduse allalaadimine.....	26
4.3.6 Kasutajaprofiil ja tellimuste ajalugu.....	27
4.3.7 Ostukorv	27
4.3.8 Tellimuse olek	28
5 Restoranist toidu tellimise veebirakenduse analüüs	29
5.1 Rakenduse testimine	29
5.2 Rakenduse testimise tulemuste analüüs.....	30

5.3 Rakenduse täiendused	31
6 Kokkuvõte.....	33
Kasutatud kirjandus	35
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	37
Lisa 2 – Andmebaasi skeem	38
Lisa 3 – Äriprotsesside voog	39
Lisa 4 – Kasutajalood	40

Jooniste loetelu

Joonis 1. Client-Side-Rendering.	16
Joonis 2. Server-side Rendering.....	16
Joonis 3. Tagarakenduse kaustade struktuur.	20
Joonis 4. Eesrakenduse struktuur.	22
Joonis 5. Sisselogimise vaade.	23
Joonis 6. Registreerimise vaade.	24
Joonis 7. Toodete filtreerimine.....	25
Joonis 8. Toote leht.	26
Joonis 9. Mobiilirakendus.	26
Joonis 10. Kasutajaprofiil ja tellimuste ajalugu.	27
Joonis 11. Ostukorv.....	27
Joonis 12. Tellimuse olek.....	28
Joonis 13. Andmebaasi skeem.	38
Joonis 14. Äriprotsesside voog.....	39
Joonis 15. Kasutajalood.....	40

1 Sissejuhatus

On olemas tuttavad, kes tahavad avada uut restorani ja neil on vajadus veebirakenduse järele, millega restorani kliendid saaksid toitu koju tellida.

Seoses COVID 19 epideemia puhanguga tekkis vajadus tellida toitu kodust lahkumata, mis vähendab oluliselt elanikkonna haigestumust. Loomulikult võimaldab see tava ettevõtetel (eriti restoranidel) mitte kaotada sissetulekuid ja säilitada tarbijaskonna. Samuti rakendused toidu koju tellimiseks on väga aktuaalsed ka puuetega inimestele.

Käesoleva lõputöö eesmärgiks on luua veebirakendus, mis võimaldab mugavalt tellida restoranist toitu kodust lahkumata.

Varustada veebirakendus kõige vajalikuma funktsionaalsusega nii mugavaks toidu tellimiseks kui ka tellimuste ajaloo vaatamiseks.

Rakenduse võimaldab kiiresti ja mugavalt leida toote, mida kasutaja soovib tellida, samuti valida sobiva tarne- ja makseviisi. Näiteks kasutaja soovib tellida toitu kindlast kategooriast, mistõttu on rakendusel võimalus kategooriate kaupa filtreerida. Samuti soovib kasutaja näiteks tellimuse eest tasuda kättesaamisel sularahas või siis krediitkaardiga.

Ühine tagarakenduse osa ja prototüübi arendamine teise lõpetajaga koos. Kuid projekt ise ei lõpe veebirakenduse arendamisega, vaja on ka mobiilirakendust ja just sellega hakkab tegelema teine lõpetaja. Tagarakenduse väljatöötamise protsess viidi läbi ühiselt, et kõik lõpetajad oleksid toimuvaga kursis.

Töö peamiseks tulemiks on toimiv veebirakendus. Valminud rakendusel on olemas andmebaas, tagarakendus ja eesrakendus.

Selle veebirakenduse tagarakendus on tehtud keeles *JavaScript*, kasutades *NestJS* raamistikku. *NestJS* on raamistik, mis kiirendab ja lihtsustab *Node.js* tarkvaraplatformil põhinevate skaleeritavate serverirakenduste arendamist. See kasutab progressiivset *JavaScript* -i, mida toetab täielikult *TypeScript*^[7].

Rakenduse eesrakendus on realiseeritud kasutades *NextJS* raamistikku. *NextJS* on *React* põhinev raamistik, mis võimaldab tänu täiendavatele eelrenderdusfunktsioonidele luua kiirema jõudluse ja parema kasutajakogemusega veebirakendusi^[9]. Samuti rakendus töötab populaarsemates veebibrauserites.

Andmebaasi tehnoloogiaks on valitud avatud lähtekoodiga tarkvara *MongoDB*. *MongoDB* andmebaasihaldussüsteem, mis töötab dokumendipõhise andmemudeliga. Erinevalt relatsioonilistest *DBMS*-idest ei vaja *MongoDB* tabeleid, skeeme ega eraldi päringukeelt. Teavet säilitatakse dokumentide või kogude kujul^[3].

Veebirakendus töötab „*single-page application*” põhimõtte järgi. See tähendab, et kui kasutaja veebilehe avab, laadib see korraga kogu rakenduse koodi, kuid see näitab ainult konkreetset moodulit, ehk veebilehe osa, mida kasutaja vajab. Ja vajadusel laadib dünaamiliselt serverist vajaliku sisu ja kuvatakse neid juba laetud veebilehe sees^[10].

Enne seda tööd oli koostatud tehniline ülesanne koos kõigi nõuetega:

- registreerimine ja autoriseerimine
- võimalus vaadata toodete loendit
- toodete filtreerimine
- võimalus esitada tellimus
- võimalus valida sobiv makseviis
- registreeritud kasutaja näeb oma tellimuste ajalugu
- võimalus kontrollida tellimuse olekut
- administraatoril on võimalus veebirakenduses restorani menüüd hallata

Esimeses peatükis kirjeldab autor valitud teema selgitust ja aktuaalsust. Samuti kirjeldab tehniline ülesanne koos kõigi nõuetega, mida tuleb täita selles töös. Teises peatükis kirjeldab autor juba olemasolevaid alternatiivseid rakendusi, mis võimaldavad tellida toitu teistest restoranidest. Toimub nende analüüs, tugevate ja nõrkade külgede kirjeldus. Kolmandas peatükis kirjeldab autor valitud tehnoloogiaid ja nende sobivust probleemi lahendamiseks. Neljandas peatükis kirjeldab autor veebirakenduse funktsionaalsust ja selle teostamist. Viiendas peatükis kirjeldab autor veebirakenduse testimise ja klientide

tagasiside analüüsimise tulemusi. Samuti on kirjeldatud ideed, mida saab tulevikus sellesse veebirakendusse lisada.

2 Olemasolevad rakendused restoranidest toidu tellimiseks

Käesolevas peatükis kirjeldatakse juba olemasolevaid veebirakendusi restoranidest toidu, meie puhul sushi tellimiseks. Iga veebirakenduse puhul tuuakse esile selle tugevad ja nõrgad küljed.

Analüüsiks valis autor kolm juba olemasolevat sushi tellimise rakendust:

- Sushiguru – <https://sushiguru.ee>
- O'sushi – <https://osushi.ee>
- tokyo55 – <https://tokyo55.ee>

2.1 Sushi Guru

Sushi Guru veebileht annab võimaluse vormistada tellimust kasutades roogade sorteerimist kategooriate kaupa, samuti kasutada rakendusi App Store ja Google Play platvormidel.

Kõik vajalikud ettevõtte andmed – aadress, kontaktandmed, lahtiolekuajad, tagasiside, lingid sotsiaalvõrgustikele, keelevahetus – on saadaval veebilehe päises, mis teeb restorani kohta info kogumise lihtsamaks. Roa detaile vaadates saate vaadata toodete koostist.

Visuaalne komponent on kõige arusaadavam ja esteetiliselt kompetentsem, piltide kujundamisel valitseb minimalism ja kaasaegne disain.

Kuid on ka puudusi: kategooriate vahel liikudes tuleb lehtede laadimist mõnda aega oodata, mistõttu pilte kohe ei kuvata ja puudub võimalus kasutajat registreerida ja tellimuste ajalugu vaadata.

2.2 O'Sushi

O'Sushi veebileht soovib avalehel sooduspakkumisi, ülejäänud tooted on toodud vasakul lisapaneelil, mida ei saa peita. Peale toidukorvi tellimust vaatama minekut suletakse roogade kategooria leht, kui ostja soovib osta lisaportsjonit, siis tuleb see roog uuesti otsida kategooriast, mida autor ei täheldanud Sushi Guru puhul (seal avanes eraldi aken).

Erinevalt Sushi Guru-st on veebilehe kujundus tehtud küll minimalistikus stiilis ja kaasaegse disainiga, kuid tegemist on sisutiheda kujundusega.

2.3 Tokyo55

Kui soovite ostukorvi lisada roogasid, avaneb aken, et see kaup on valitud, kuid kahjuks on ka lehe värskendamisel korv tühi. Uuel lisamisel kuvatakse kasutajale teade, et toode on edasi lükatud, kuid ostukorvi on veel tühi. Tellimust on võimalik esitada nii registreeritud kui ka registreerimata kasutaja, kuid tellimuse vormistamiseks on vaja veebilehe funktsionaalsusega palju manipuleerida. Mõned nupud ei ole klõpsatavad. Mobiilseadme jaoks pole rakendusi muudel platvormidel. Veebilehe kujundus on kaootiline, kõik pildid on tumeda taustaga, mis ei harmoneeru veebilehe kontseptsiooniga.

Kogenematu kasutaja ei saa iseseisvalt tellimust esitada, selle esitamiseks peate helistama restorani telefoni teel.

2.4 Olemasolevate rakenduste analüüs

Analüüsides sarnaseid sushi tellimise veebirakendusi, saame kokku võtta:

- mõne veebirakendusi puhul ei ole võimalik kontot registreerida, samuti puudub võimalus vaadata tellimuste ajalugu,
- mitte kõigil restoranidel olemas oma mobiilirakendus
- mõnel veebirakendusel on lisafunktsioonid, millega kogenematu kasutaja hakkama ei saa
- mitte kõigi veebirakendusel olemas võimalus vaadata oma tellimuse olekut

- mitte kõik veebirakendused annavad võimalust valida sobivat tarne- ja makseviisi
- mõned veebirakendused ei tööta hästi erinevates brauserites ja erineva suurusega akendes
- paljudel rakendustel puudub kaheastmeline autentimine, mis on privaatsusprobleem
- mitte kõigil on olemas võimalus vigade või küsimuste korral veebitoega ühendust võtta

Samuti on täheldatud, et paljud restoranid ohverdavad veebirakenduste jõudluse kõrge funktsionaalsuse nimel. Selle töö autori arvates väga sageli on need funktsioonid üleliigsed ja neid kasutatakse vähe.

Autori arvamusel peaks veebirakenduste üks põhiomadusi olema kasutusmugavus, et ka kõige kogenematamad kasutajad saaksid tellimuse esitada, sest selliste rakenduste puhul on väga oluline, et ostuprotsess oleks võimalikult lihtne ja mugav.

3 Tehnoloogiad

Käesolevas peatükis on kirjeldatud rakenduse arendamisel kasutatud tehnoloogiad ning nende valimise põhjused.

3.1 Veebirakendus

Restoranist toidu tellimise rakenduse teostamise platvormiks on valitud veebirakendus. Tänapäeval ei ole veebirakendused toidu tellimise nii populaarsed kui mobiilirakendused, kuid neil on oma tugevad ja nõrgad küljed. Veebirakenduste tugevus seisneb selles, et neid on üldiselt lihtne kasutada. Veebirakenduse kasutamiseks ei pea kasutaja midagi uuendama ega alla laadima, piisab vaid brauserist ja internetiühendusest, mis tähendab, et veebipõhised rakendused universaalselt kasutatavad. Samuti on laiem valik veebirakenduste arendamiseks sobivaid programmeerimiskeeli ja raamistikke^[4]. Tänu sellele, et kasutaja informatsiooni salvestatakse serverisse, on see saadaval ka teisest seadmest ennast identifitseerides ja veebilehele sisenedes, näiteks saab kasutaja vaadata oma tellimuste ajalugu ka teistest seadmetest.

Kuid kahjuks on veebirakendustel ka miinuseid. Veebirakenduse töötamiseks on vaja internetiühendust. Statistika näitab, et häkkimine ja identiteedivargused on iga päevaga tõusuteel, mistõttu ei ole veebirakendused just kõige turvalisemad. Veebirakenduste peamiseks puuduseks on nende kehv jõudlus. Teised rakendused näitavad suurusjärgus paremat jõudlust. Lisaks on võimalik rakenduse arendamisel kasutada kõiki konkreetse platvormi spetsiifilisi funktsioone^[4].

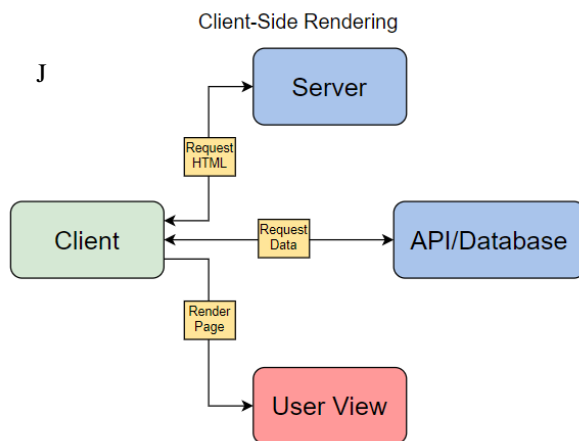
Rakendused pakuvad arendajatele palju võimalusi, kuid veebirakendused on palju odavamad, neid on lihtsam juurutada ja paljudes seadmetes lihtsam hooldada.

3.2 NextJS

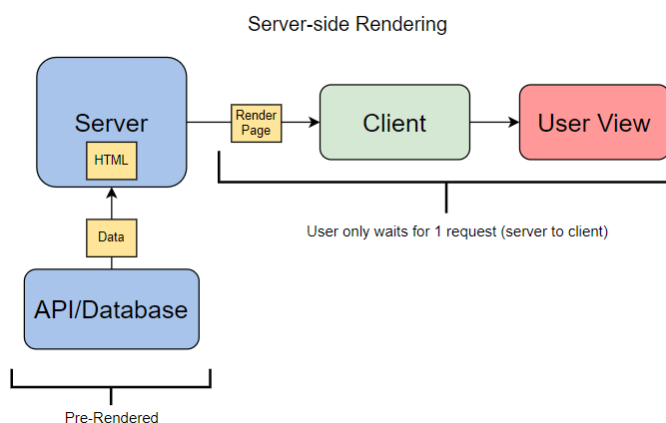
Restorani toidu tellimise veebirakenduse eesrakenduse tehnoloogiaks on *React* baasil töötav *NextJS*. *React* omakorda on *JavaScript* teek kasutajaliideste loomiseks.

NextJS on *React* - põhine raamistik, mis võimaldab teil luua suurema jõudluse ja parema kasutajakogemusega veebirakendusi koos täiendavate eelrenderdusfunktsioonidega, nagu täielik serveripoolne renderdus (SSR) ja staatiline lehtede genereerimine (SSG). Seega kasutavad *NextJS* rakendused täielikult *React* teegi eeliseid ja lisavad lihtsalt lisafunktsioone^[9].

Kogu serveris oleva teabega saab klient lehe *HTML*-koodi eelvaate. Klient saab saata serverile ühe päringu ja saada täieliku *HTML*-lehe, selle asemel, et taotleda iga komponenti kliendilt eraldi^[9].



Joonis 1. *Client-Side-Rendering*.



Joonis 2. *Server-side Rendering*.

NextJS rakenduste eeliseks on see, et tänu sissehitatud serveripoolsele renderdamisele

laaditakse need oluliselt kiiremini kui *React* rakendused. Samuti ka sisseehitatud tugi lehtede marsruutimisele, *CSS*, *JavaScript* ja *TypeScript* Ainus *NextJS* tõeline nõrk külg on see, et see on iseseisev raamistik, mis tähendab, et sellel on konkreetne meetod ja tööriistade komplekt, mida ta soovib oma rakenduste koostamiseks kasutada

3.3 Node.js

Tagarakenduse arendamiseks kasutatakse *Node.js*. *Node.js* on avatud lähtekoodiga *JavaScript* raamistik, mis on üles ehitatud Chrome V8 mootorile. See võimaldab kirjutada serveri koodi veebirakenduste ja dünaamiliste veebilehtede ning käsureaprogrammide jaoks. Platvorm põhineb sündmuste põhisel mudelil, millel on mittemblokeerivad I/O-toimingud, mis muudab selle tõhusaks ja kergeks.

Lihtsamalt öeldes on *Node.js* C++ rakendus, mis võtab sisendiks *JavaScript* ja käivitab selle. Arvuti sisend-/väljundseadmetega suhtlemiseks pakub platvorm oma liidest C++ keeles. Seega muudab platvorm spetsiaalse *JavaScript* skriptikeele üldotstarbeliseks keeleks, nii et *Node.js* saab kirjutada mistahes arvutiprogramme.

Node.js võimaldab koodi kirjutamiseks kasutada ühte *JavaScript* keelt nii kliendi kui ka serveri poolel. Need *Node.js* funktsioonid on olulised sündmustel põhinevate reaalajas rakenduste arendamiseks^[8].

3.4 NestJS

NestJS on raamistik, mis kiirendab ja lihtsustab *Node.js* tarkvaraplatvormil põhinevate skaleeritavate serverirakenduste arendamist. See kasutab progressiivset *JavaScript*, toetab täielikult *TypeScript*.

NestJS pakub selge, arusaadava ja funktsionaalse ülesehitusega rakendust, mis vastab kõigile *Clean Architecture* põhimõtetele. Selline rakendus on stabiilne ning seda saab hõlpsasti laiendada ja skaleerida. Võimaldab arenduse ajal kasutada mis tahes kolmanda osapoole teeki, ühendab kõik andmebaasid ja teenused *Node.js* ja kiirendab arendusprotsessi ning lihtsustab valmis rakenduse hooldust^{[1][7]}.

3.5 MongoDB

MongoDB on juhtiv dokumentide andmebaas. See on üles ehitatud hajutatud, laiendatavale arhitektuurile ja sellest on saanud terviklik pilvepõhine platvorm andmete haldamiseks ja rakendustesse edastamiseks^[5]. *MongoDB* ei kasuta skeeme nagu relatsiooniandmebaasid, mis parandab süsteemi üldist jõudlust. *MongoDB* kasutab tabelite asemel kogusid. Need sisaldavad erinevat tüüpi andmekogumeid^[3].

Alternatiiv *MongoDB* süsteemile on *PostgreSQL*, mis on samuti üks populaarsemaid andmebaasi haldamise süsteeme. *PostgreSQL* on kaljukindel avatud lähtekoodiga ettevõtetasemel SQL-andmebaas, mis on oma võimalusi laiendanud 30 aastat^[5].

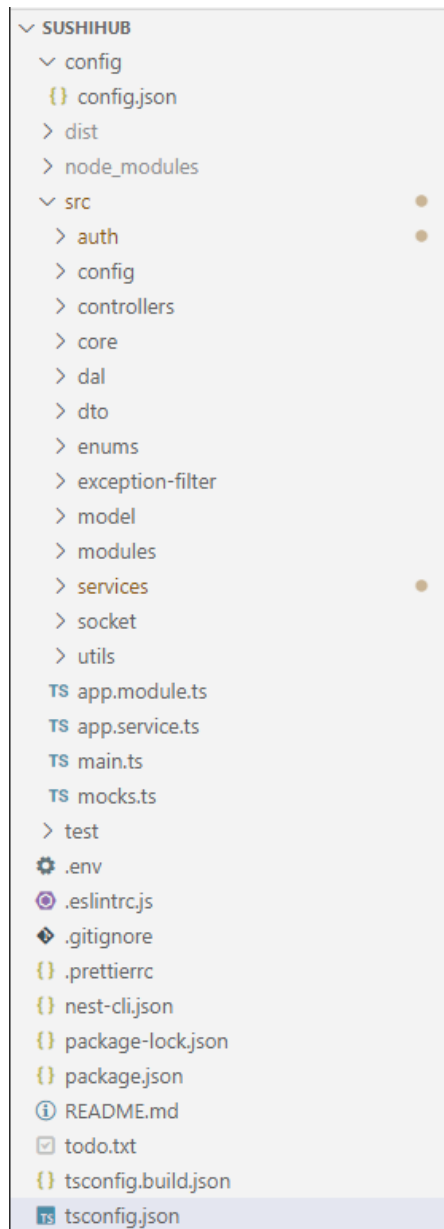
4 Veebirakenduse arendus restoranist toidu tellimiseks

Käesolevas peatükis on kirjeldatakse veebirakenduste arendusprotsessi ja selle funktsionaalsust. Lisatud on rakenduse andmebaasi skeem (Lisa 1) ja äriprotsesside voog (Lisa 2).

4.1 Tagarakenduse üldloogika

Tagarakenduse projekti struktuur on jagatud erinevatesse kaustadesse. Iga kaust vastutab ühe rakenduse osa eest (Joonis 3):

- „*auth*” sisaldab endas komponente ja teenuseid, mis vastutavad rakenduse autentimise eest
- „*controllers*” sisaldab kõiki saadaolevaid kontrollereid
- „*core*” sisaldab kõiki erinevaid abstraktsioone ja *MongoDB* seotud teenuseid, hoidlaid ja mooduleid
- „*dal*” on *Data Access Layer* ehk andmetele kättesaadavuse kiht
- „*dto*” on *Data Transfer Object*, Andmete edastamise toimingute
- „*enums*” sisaldab kõiki muutumatuid väärtusi ja nimetusi
- „*exception-filter*” see on *middleware*, mis vastutab kõigi vigade käsitlemise eest
- „*model*” sisaldab kõiki andmebaasi olemi mudeleid
- „*modules*” sisaldab klassid, millele on lisatud *@Module()* dekoraator. Dekoraator *@Module()* pakub metaandmeid, mida *NestJS* kasutab rakenduse struktuuri korraldamiseks^[6]
- „*services*” sisaldab kõiki saadaolevaid teenuseid, *Business Logic Layout*



Joonis 3. Tagarakenduse kaustade struktuur.

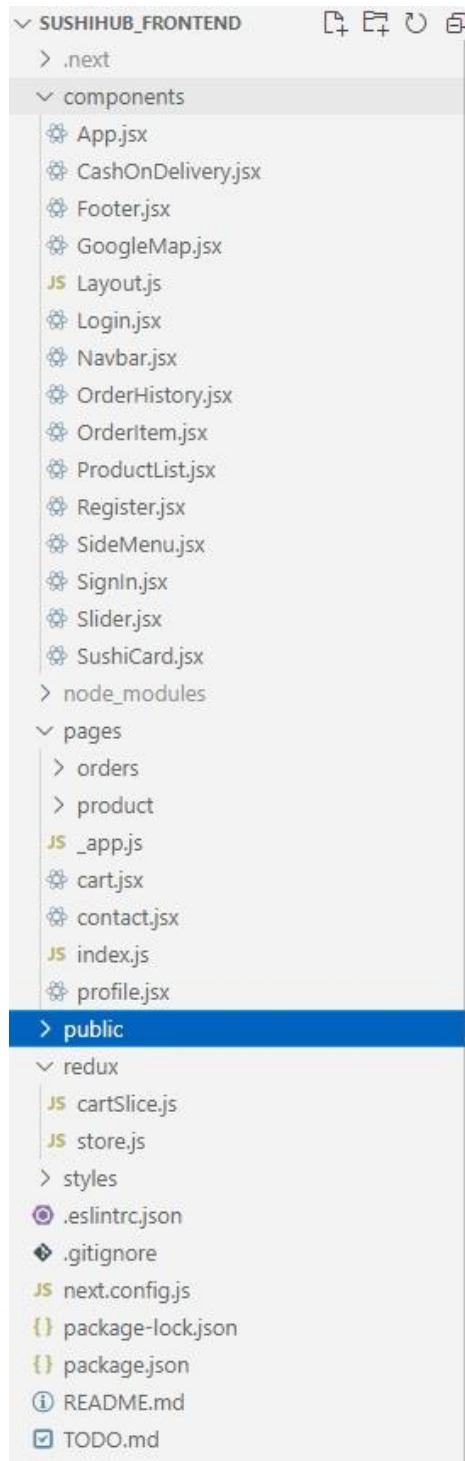
4.2 Eesrakenduse üldloogika

Eesrakenduse projekt on loodud kasutades *NextJS* raamistikku. Igal komponendil on oma CSS-fail. Komponentid on funktsionaalsuse järgi jagatud erinevatesse kaustadesse (Joonis 4):

- „*components*” kaust sisaldab endas kõiki saadaolevaid komponente nagu navigeerimine, sisselogimine, külghüüde toodete filtreerimiseks, liugur ja nii

edasi. Selliseid komponente saab manustada mis tahes lehele või muusse komponenti. Näiteks tootekaardi kujunduskomponent, mis omakorda paigutatakse kõikide veebilehe toodete komponendi sisse, mis on manustatud avalehele. Komponentidel on ka olemas fail *Layout*. See fail sisaldab komponente, mis on saadaval absoluutselt kõigil lehtedel. Näiteks navigeerimine ja *Footer*

- „*pages*” kaust sisaldab eraldi lehti, aga ka lehti, mis võimaldavad id abil minna konkreetse toote lehele. Näiteks igal tootel on oma id ja sellel tootel klõpsates avaneb uus leht, mis sisaldab infot ainult selle toote kohta. Nii nagu tavalised lehed on loodud sisaldama üksikuid komponente või teavet. Näiteks „*contact*“, mis sisaldab teavet restorani kohta, samuti GoogleMap komponent, mis omakorda vastutab kaardi joonistamise ja kuvamise eest. „*Pages*“ sisldab ka faili nimega „*_app*“. See on kõige olulisem fail, mis sisaldab endas *Layout* ja muud täiendused, mis mõjutavad kogu veebilehte
- „*public*” kaust sisaldab kõiki pilte, logod, märgid, ikoonid
- „*redux*“ sisaldab *Redux* ja selle reduktoriga töö tegemiseks vajalikke faile. *Redux* on tööriist andmete oleku ja kasutajaliidese haldamiseks olemirohketes *JavaScript* -i rakendustes. See on *JavaScript* i- teek^[13]. Selles töös täidab ta kaupade ostukorvi lisamise funktsiooni ja haldab ka nende kogust.
- „*styles*” sisaldab kõiki komponentide ja lehtede css-fail. Igal komponendil on oma eraldi css-fail. Kuid on ka erandeid, näiteks sisselogimine ja registreerimine. Neil on sama stiil ja nad kasutavad sama css-faili



Joonis 4. Eesrakenduse struktuur.

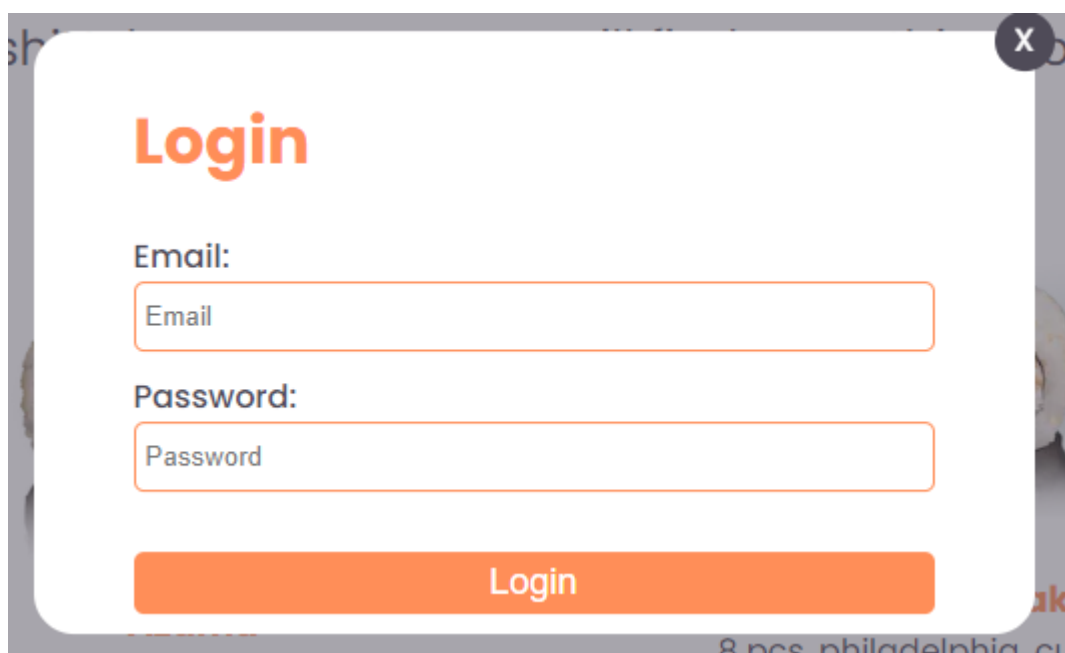
4.3 Rakenduse funktsionaalsus

Kasutajalugu on tarkvara funktsiooni mitteametlik üldine selgitus, mis on kirjutatud lõppkasutaja vaatenurgast. Selle eesmärk on sõnastada, kuidas tarkvara funktsioon kliendile väärtust pakub^[12]. Projekti raames koostatud kasutajalood on toodud lisas 4.

Igal kasutajalool on oma prioriteet ja eesmärk mida see loo lõpuks tegema peaks.

4.3.1 Sisselogimine

Klõpsates nuppu *Login*, saab kasutaja sisse logida olemasolevale kontole (Joonis 5).

The image shows a login form with a white background and orange accents. At the top left, the word "Login" is written in a large, bold, orange font. Below it, there are two input fields: "Email:" followed by a text box containing the placeholder "Email", and "Password:" followed by a text box containing the placeholder "Password". At the bottom of the form is a wide, orange button with the word "Login" in white text. The form is set against a dark grey background that shows parts of other elements, including a close button (an 'X' in a circle) in the top right corner and some text at the bottom right.

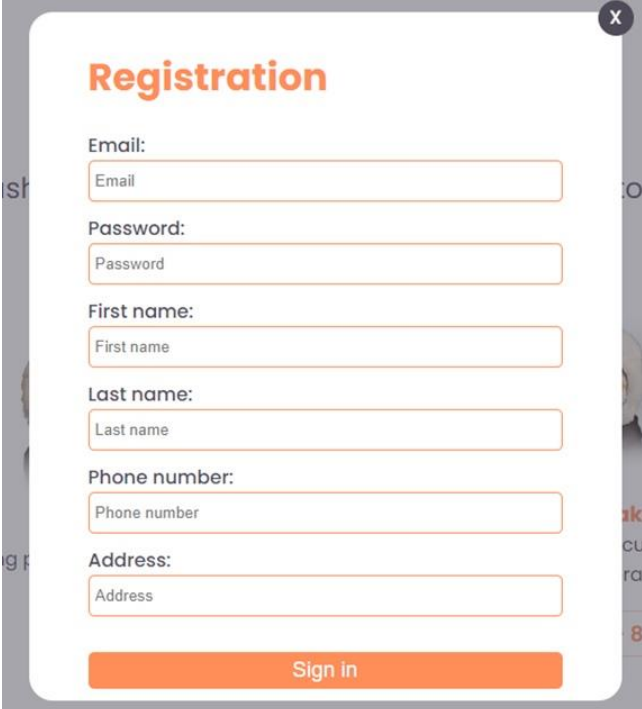
Joonis 5. Sisselogimise vaade.

Kasutajal on võimalik sisestada enda email ja parool, mis saadetakse autentimise eesmärgil teenuskihti. Juhul, kui kasutaja info on korrektne, logitakse ta sisse ja suunatakse veebilehe avalehele. Navigeerimismenüüsse ilmuvad lisafunktsioonid, näiteks Minu profiil.

Kui administraator logis sisse, suunatakse ta veebilehe haldusesse, kus avatakse talle eraldi funktsioonid.

4.3.2 Registreerimine

Klõpsates nuppu *Registration*, kasutaja saab endale konto luua (Joonis 6)

A screenshot of a registration form titled "Registration" in orange text. The form is contained within a white rounded rectangle with a grey border and a close button (X) in the top right corner. It features several input fields: "Email", "Password", "First name", "Last name", "Phone number", and "Address". Each field has a light orange border and a placeholder text. At the bottom of the form is a prominent orange button labeled "Sign in".

Registration

Email:
Email

Password:
Password

First name:
First name

Last name:
Last name

Phone number:
Phone number

Address:
Address

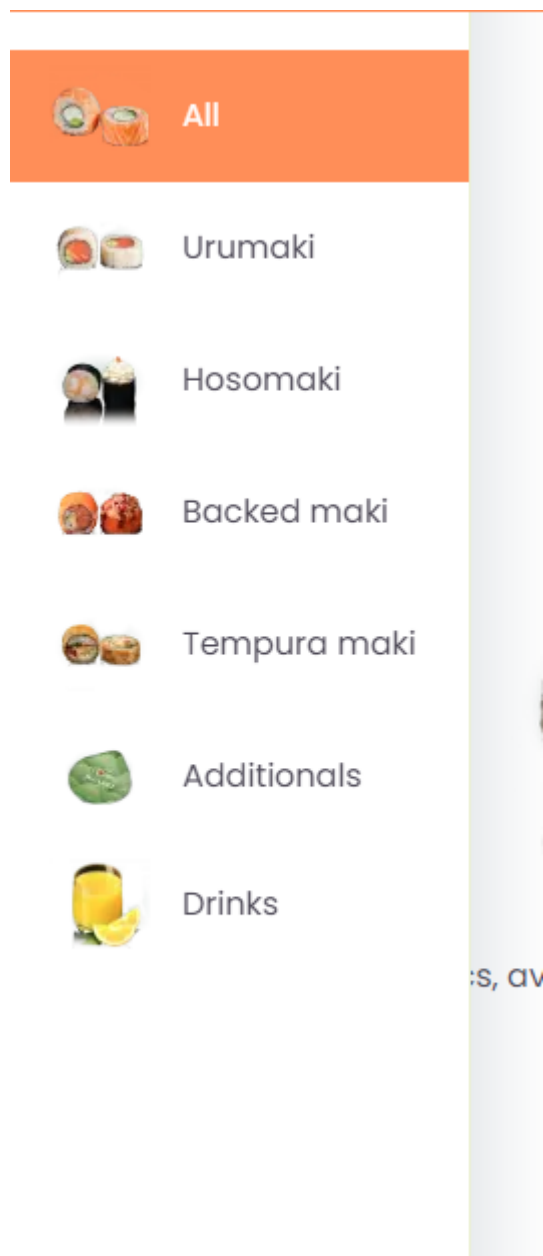
Sign in

Joonis 6. Registreerimise vaade.

Kasutaja peab sisestama enda email, parool, eesnimi, perekonnanimi, telefoni number ja aadress. Edaspidi võetakse seda infot tellimuse vormistamisel arvesse, kuid igal juhul võib näiteks aadress olla teisiti määratud.

4.3.3 Toodete filtreerimine

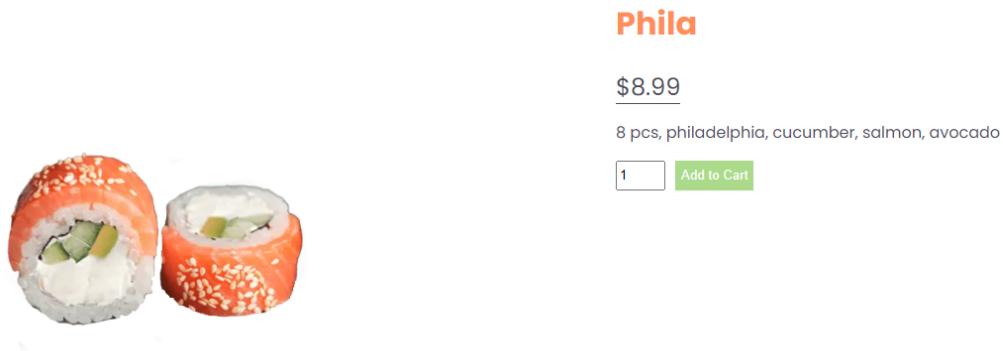
Kasutaja saab tooteid kategooriate kaupa filtreerida. Selleks, peab kasutaja valima menüüst vasakult kategooria (Joonis 7).



Joonis 7. Toodete filtreerimine.

Vaikimisi on valitud automaatselt kategooria Kõik (*All*), mis näitab menüüs kõiki saadaolevaid tooteid.

4.3.4 Tootle leht

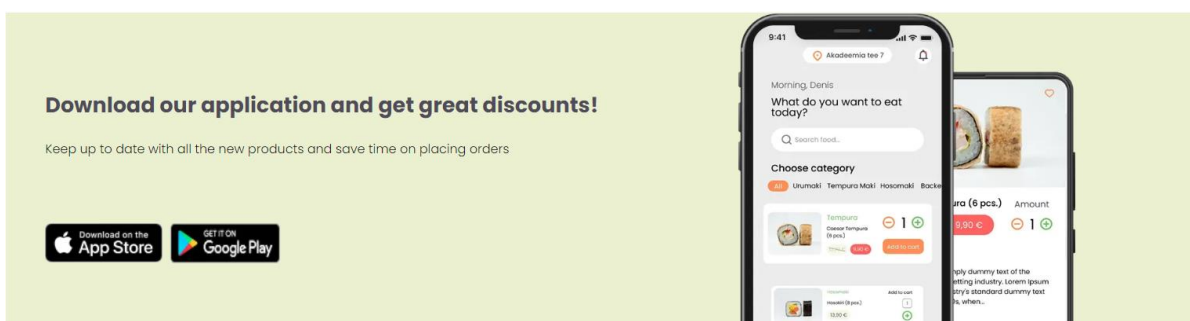


Joonis 8. Tootle leht.

Igal roal on oma leht kirjelduse, hinna ja ostukorvi lisamise võimalusega (Joonis 8). Vasakul on toote pilt, paremal on hind, kirjeldus ja võimalus juure lisada

4.3.5 Mobiilirakenduse allalaadimine

Eripakkumiste ja allahindluste saamiseks on kasutajal võimalus alla laadida mobiilirakendus (Joonis 9).



Joonis 9. Mobiilirakendus.

Selleks peab kasutaja leidma pealehelt banneri ja valima endale sobiva platvormi.

4.3.6 Kasutajaprofiil ja tellimuste ajalugu

Endale konto loonud kasutajatel on võimalus näha nii oma konto infot kui ka täielikku tellimuste ajalugu, kus on märgitud tellitud tooted, kuupäev ja hind (Joonis 10).

My Profile

First name: Firstname
Last name: Lastname
Email: admin@mail.ru
Phone number: +37259653465
Address: Akadem 1-5

Order history

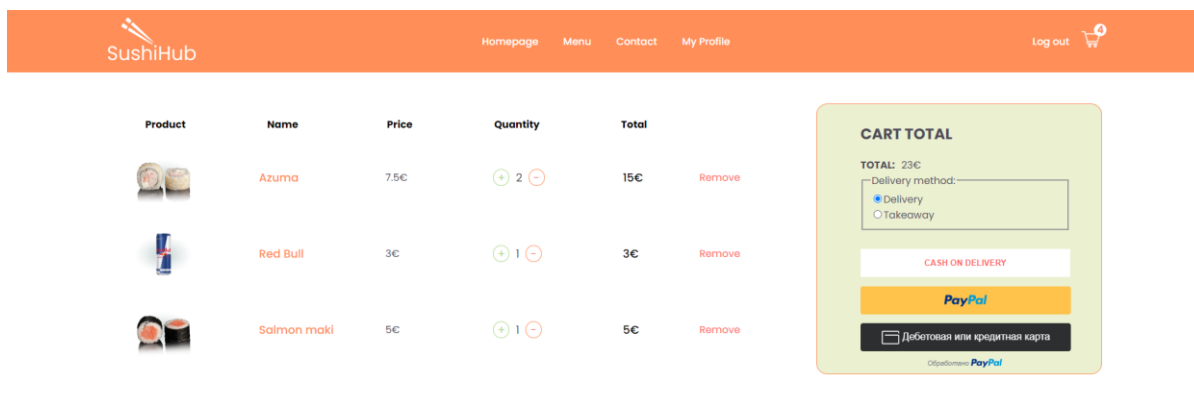
Azuma 2022-04-30 20:03:39.882	Total: 7.5€
Rakoshi 2022-04-30 20:15:32.670	Total: 8.5€
Azuma 2022-05-01 13:42:19.162	Total: 15€
Azuma 2022-05-01 14:52:09.675	Total: 7.5€
Pepsi, Rioto 2022-05-01 14:53:15.660	Total: 14€
Rakoshi	




Joonis 10. Kasutajaprofiil ja tellimuste ajalugu.

Selle jaoks kasutaja peab kontole sisse logima ja tellimuse esitama.

4.3.7 Ostukorv

Iga kord, kui kasutaja lisab toote ostukorvi, saab ta selle kohta teate ekraanile, samuti liigub see toode ostukorvi, kus kasutajal on olemas kogu saadaolev info, nimetus, hind. Klient saab valida ka kohaletoimetamise viisi ja makseviisi (Joonis 11).



Product	Name	Price	Quantity	Total	
	Azuma	7.5€	+ 2 -	15€	Remove
	Red Bull	3€	+ 1 -	3€	Remove
	Salmon maki	5€	+ 1 -	5€	Remove

CART TOTAL

TOTAL: 23€

Delivery method:

Delivery
 Takeaway

CASH ON DELIVERY

PayPal

Дебетовая или кредитная карта

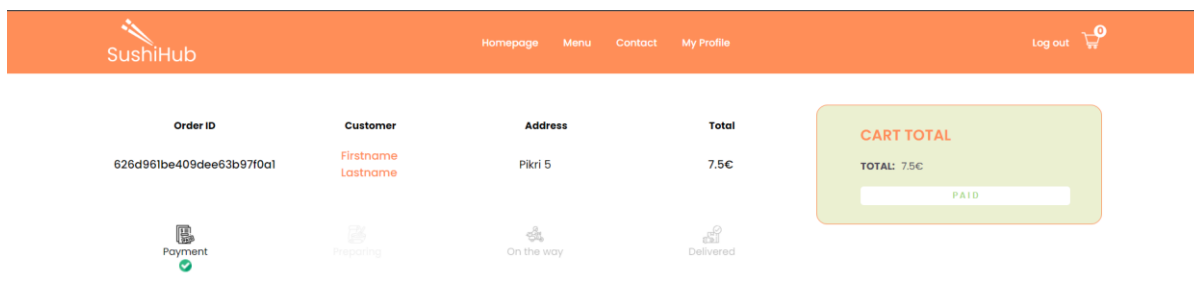
Обработка PayPal

Joonis 11. Ostukorv.

Kliendil on võimalus tellimusele ise järgi tulla või valida kojuvedu. Samuti on makseviisiks PayPal, krediitkaart ja sularaha.

4.3.8 Tellimuse olek

Pärast edukat tellimist suunatakse kasutaja lehele, kus ta näeb oma tellimuse olekut ja teavet (Joonis 12).



Joonis 12. Tellimuse olek.

5 Restoranist toidu tellimise veebirakenduse analüüs

Selles peatükis analüüsitakse arendatud rakendust ja selle testimist. Samuti on välja toodud rakenduse võimalikud edasi arendused tulevikus.

5.1 Rakenduse testimine

Veebirakendust testisid selle restorani potentsiaalsed kliendid. Seda veebi rakendust testinud inimesed andsid oma arvamuse vabas vormis järgmiste kriteeriumide kohta:

- kasutusmugavus
- funktsionaalsus
- töökindlus
- võrdlus teiste sarnaste rakendustega

Testijaid oli kokku 20, neist 15 kliendina, 5 administraatorina. See jaotus oli tehtud selle põhjal, et kasutajal on palju rohkem funktsioone, kui administraatoril. Testimine viidi läbi koosolekul, kus potentsiaalsed kliendid said tööarvuti abil teste läbi viia ja oma muljeid verbaalselt kirjeldada. Nende arvamuse põhjal saadi analüüsid ja järeldused.

Rakenduse testijad jagasid ka oma ideid, kuidas ja mida saab selles veebirakenduses muuta, milliseid lisafunktsioone võiks lisada, kuidas muuta see rakendus veelgi kasutajasõbralikumaks ning kuidas selle konkurentsivõimet tõsta.

Kasutusmugavuse poolest jäid kasutajad rakendusega rahule. Nuppude värvi sõltuvus funktsionaalsusest on märgitud positiivseks. Rakenduste vaated nägid brauserist ja ekraani suurusest hoolimata õiged.

Väga mugav on see, et saate alati tellimuse olekut kontrollida. Lisaks veebi- ja mobiilirakenduse ühine andmebaas võimaldab kasutajatel kasutada sama kontot.

Negatiivsena toodi välja rakenduse keelevaliku puudumise – rakendust on võimalik kasutada ainult inglise keeles.

Funktsionaalsuse poolest jäid kasutajad rahule. Eriti meeldis neile kategooriate filtreerimise menüü, kuna see on tehtud nende arvates ebatavalisel viisil ja on kasutajale alati kergesti ligipääsetav (Joonis 7).

Neile meeldis ka ebavajaliku funktsionaalsuse puudumine, näiteks võimalus valitud toodet edasi lükata ja et administraatoril on võimalus uusi tooteid lisada, tooteid kustutada, muuta ning selle restorani töötajal on võimalus hallata tellimuste olekut. Meeldis ka võimalus vaadata tellimuste ajalugu.

Töökindluse poolest töötas rakendus vigade ja probleemideta. Kasutajad on määranud veebirakenduse kõrge jõudlus. Isegi mitte kõige parema internetiühendusega ei võtnud andmete laadimine palju aega, tänu *NextJS* raamistikule. Kõik tellimused töödeldi probleemideta.

Miinustest nimetasid nad isikuandmete nõrka turvalisust, kuna veebirakenduses ei ole realiseeritud kaheastmelist autentimist.

Seda rakendust testinud kasutajate sõnul teeb see oma tööd. Sellel rakendusel on mõned funktsioonid, mis pole teistes veebirakendustes saadaval. Kuid on ka palju suurema funktsionaalsusega veebirakendusi, mida sellel rakendusel pole. Samuti märgiti ära võimalus vaadata tellimuse olekut, mis pole paljudel teistel veebirakendusel võimalik.

Veebirakenduses konto loomisel on kasutajal konto ka mobiilirakenduse jaoks, kuna rakendused kasutavad ühist andmebaasi.

Kokkuvõttes võib öelda, et üldiselt jäid rakendust testinud kasutajad tööga rahule. Eriti meeldis neile rakenduse kasutusmugavus, töö kiirus, disain ja ebavajaliku funktsionaalsuse puudumine. Kuid ka üks suurimaid miinuseid oli keelevaliku puudumine. Funktsionaalsuse üks põhipunkte oli võimalus vaadata oma tellimuse olekut. Samuti eraldi funktsionaalsus selle restorani administraatorile ja töötajale.

5.2 Rakenduse testimise tulemuste analüüs

Testimise tulemusi analüüsides võime järeldada, et töö käigus loodud rakendus töötab probleemideta ja pakub algselt planeeritud funktsionaalsust. Sellel rakendusel on potentsiaali, kuna enamik potentsiaalseid kasutajaid jäid selle rakendusega rahule. Samuti ei võimalda kõik teised sarnased rakendused kontot luua ja mobiilirakendust telefoni laadida.

Veebirakenduses on realiseeritud toodete filtreerimine kategooria järgi, võimalus luua konto, vaadata tellimuste ajalugu, valida makse- ja tarneviisi. Kõik tooted, nende kirjeldus ja hinnad on salvestatud andmebaasi.

Administraatoril on võimalus uusi tooteid lisada, tooteid kustutada, muuta. Selle restorani töötajal on võimalus hallata tellimuste olekut.

Autori hinnangul on rakendus saavutanud oma eesmärgi ja soovi korral saab seda kasutada. Kuna alternatiivseid rakendusi on palju, tuleks seda rakendust alati uuendada, lisada uusi mugavaid funktsioone, välja mõelda midagi uut, et rakendus oleks konkurentsivõimeline ja oleks võimalus pakkuda seejuures unikaalseid teenuseid.

5.3 Rakenduse täiendused

Rakenduses on realiseeritud esialgselt plaanitud funktsionaalsus, kuid kasutajad, kes on seda rakendust testinud, andsid mõned ideed, mida võiks tulevikus lisada:

- võimalus kasutada rakenduses teisi keeli. Nende arvates on see väga oluline funktsioon, mis aitab sellel rakendusel areneda ja uusi kasutajaid hankida. Kuna kõik inimesed ei oska inglise keelt piisaval tasemel, et iseseisvalt tellimuse esitada. Keelte arvu tuleks laiendada, lisades eesti ja vene keele
- lisada võimalus kasutada Google Maps täpsema aadressi saamiseks. See aitab kulleril õige maja ja sissepääsu täpsemalt üles leida
- realiseerida võimalus sisse logida *Face ID* kasutamisega. See funktsioon aitab kiirendada ja lihtsustada autentimisprotsessi ning lisab turvalisust
- lisada võimalus registreeruda ja sisse logida Apple'i, Facebooki või Google'i konto abil
- realiseerida võimalus probleemi või küsimuse korral võrgutoega ühendust võtta. Tihti on probleeme maksimisega või tekib lisaküsimusi, seega oleks see funktsioon kasutajatele sama kasulik ja mugav
- võimalus vaadata ostukorvi maksumust ilma seda vaatamata, näiteks nii, et hind on kirjutatud korviga ikooni lähedale. Sageli juhtub, et kasutajad unustavad, kui palju nad kaupa kogusid

Analüüsidest ideid selle rakenduse täiustamiseks, võime järeldada, et enamik täiendused on suunatud mugavusele ja funktsionaalsusele. Võib julgelt öelda, et need täiendused suurendavad selle veebirakenduse juurutamise potentsiaali, muudavad veebirakenduse veelgi kasutajasõbralikumaks ja muudavad veebirakenduse konkurentsivõimelisemaks.

6 Kokkuvõte

Selle töö eesmärgiks oli luua restoranist toidu tellimiseks veebirakendus, mida saaksid kasutada selle restorani kliendid, ostes toitu kodust lahkumata ning maksta ja tellimust vastu võtta endale sobival viisil. Samuti töös oli kirjeldatud ka projekti esialgseid nõudeid, mida tuli järgida, kirjeldatud ja analüüsitud sarnased alternatiivsed veebirakendused restoranist toidu tellimiseks ning samuti märgiti ära nende puudused.

Käesolevas töös oli kirjeldatud tehnoloogiaid ja arendusmeetodeid, mille abil see rakendus töötatud välja ja mille abil sai täidetud selle töö algne eesmärk, kirjeldatud valminud rakenduse funktsionaalsust, rakendust oli testitud selle restorani potentsiaalsete klientidega ning nende kommentaarid ja hinnangud olid vastu võetud ja analüüsitud.

Töö käigus analüüsi testitulemusid ja tehti järeldused ning välja toodud soovitused funktsionaalsete täienduste jaoks ja ideed, kuidas muuta veebirakendus veelgi paremaks ja konkurentsivõimelisemaks.

Selle veebirakenduse tagarakendus oli loodud *JavaScript* programmeerimiskeeles ja *NestJS* raamistiku abil. Andmebaasina olid kasutatud *MongoDB*, mis sisaldab kogu infot kasutajate, toodete, tellimuste kohta. Eesrakendus oli realiseeritud kasutades *JavaScript* programmeerimiskeele ja *NextJS* raamistiku.

Valmis rakendus võimaldab kasutajatel saada selle restorani klientideks, registreerides konto, valida soovitud toidu kasutajasõbralikul viisil, valida kliendile sobiva makse- ja tarneviisi, vaadata tellimuse olekut pärast selle esitamist ja ka võimaldab vaadata nende kontot ja tellimuste ajalugu „Minu profiilis“. Selle restorani administratsioonil ja personalil on võimalus näha kõigi kasutajate tellimusi, muuta nende staatust, samuti lisada uusi roogasid või muuta olemasolevaid.

Selle restorani potentsiaalsete klientide testimise tulemuste põhjal võib öelda, et see veebirakendus on valmis oma esialgset eesmärki täitma, kuid aja jooksul vajab see täiustamist, lisades punktis 5.3 mainitud kasutajate ideid ja soove, et teha see rakendus

veelgi funktsionaalsemaks ja kasutajasõbralikumaks.

Kasutatud kirjandus

- [1] „NestJS and its advantages for backend developers,“ [Võrgumaterjal]. Available: <https://blog.devgenius.io/nestjs-and-its-advantages-for-backend-developers-2fdfa785d0b3>. [Kasutatud: 23 aprill 2022].
- [2] Academy Yandex, „Что такое React и как его освоить?,“ (2021, Mar. 29). [Võrgumaterjal]. Available: <https://academy.yandex.ru/posts/что-такое-react-i-kak-ego-osvoit>. [Kasutatud: 23 aprill 2022].
- [3] ITGLOBAL, “MongoDB,“ [Võrgumaterjal]. Available: <https://itglobal.com/ru-ru/company/glossary/mongodb/>. [Kasutatud: 23 aprill 2022].
- [4] Mobilab, „Плюсы и минусы нативных и web приложений,“ [Võrgumaterjal]. Available: <https://inlnk.ru/68wPkQ> [Kasutatud: 23 aprill 2022].
- [5] MongoDB, „Comparing MongoDB vs PostgreSQL,“ [Võrgumaterjal]. Available: <https://www.mongodb.com/compare/mongodb-postgresql>. [Kasutatud: 23 aprill 2022].
- [6] NestJS „Modules,“ [Võrgumaterjal]. Available: <https://docs.nestjs.com/modules>. [Kasutatud: 23 aprill 2022].
- [7] SkillFactory, „NestJS,“ (2022, Feb. 18). [Võrgumaterjal]. Available: <https://inlnk.ru/Rj6Mav>. [Kasutatud: 23 aprill 2022].
- [8] SkillFactory, „Node.js,“ (2021, Nov. 19). [Võrgumaterjal]. Available: <https://blog.skillfactory.ru/glossary/node-js/>. [Kasutatud: 23 aprill 2022].
- [9] Вадим Пашев, „Что такое Next.js и для чего он нужен?,“ (2021, Aug. 12). [Võrgumaterjal]. Available: <https://pxstudio.pw/blog/что-такое-next-js-i-dlya-chego-on-nuzhen>. [Kasutatud: 23 aprill 2022].
- [10] TheCode, „Single Page Application: как работает сайт-приложение,“ (2020, Nov. 28). [Võrgumaterjal]. Available: <https://thecode.media/spa/>. [Kasutatud: 23 aprill 2022].
- [11] “Web-приложение – преимущества и недостатки,“ (2010, Nov. 14). [Võrgumaterjal]. Available: https://mydiv.net/arts/view-web-prilozhenija_preimuxhestva_i_nedostatki.html [Kasutatud: 23 aprill 2022].

- [12] Max Rehkopf “User stories with examples and a template,” [Võrgumaterjal]. Available: <https://www.atlassian.com/agile/project-management/user-stories> [Kasutatud: 12 mai 2022].
- [13] SkillFactory „Redux,” (2022, Feb. 11) [Võrgumaterjal]. Available: <https://inlnk.ru/0QnPK3> [Kasutatud: 12 mai 2022].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

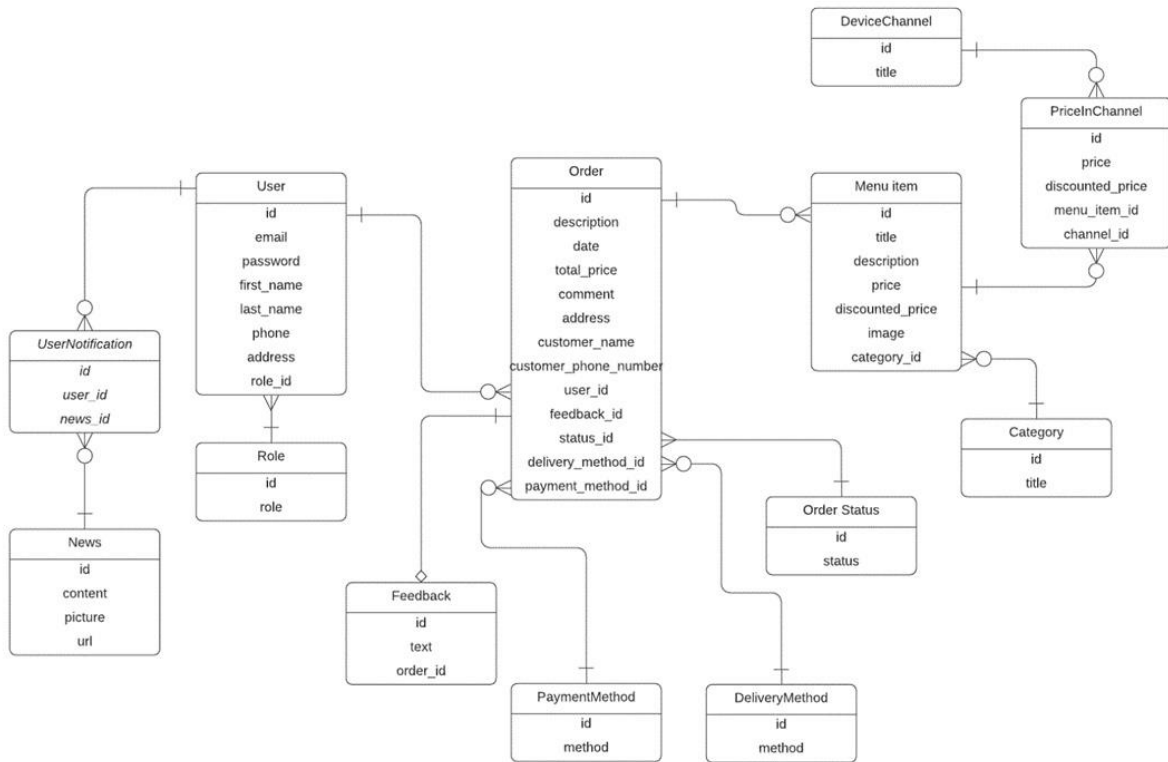
Mina, Martin Aleksander Kostjajev

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Veebirakenduse arendus restoranist toidu tellimiseks", mille juhendaja on German Mumma
 - 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.

13.05.2022

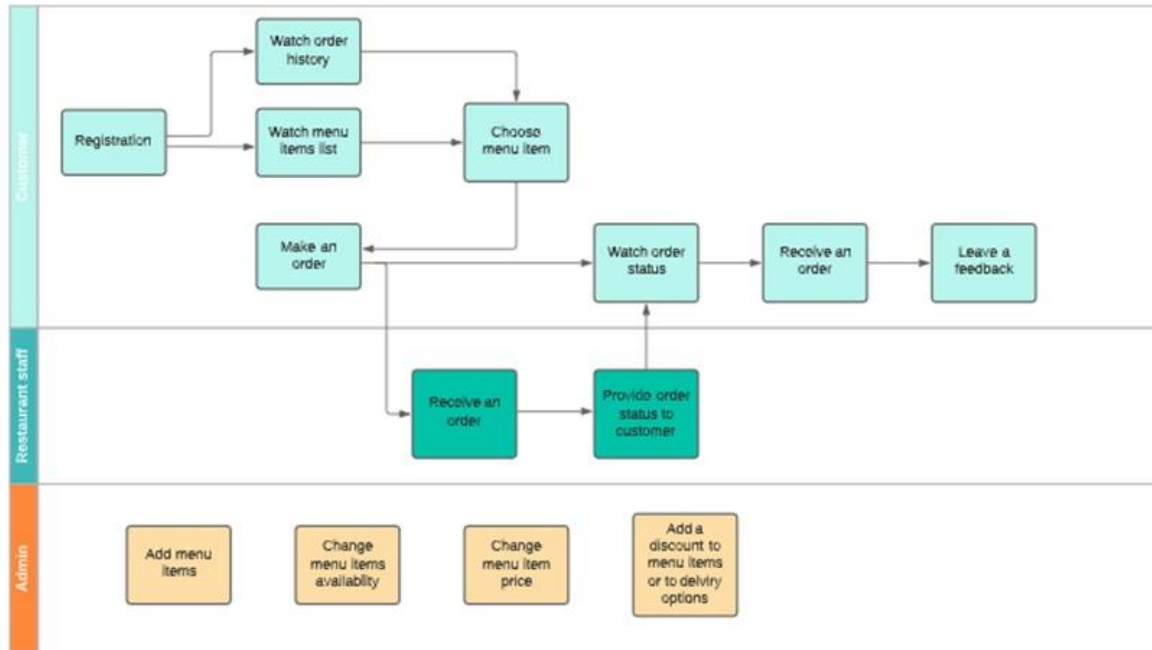
¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või 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 – Andmebaasi skeem



Joonis 13. Andmebaasi skeem.

Lisa 3 – Äriprotsesside voog



Joonis 14. Äriprotsesside voog.

Lisa 4 – Kasutajalood

ID	Priority	As a <type of user>	I want to <perform some task>	so that I can <achieve some goal>
1	Low	kliendina	soovin luua enda kontot	selleks, et näha tellimuste ajaloo
2	High	kliendina	soovin näha restorani menüü	selleks, et valida toitu, mida sooviksin tellida
3	High	kliendina	soovin, et oleks võimalus valida toidu kategooriat	selleks, et oleks lihtsam valida otsitavat toitu
4	Medium/Low	kliendina	soovin tellimuse vormistamise ajal automaatselt valida	selleks, et määrata täpset asukohta
5	Medium	kliendina	soovin jälgida oma tellimuse staatust	selleks, et näha millises etapis on minu tellimus
6	High	kliendina	soovin valida kohaletoitumise viisi	selleks, et oleks võimalus mugava viisi valimiseks
7	High	kliendina	soovin valida maksimisemeetodit	selleks, et oleks võimalus mugavalt maksta
8	High	klienditeenindajana	soovin võta tellimuste vastu	selleks, et kliendil oleks võimalus toitu tellida
9	Medium	klienditeenindajana	soovin muuta tellimuste staatust	selleks, et klient näeks millises etapis on tema tellimus
10	Medium/High	adminina	soovin muuta toitude hinnad	selleks, et reguleerida ökonoomika
11	Medium/High	adminina	soovin toitu menüüsse lisada või sealt eemaldada	selleks, et oleks võimalus uuendada restorani menüü

Joonis 15. Kasutajalood.