

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Marko Gordejev 185751IADB

# **Veebirakendus tekstiilkäsitöö mustriskeemide ja projektide loomiseks**

Bakalaureusetöö

Juhendaja: Toomas Lepikult  
PhD

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: Marko Gordejev

13.05.2022

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärk on luua veebipõhine rakendus tekstiilkäsitöö projektide ning mustriskeemide loomiseks. Antud rakendus vähendaks märkimisväärselt paberikulu ning lahendaks jagamis- ja arhiveerimisprobleeme, mis esinevad paberkujul projekti kavandite ja mustriskeemide loomisel.

Töö teoreetilises osas analüüsitakse olemasolevaid sarnaste eesmärkidega lahendusi ning määratakse kliendiga kooskõlas uue lahenduse nõuded. Pärast seda viiakse läbi analüüs potentsiaalsete tehnoloogiate osas, et leida sobivad tööriistad arendatava rakenduse realiseerimiseks. Seejärel kirjeldatakse loodava rakenduse arendusprotsessiga seotud otsuseid ning lõpuks tuuakse välja uue lahenduse testimise tulemused ja võimalikud edasised arendused.

Bakalaureusetöö praktilise osa tulemusena valmib veebirakendus, mis täidab kõiki eelnevalt seatud nõudeid. Samuti testitakse praktilises osas rakenduse funktsionaalsusi ning vastavust seatud nõuetele nii autori kui ka kliendi poolt.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 38 leheküljel, 8 peatükki, 18 joonist.

## **Abstract**

### **Web Application for Creating Textile Handicraft Pattern Schemes and Projects**

The aim of this Bachelor's thesis is to develop a web application meant for creating textile handicraft projects and pattern schemes. This application would significantly reduce paper consumption and also solve sharing and archiving problems that are present when describing projects and pattern schemes on paper.

In the theoretical part of the final thesis, already existing solutions with similar goals are analyzed and the requirements of the new solution to be implemented are set in accordance with the client. After this, an analysis of potential technologies is carried out to find the best suitable tools for implementing the new web application. This is then followed by describing the decisions related to the development process of the application to be created. Finally, the testing results and possible further developments are pointed out.

As a result of the practical part of this Bachelor's thesis, a new web application is developed that meets all the previously set requirements for it. Furthermore the functionalities of the application and the compliance with the set requirements are tested by both the author and the client.

The thesis is in Estonian and contains 38 pages of text, 8 chapters, 18 figures.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakenduse programmeerimisliides
AWS S3	<i>Amazon Web Services Simple Storage Service</i> , objektide salvestusteenus
BLL	<i>Bussiness Logic Layer</i> , äriloogika kiht
CORS	<i>Cross-Origin Resource Sharing</i> , brauseri mehhanism, mis võimaldab kontrollitud juurdepääsu ressurssidele, mis asuvad väljaspool antud domeeni
CRUD	<i>Create, Read, Update, Delete</i> , neli põhitoimingut, mida rakendatakse andmebaasirakendustes
CSS	<i>Cascading Style Sheets</i> , keel veebilehtede kujundamiseks
DAL	<i>Data Access Layer</i> , andmekiht
DOM	<i>Document Object Model</i> , veebidokumentide programmeerimisliides
DTO	<i>Data Transfer Object</i> , andmeedastusobjekt
GUID	<i>Globally Unique Identifier</i> , Microsofti juurutatud universaalselt kordumatu identifikaator
HTML	<i>HyperText Markup Language</i> , hüpertexti märgistuskeel
HTTP	<i>HyperText Transfer Protocol</i> , protokoll teabe edastamiseks arvutivõrkudes
JSON	<i>JavaScript Object Notation</i> , andmevahetusformaad
JWT	<i>JSON Web Token</i> , JSON formaadis avatud standard, mida kasutatakse turvateabe jagamiseks kahe süsteemi vahel
LINQ	<i>Language Integrated Query</i> , ühtne päringu süntaks C# keeles, mida kasutatakse andmete hankimiseks erinevatest allikatest
NPM	<i>Node Package Manager</i> , paketi haldur JavaScripti programmeerimiskeele jaoks
ORM	<i>Object-Relational Mapping</i> , programmeerimistehnika, mis võimaldab objektipõhise paradigma abil andmebaasist päringuid teha ja andmeid manipuleerida
REST	<i>Representationl State Transfer</i> , tarkvara arhitektuuri stiil, mis seab paika standardid erinevate süsteemide vahel suhtlemiseks
SPA	<i>Single-Page Application</i> , üheleheline rakendus

SQL	<i>Structured Query Language</i> , struktureeritud päringu keel
UOW	<i>Unit of Work</i> , tööühik
URL	<i>Uniform Resource Locator</i> , aadress, mida kasutatakse infoallikate leidmiseks internetis

## Sisukord

1 Sissejuhatus .....	10
2 Probleemi analüüs.....	11
2.1 Probleemi taust .....	11
2.1.1 Silmuskudumine ja mustrite skeemid.....	11
2.1.2 Mustriskeemi loomine .....	12
2.2 Probleemi püstitus .....	13
2.3 Olemasolevate lahenduste analüüs .....	14
2.3.1 Stitch Fiddle.....	14
2.3.2 Adobe Illustrator.....	15
2.3.3 Stitchmastery .....	16
2.4 Uue lahenduse skoop.....	16
3 Rakenduse nõuded.....	18
3.1 Mustriskeem .....	18
3.2 Projekt.....	18
3.3 Kasutaja .....	19
3.3.1 Tavakasutaja .....	19
3.3.2 Administraator .....	19
4 Kasutatavate tehnoloogiate analüüs .....	20
4.1 Andmebaas .....	20
4.1.1 PostgreSQL.....	21
4.1.2 MySQL .....	21
4.1.3 MSSQL.....	22
4.1.4 Autori andmebaasi valiku otsus.....	23
4.2 Tagarakendus.....	23
4.2.1 Java .....	23
4.2.2 C#.....	24
4.2.3 PHP.....	25
4.2.4 Autori tagarakenduse tehnoloogiate valiku otsus.....	26
4.3 Klientrakendus.....	26

4.3.1 JavaScript.....	26
4.3.2 TypeScript .....	26
4.3.3 Raamistik .....	27
4.3.4 React .....	28
4.3.5 Vue.....	29
4.3.6 Angular .....	30
4.3.7 Autori klientrakenduse tehnoloogiate valiku otsus .....	31
4.4 Arenduskeskkond .....	31
5 Rakenduse arendus .....	33
5.1 Andmebaas .....	33
5.2 Tagarakendus.....	35
5.2.1 Struktuur .....	35
5.2.2 Andmekiht .....	36
5.2.3 Äriloogika kiht.....	36
5.2.4 REST API kontrollid .....	37
5.2.5 Turvalisus .....	37
5.3 Klientrakendus.....	38
5.3.1 Struktuur .....	39
5.3.2 Piltide hoiustamine .....	40
5.3.3 Abistavad teegid ja komponendid .....	41
6 Testimine .....	43
6.1 Autoripoolne testimine .....	43
6.2 Kliendipoolne testimine.....	44
6.3 Rakenduse edasised arendused.....	44
7 Tulemused .....	46
8 Kokkuvõte .....	47
Kasutatud kirjandus .....	48
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	53
Lisa 2 – Rakenduse peamised vaated .....	54
Lisa 3 - Valminud rakenduse lähtekood .....	64



## Jooniste loetelu

Joonis 1. Muustriskeemi märkimine edasi-tagasi ridadena. [4] .....	12
Joonis 2. Muustriskeemi märkimine ringselt kudumiseks. [4] .....	13
Joonis 3. Populaarseimad klientrakenduse raamistikud kasutatavuse poolest 2021. aasta seisuga. [38] .....	27
Joonis 4. React, Angular ja Vue raamistikuga seotud StackOverflows postitatud küsimuste hulk protsendina. [40] .....	28
Joonis 5. Olemi-suhte diagramm. ....	34
Joonis 6. JWT genereerimise meetod. ....	38
Joonis 7. AWS S3 konteiner muustriskeemide ning tingmärkide kaustadega. ....	41
Joonis 8. Kodulehekülj. ....	54
Joonis 9. Muustrite koostamise lehekülj. ....	55
Joonis 10. Minu muustrite lehekülj. ....	56
Joonis 11. PDF väljund valmistatud muustriskeemist. ....	57
Joonis 12. Administraatori lehekülj. ....	58
Joonis 13. Kategooria muutmine administraatori leheküljel. ....	58
Joonis 14. Minu projektide lehekülj. ....	59
Joonis 15. Uue projekti loomise lehekülj. ....	60
Joonis 16. Projekti PDF väljund 1. leht. ....	61
Joonis 17. Projekti PDF väljund 2.leht. ....	62
Joonis 18. Minu kasutaja lehekülj. ....	63

# 1 Sissejuhatus

Alustades uue tekstiilkäsitöö projektiga on tavaks luua enne reaalsed käsitööd projekti kavand ning mustriskeem kasutades tingmärke. Enamik tänapäeva käsitöölisi loovad oma mustrite skeemid ja projektid paberi peale, millel on omad negatiivsed küljed: tänu suurele paberi kulule suurendab see käsitöö tootmisprotsessi ökoloogilist jalajälge, see on ajakulukas ja kohati ebamugav tegevus ning samuti on paberi peale loodud kavandeid ja mustriskeeme ebatõhus tuleviku jaoks arhiveerida.

Käesoleva lõputöö eesmärgiks on luua veebipõhine rakendus, kus on võimalik luua ja jagada tekstiilkäsitöö projekte ja mustriskeeme. Antud rakendus võimaldab muuta tekstiilkäsitöö projektide ja mustriskeemide loomise elektrooniliseks protsessiks, mis omakorda lihtsustab käsitööle eelnevat tööd.

Kuna tekstiilkäsitöö koosneb mitmest harust siis on igal harul kasutuses omad kindlad tingmärgid. Lõputöö raames valminud rakendust arendatakse autori poole pöördunud tellijale, kes tegeleb silmusekudumisega ning sellest tulenevalt on selles lõputöös keskendatud silmuskudumisele ning sellega seotud tingmärkidele. Tulevikus on plaanis lisada rakendusesse tugi ka teistele käsitööharudele.

Lõputöö teoreetilise osa esimeses peatükis tutvustatakse kõigepealt silmuskudumise olemust, mustrite skeeme ja nendega seotud tingmärke. Järgmiseks analüüsitakse olemasolevaid lahendusi ning tuuakse välja nende negatiivsed ja positiivsed küljed. Samuti kirjeldatakse uue lahenduse skoopi. Sellele järgneb analüüsi tulemuste põhjal autori ja tellija poolt uue rakenduse nõuete välja kaardistamine. Peale seda analüüsitakse ja võrreldakse rakenduse arendusega seotud tehnoloogiaid, et leida kõige sobilikumad ja efektiivsemad tööriistad rakenduse arenduse jaoks. Seejärel kirjeldatakse rakenduse arendusfaasiga seotud otsuseid ja rakenduse struktuuri ning sellele järgneb testimise tulemuse ja uue lahenduse kasuteguri väljatoomine. Töö praktiline osa hõlmab endas rakenduse arendusfaasi ja testimist.

## **2 Probleemi analüüs**

Antud peatükis antakse ülevaade probleemi taustast ning püstitatakse ülesanne, mida asutakse lõputöös lahendada. Samuti tuuakse selles peatükis välja populaarseimad olemasolevad käsitõomustrite loomise programmid, võrreldakse nende funktsionaalsusi ning nende tulemuste põhjal pannakse kirja uue lahenduse skoop.

### **2.1 Probleemi taust**

Autori poole pöördunud lahenduse tellija on silmuskuduja. Sellest tulenevalt on peamine rõhk selle lõputöö käigus silmuskudumise käsitöötehnikal. Järgnevalt tutvustatakse silmuskudumise olemust.

#### **2.1.1 Silmuskudumine ja mustrite skeemid**

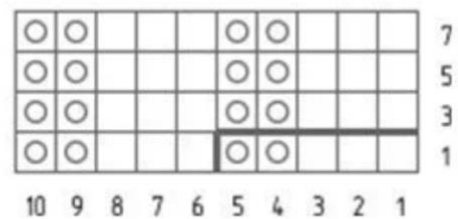
Silmuskudumine on paljude rahvaste kultuurides kaugemale minevikku ulatuvate traditsioonidega käsitöötehnika, mis on aja jooksul teeninud erinevaid eesmärke. Kududa on võimalik nii kudumisvarrastega kui ka kudumismasinaga. Eesti rahvakunstis on silmuskudumine olnud oma populaarsuse tõttu auväärse kohal. [1]

Võib oletada, et mustriskeeme on Eestis laiemalt kasutama hakatud alles 20. sajandil. Enne seda võeti keerukama mustri kudumisel mõni varem kootud ese ette. Samuti kasutati ka pikki prooviribasid, kuhu järjest mustreid kooti. Selliseid mustriäidiste ribasid võeti vahel ka teistelt laenuks. Pärimuse kohaselt tuli laenamise eest üks muster ribale juurde kududa. Üldisemalt võiks mustrite ülesmärkimise jaotada kaheks – märkimine skeemiga ja sõnalise kirjeldusega. Selleks, et kirjeldus liiga pikk ei saaks, siis kasutatakse tavaliselt lühendeid. Alates 1930. aastatest on ajakirjades nagu näiteks "Taluperenaine" ja "Eesti Naine" hakatud mustreid edasi andma skeemidena ning sealtmaalt on ka koolides õpetatud just skeemi järgi kudumist. [2]

20. sajandi keskpaiku hakati kasutama mustritel lihtsama kujuga tingmärke, näiteks parempidine silmus märgiti vertikaalse ja pahempidine horisontaalse kriipsuna. Kuigi hiljem on nii kooliõpikutes kui käsiraamatutes uuesti üle mindud vanadele märkidele,



tuleb tagasiridadel kududa skeemil näidatule tegelikult vastupidine põhisilmus. Selline skeem on ülevaatlikum, sest sellel on muster kohe märgatav (Joonis 2). [2], [3]



Joonis 2. Mustriskeemi märkimine ringselt kudumiseks. [4]

Mõlemate skeemide puhul hakatakse mustrit lugema alt üles ning kudumist alustatakse alati rea numbri poolt. Kui skeemidel on vaja ära märkida eraldatud sektsioon, mida vajadusel korratakse, siis selleks on tavaliselt kasutusel rõhutatud jooned.

## 2.2 Probleemi püstitus

Selleks, et lahendada lõputöös käsitlevat probleemi edukalt, on vajalik sõnastada probleemi olemus korrektselt. Paberi peal mustriskeemide ning projekti kavandite loomisest saab järeldada järgnevad probleemid:

- Paberi kulu - paberi peal mustriskeemide tegemine ning projektide kirjeldamine suurendab käsitöö tootmisprotsessi ökoloogilist jalajälge. Sellest tulenevalt on lõputöö üheks eesmärgiks paberivaba protsessi korraldamine - loobudes paberist väheneks paberikulu iga käsitöölise kohta;

- Jagamine - valmis tehtud mustriskeeme on tihtipeale ebamugav paberi peal teistega jagada. Jagamise protsess oleks palju lihtsam ning mugavam kui kõik käsitööga seotud toimingud saaks sooritada ühes keskkonnas. Loobudes paberist ning muutes jagamise protsessi elektrooniliseks, kiireneks käsitööliste vahel andmete vahetus ning sellest tulenevalt suureneks ka produktiivsus;
- Arhiveerimine - kuna iga mustriskeem võtab ligikaudu 1-3 A3 lehte, siis on nende kujutamine ning arhiveerimine paberi formaadis ebatõhus. Hoides projekti kavandeid ning mustriskeeme digitaalses andmebaasis on seetõttu neid lihtsam tuleviku jaoks hoiustada ja kergem üles leida;

Käesoleva lõputöö peamiseks probleemiks, mida asutakse põhjaliku uurimise ning analüüsi järgselt lahendama on järgmine – tekstiilkäsitöö projektide ja mustriskeemide kujutamine paberile on ebaefektiivne. Sellest tulenevalt on plaanis muuta tekstiilkäsitöö projektide ja mustriskeemide loomine elektrooniliseks protsessiks.

## **2.3 Olemasolevate lahenduste analüüs**

Selles peatükis analüüsitakse olemasolevaid lahendusi, mis omavad funktsionaalsuse poolest sarnasusi lõputöö tulemusena valminud veebirakendusega. Igat lahendust tutvustatakse põhjalikult ning tuuakse välja ka positiivsed ja negatiivsed küljed. Samuti tuuakse välja nende sobivus või mittesobivus antud lõputöö probleemi lahendamiseks.

### **2.3.1 Stitch Fiddle**

Stitch Fiddle on veebipõhine kudumis-, ristpiste- ja heegeldamismustrite kujundamise rakendus, mille eesmärgiks on aidata käsitöölistel kulutada vähem aega mustriskeemide tegemisele, et reaalseks käsitööks jääks rohkem aega. Rakendus loodi kuna sellel ajal olemasolevad lahendused olid liiga kallid, vananenud või raskesti kasutatavad. Rakenduse esimene versioon oli mõeldud eelkõige värvitabelite loomise jaoks, ilma et neid oleks võimalik salvestada ega printida. Pärast mitmete kasutajate positiivset tagasisidet tehti see kõigile kättesaadavaks. [5]

Üldiselt omab Stitch Fiddle enamike funktsionaalsusi, mis on käsitöölisel vaja. Mustrite skeeme on võimalik salvestada ning neid pärast hiljem edasi kujundada. Samuti on

rakendus veebipõhine, ehk rakendust saab kasutada ilma, et peaks alla laadima spetsiaalse programmi, mis hõivaks seadmel mahtu.

Paraku on tehtud mustriskeemi ruudustiku esialgselt määratud mõõtude muutmine äärmiselt tülikaks. Samuti on sellel rakendusel puudu üks väga oluline funktsionaalsus – mustri tingmärke ei ole võimalik kasutaja poolt juurde luua. Rakenduses eksisteerib piiratud kogus tingmärke, mille vahel kasutaja saab valida. Samuti puudub antud veebirakenduses võimalus luua ning kirjeldada käsitöö projekti kavandeid, mistõttu on üksteisest eraldatud mustriskeem ning projekti kavand.

Kuigi rakenduse esialgne versioon on tasuta, eksisteerib sellest ka tasuline versioon. Lisa funktsionaalsuste nagu näiteks mustriskeemide salvestamisemahu suurendamine, teatud kvaliteediga PDF-i formaadis skeemi alla laadimise eest peab maksma. Rakenduse tasulise versiooni hankimiseks ning kasutamiseks peab maksma iga kuu 1.75 €. [5]

### **2.3.2 Adobe Illustrator**

Adobe Illustrator on vektorgraafika redaktor ja kujundusprogramm, millega on võimalik luua jooniseid, illustratsioone ja kunstiteoseid Windows-i või MacOS-i arvutis. Rakendus ilmus algset 1987. aastal ja seda värskendatakse korrapäraste ajavahemike järel. Illustratorit kasutavad peamiselt graafilised disainerid, veebidisainerid, visuaalkunstnikud ja professionaalsed kunstnikud. See sisaldab endas mitmeid erinevaid keerukaid joonistustööriistu, mis võivad vähendada illustratsioonide loomiseks kuluvat aega. [6]

Antud rakendusel on suur tingmärkide kogum ning neid on võimalik hankida juurde erinevate komplektide kaupa. Samuti on võimalus laiendada tingmärkide kogumit laadides alla välistest allikatest sümboliteid ja erinevaid valmis tehtud mustreid. [6]

Adobe Illustratorist on olemas tasuta versioon, mis kestab ühe nädala pärast programmi kasutama hakkamist. Pärast seda tuleb rakenduse säilitamiseks sooritada igakuine makse 19.99\$ või maksta 17.99\$ iga kuu kui võtta aastane pakett. [6]

Kahjuks on antud programmi õppimiskõver suhteliselt järsk ehk seda on tavalisel käsitöölisel, kes ei ole arvutiprogrammidega pidanud varem palju kokku puutama, raske selgeks saada. Kuna Adobe Illustrators on võimalus teha väga palju erinevaid toiminguid ning mustriskeemide tegemine on sellest väike osa, siis raskendab info liigsus

mustriskeemide loomisprotsessi. Kuna see rakendus on mõeldud ainult illustratsioonide valmistamiseks siis puudub sellel võimalus luua samas keskkonnas käsitöö projektide kavandeid ning neid salvestada.

### **2.3.3 Stitchmastery**

Stitchmastery programm võimaldab kujundada keerukaid ja professionaalse välimusega mustreid. Loodud mustriskeeme on võimalik eksportida SVG ning PDF ja samuti ka JPG ja PNG formaadis. Rakenduses on võimalik luua ka tingmärke ise juurde. See tuleb käsitöölisele suureks kasuks, sest alati ei pruugi sümbolite valikust leida sobivat. Paraku on aga uute tingmärkide tegemine seotud juba olemasolevate sümbolite taaskasutamisega ehk puudub otsene võimalus vaba käega uusi tingmärke luua. [7]

Stitchmastery kasutamine ei pruugi käsitöölise jaoks olla nii intuiitiivne võrreldes näiteks varasemalt mainitud Stitch Fiddle rakendusega. Selle tarkvara kasutama õppimine võib võtta aega, kuid õnneks leidub veebilehel erinevate funktsionaalsuste kohta detailseid õppematerjale ning võimalus on ka küsida lehe haldajate käest täpsustavaid küsimusi. Samuti on võimalik registreerida igakuisele uudistekirjale, kus leidub informatsiooni rakenduse edasiarenduste kohta ning nõuandeid ja nippe, kuidas programmi täpsemalt kasutada.

Rakendust saab kasutada tasuta, kuid sellel on omad piirangud. Tasuta versiooniga ei ole näiteks võimalik genereerida kirjalike juhiseid, eksportida mustriskeemi pildi- või tekstifailina ega salvestada enda loodud mustreid, mis on väga oluline funktsionaalsus, kui on vajadus mustriskeeme tulevikus muuta. Programmi ostmiseks ja kõikide funktsionaalsuste kasutamiseks peab maksma ühekordse tasu, milleks on ligikaudu 70€. [7]

## **2.4 Uue lahenduse skoop**

Analüüsi järgselt selgus, et suurimaks puuduseks olemasolevate rakenduste ülesehituses on mustriskeemide ja projektide kavandite eraldavus. Analüüsitud rakendused võimaldavad luua mustreid, kuid projekte pole nendes võimalik luua. Käsitöölisele tuleks suureks kasuks, kui kõik projektiga seotud informatsioon oleks ühes kohas, et vältida lisakeerukust teabe otsimise kujul. Samuti puudub enamikes olemasolevates lahendustes võimalus kasutajal luua kohandatud tingmärke juurde.



Seoses eelnimetatud puudujääkide parandamisega, luuakse selle lõputöö käigus veebipõhine rakendus. Selle esialgne nimi on HandKrafts. Uue veebirakenduse puhul on eraldatud tavakasutajate ja registreeritud kasutajate funktsionaalsused. Registreerimata kasutajatel on võimalus mustriskeeme luua ning neid soovi korral eksportida PDF formaadis. Registreeritud kasutajatel on võimalus loodud mustreid salvestada ja neid hiljem muuta. Samuti on registreeritud kasutajatel võimalus luua juurde uusi tingmärke, projekte ja vaadata ka teiste kasutajate tehtud projekte. Projekti loomise protsessi käigus peab kasutaja määrama selle raskustaseme ning kategooria.

Rakendusega on seotud ka administraatori rollis kasutajad, kes haldavad rakenduses toimuvat. Administraatori rolliga kasutajatel on lisaks eelnevalt nimetatud funktsionaalsustele võimalus lisada, muuta ning kustutada projektidega seotud raskustasemeid ja kategooriaid. Samuti saab muuta ning kustutada ebasobivaid projekte ja mustreid. Veel saavad administraatorid lisada juurde kõikidele kasutajatele kättesaadavaid tingmärke.

Esialgu piirduakse loodavas lahenduses silmuskudumise tingmärkidega kuna see on autori poole pöördunud tellija käsitöötehnika suund. Tulevikus on plaanis lisada rakendusesse ka teiste tekstiilkäsitöölade sümboleid nagu näiteks heegeldamise- ja tikkimise- tingmärke. Rakendusega seotud andmebaasi on vaikumisi lisatud populaarseimad ning enam levinud tingmärgid ja nende selgitused. Registreeritud kasutajatel on soovi korral võimalus valmistada personaalseid tingmärke, mis jäävad tema kasutajaga seotuks.

Kliendiga arutades mustriskeemi ruudustiku ülesehituse osas otsustati kasutada valmivas rakenduses niinimetatud hübriid-lahendust, kus on võimalik koostada mustriskeeme nii edasi- ja tagasireal kui ka ringselt. See paindlik lahendus võimaldab kirjeldada iga realist mustriskeemi, kus iga uue rea alguse määrab ära tingmärk „ületõstetud ääresilmus“. Samuti otsustati märkida mustriskeemi ruudustikul ridade arvud vasakule ja paremale ja silmuste arvud üles ja alla. See lubab ka suurte mustriskeemide puhul paremini näha ridade ja silmuste arvu.

## **3 Rakenduse nõuded**

Selles peatükis tuuakse välja kõik vajalikud nõuded, mis peaksid rakenduses esindatud olema. Nõuded sai koostatud ning kooskõlastatud nelja kogenenud silmuskudujaga, kes tegelevad mustrite ja projektide koostamisega pidevalt. Ülesmääratud nõuete järgimist jälgib autor rakenduse arendusfaasis. Arendusfaasi saab lugeda edukalt lõppenuks, kui kõik nõuded on rakenduses toetatud.

### **3.1 Muustriskeem**

Mustriskeemiga seotud nõuded:

- Muustriskeemi loomine, kustutamine, muutmine;
- Muustriskeemi ruudustiku dimensioonide muutmine;
- Muustriskeemi eksportimine PDF failina;
- Muustriskeemi salvestamine;
- Uue personaalse tingmärgi loomine, kustutamine, muutmine.

### **3.2 Projekt**

Projektiga seotud nõuded:

- Projekti loomine, kustutamine, muutmine;
- Projekti privaatseks või avalikuks märkimine;
- Projekti valmisoleku märkimine;
- Avalike projektide vaatamine;
- Projekti eksportimine PDF failina;

- Projekti salvestamine;
- Muustriskeemi sidumine projektiga;
- Projekti nime, lühikirjelduse, kategooria, raskusastme, töövahendite, materjalide, tööproovi ja töökäigu kirjeldamine.

### **3.3 Kasutaja**

Järgnevalt on loetletud süsteemi kasutajatega seotud nõuded.

#### **3.3.1 Tavakasutaja**

Tavakasutajaga seotud nõuded:

- Kasutajaks registreerimine;
- Kasutajana sisse logimine;
- Kasutaja eesnime, perekonnanime, parooli muutmine.

#### **3.3.2 Administraator**

Administraatoriga seotud nõuded:

- Avalike tingmärkide lisamine, kustutamine, muutmine;
- Kategooriate lisamine, kustutamine, muutmine;
- Raskusastmete lisamine, kustutamine, muutmine;
- Projektide kustutamine;
- Muustriskeemide kustutamine.

## 4 Kasutatavate tehnoloogiate analüüs

Enne rakenduse arenduse alustamist on vajalik valida selleks sobivad tehnoloogiad ja tööriistad. Valiku langetamiseks viis autor läbi põhjaliku analüüsi erinevatest tehnoloogiatest, et leida kõige efektiivsemad ning sobivamad vahendid rakenduse arendamiseks. Analüüsi käigus võrreldi omavahel erinevaid andmebaasihaldussüsteeme ning taga- ja klientrakenduse programmeerimiskeeli ja raamistikke.

### 4.1 Andmebaas

Veebirakendusega seotud andmete säilitamiseks kasutas autor arvutipõhist andmebaasi. Andmebaasi mudel näitab andmebaasi loogilist struktuuri, sealhulgas seoseid ja piiranguid, mis määravad, kuidas andmeid saab salvestada. Kõige levinumateks andmemudeliteks peetakse relatsioonilist andmemudelit, hierarhilist andmemudelit ja objekt-relatsioonilist andmemudelit. [8]

Relatsioonilises andmebaasis hoitakse andmeid tabelites, mis on omavahel üksteisega seotud. Tabelites salvestatakse andmed ridadesse ja veergudesse. Tabelite vahelised suhted on seotud unikaalse primaarvõtmega. Primaarvõti tagab tabeli kirjade unikaalsuse. Primaarvõtme abil on võimalik ka tuvastada teistes tabelites olevaid kirjeid. [8], [9]

Hierarhilised andmebaasid kasutavad andmete salvestamiseks vanem-laps mudelit. Selles mudelis salvestatakse andmed kirjade kujul ja need omakorda korraldatakse puutaoliseks struktuuriks. Paraku on selle andmebaasi tüübi puhul tegemist üks-mitmele vormiga ehk alamkirjetel ei saa olla rohkem kui üks ülemkirje [10].

Objekt-relatsiooni andmebaasideks nimetatakse andmebaasihaldussüsteeme, mis ühendavad relatsiooni andmemudeli lihtsuse ja objektorienteeritud andmemudeliga seotud funktsionaalsused. Objekte, klasse ja nende pärimist toetatakse otse andmebaasiskeemides ja päringukeeles. Lisaks on toetatud andmudeli laiendamine erinevate andmetüüpide ja meetoditega [11].

Populaarseimate andmemudelite analüüsi järgselt saab järeldada, et selle lõputööga seotud rakenduse arenduseks sobivad nii objekt-relatsiooniline kui ka relatsiooniline andmemudelid. Nendega on autor kõige rohkem kokku puutunud ning need sisaldavad kõiki vajaminevaid funktsionaalsusi, mis on rakenduse arenduseks vajalikud.

Sobiva andmebaasihaldussüsteemi valimisel keskenduti peamiselt populaarsusele. Kõrgema populaarsuse korral on suurem võimalus, et probleemi tekkimisel on selle kohta rohkem infot. Samuti võeti arvesse autori kogemus vastava süsteemiga. Järgnevalt on analüüsitud autorile tuttavaid ning kõige populaarsemaid objekt-relatsioonilisi ning relatsioonilisi andmebaasihaldussüsteeme.

#### **4.1.1 PostgreSQL**

PostgreSQL on avatud lähtekoodiga tasuta relatsiooniline andmebaasisüsteem, millel on üle 30 aasta kestnud aktiivset arendustööd. Selle tulemusena on teeninud see tugeva maine töökindluse, funktsioonide vastupidavuse ja jõudluse poolest. PostgreSQL toetab mitmeid populaarseid programmeerimiskeeli nagu näiteks Java, C#, PHP ja Python. [12]

SQL (*Structured Query Language*) vastavusstandard on reeglite kogum, mida andmebaas peab kõigi SQL-i juhiste ja standardite rakendamisel järgima. PostgreSQL-il on kõige enam SQL-iga ühilduv, kuna see vastab 160-le SQL-standardi põhifunktsioonile 179-st. Samuti sisaldab see mitmeid valikulisi funktsioone. [13]

2022 aasta jaanuari seisuga on PostgreSQL ligikaudu 74 000 osavõtjaga StackOverflows läbi viidud küsitluse järgi teisel kohal – seda eelistab kasutada 40.42% vastanutest [14]. PostgreSQL-i populaarsust soodustavad mitmed tegurid. Esiteks on sellel aktiivne avatud lähtekoodiga kogukond, mida ei kontrolli ükski sponsor ega ettevõtte. Teiseks on PostgreSQL-i funktsioonide komplekt pidevas täiendamises ning selle kogukond testib igat uut funktsiooni põhjalikult enne selle integreerimist. Samuti on PostgreSQL-i väga lihtne paigaldada ning üles seada. [15]

PostgreSQL andmebaasisüsteemiga omab autor kõige rohkem kogemusi – seda on kasutatud nii erinevates hobiprojektides kui ka tööl olles mitmes projektis.

#### **4.1.2 MySQL**

MySQL on Oracle poolt arendatud ning toetatud tasuta avatud lähtekoodiga SQL-i andmebaasihaldussüsteem. Oma loomu poolest on MySQL andmebaasi server kiire,

usaldusväärne, skaleeritav ning kergesti kasutatav. Kuigi seda saab kasutada paljudes rakendustes, seostatakse MySQL-i kõige sagedamini veebirakendustega. [16]

MySQL on üles ehitatud klient-server arhitektuurile - see toetab mitmesuguseid taustaprogramme ja ka erinevaid programmeerimisliideseid. Tänu arhitektuurilistele ja keelelistele sarnasustele saab olemasolevaid andmeid otse MySQL-ist enamikesse teistesse relatsiooni andmebaasisüsteemidesse üle viia. [17]

Eelnevalt mainitud StackOverflow küsitluses oli MySQL populaarsuse järjekorras esimene – seda eelistab kasutada 50.18% vastanutest [13]. Selle peamisteks põhjusteks on MySQL-i väike maht, kiirus, kergesti kasutatavus ning skaleeritavus. Kuigi MySQL sisaldab vähem funktsioone kui PostgreSQL, on see pidevas arendusprotsessis, mistõttu luuakse vajalike ning kasulike funktsioone pidevalt juurde. [18]

Kuigi autor on antud andmebaasihaldussüsteemiga varasemalt projektides kokku puutunud, on sellega siiski vähem kogemusi kui PostgreSQL-iga.

#### **4.1.3 MSSQL**

MSSQL ehk Microsoft SQL Server on relatsiooniline andmebaasisüsteem, mille on välja töötanud Microsoft. SQL Server on kasutusel olnud Windowsi platvormil enam kui 20 aastat. Alles 2016. aastal tegi Microsoft selle kättesaadavaks Linux operatsioonisüsteemidel töötavatele süsteemidele. MSSQL on seotud Transact-SQL-iga, mis on Microsofti poolt laiendatud versioon tavalisest SQL-ist. See lisab tavalisele SQL-ile juurde hulga lisafunktsioone. [19], [20]

SQL Serverist eksisteerib hetkeseisuga viis erinevat peamist väljaannet, millest kõik erinevad üksteisest kättesaadavate teenuste või tööriistade poolest. Kaks nendest – SQL Server Developer Edition ning SQL Server Express on tasuta versioonid ning on mõeldud eelkõige väiksemate andmebaaside jaoks või on litsenseeritud ainult arenduse ja testsüsteemide tarbeks. Ülejäänud kolm on tasulised versioonid ning sisaldavad rohkem tööriistu. [20]

StackOverflow andmebaasisüsteemide küsitlusest selgus, et MSSQL on populaarsuse poolest viiendal kohal - seda eelistab kasutada 26.87% arendajatest [14]. Microsoft SQL-i populaarsus võib olla tingitud mitmest asjaolust - Microsofti andmetel on MSSQL-is viimaste aastate jooksul olnud kõige vähem turvaauke võrreldes teiste andmebaasidega.

Kui 2010-2016 aastate vahemikul registreeriti kõigest 19 turvaauku, siis teised samaväärsed andmebaasid olid selle aja jooksul registreerinud 200 või rohkem turvaauku. Samuti kasutab MSSQL andmete taastamiseks ja varundamiseks spetsiaalset andmekaitsehaldurit ehk sellel on palju võimalusi andmete varundamiseks ja taastamiseks logifailide ja vahemällu salvestamise kaudu. [21]

Kogemuse poolest on autor varasemalt kokku puutunud MSSQL-iga ühe projekti käigus. Samuti on autori arvates MSSQL-i andmebaasi ülesseadmine võrreldes teiste samaväärsete andmebaasihaldussüsteemidega, märksa keerulisem ning pikem protsess.

#### **4.1.4 Autori andmebaasi valiku otsus**

Andmebaasihaldussüsteemide analüüsi järgselt otsustas autor valida PostgreSQL andmebaasi kasuks. See sisaldab kõiki olulisi funktsionaalsusi, mis on lõputöös valmiva rakenduse jaoks vajalikud. Samuti kuna tegemist on ühe populaarseima ning avatud lähtekoodi sisaldava andmebaasihaldussüsteemiga, on probleemide esinemisel suur võimalus leida üles abistavad lahendused. Lisaks on autoril eelnimetatud andmebaasidest kõige rohkem kogemusi PostgreSQL-ga.

## **4.2 Tagarakendus**

Tagarakenduse ülesandeks selle lõputöö raames on sooritada erinevaid toiminguid andmetega ning selle tulemusena saadud andmetest panna kokku sisu ning serverida see klientrakendusele. Tagarakenduse programmeerimiskeele ja raamistiku valimisel tugines autor oma olemasolevatele oskustele ning vastava tehnoloogia populaarsusele. Enne otsuse langetamist viis autor läbi põhjaliku uuringu erinevatest programmeerimiskeeltest, millega ta ennast kõige kindlamalt tunneb. Samuti võrreldakse kõige populaarsemaid veebirakenduste ehitamise jaoks mõeldud raamistike, mille eesmärgiks on lihtsustada rakenduse arendusfaasi.

### **4.2.1 Java**

Java on üldotstarbeline klassipõhine objektorienteeritud programmeerimiskeel, mis avaldati esialgselt Sun Microsystemsi poolt 1995. aastal [22]. Javat kasutatakse laialdaselt rakenduste arenduses, sülearvutites, andmekeskustes, mängukonsoolides, mobiiltelefonides ja serveripoolsete tehnoloogiate jaoks nagu näiteks Apache, JBoss või GlassFish [23].

2021 aasta mais viidi StackoverFlows läbi põhjalik küsitlus erinevate arendusega seotud tehnoloogiate kohta, millest võttis osa 80 000 arendajat üle maailma. Ühes küsitluse küsimuses tuli märkida vastajatel programmeerimiskeel, millega neil on viimase aasta jooksul kõige rohkem arendustööd tehtud ning millise keelega nad järgmise aasta jooksul töötada sooviksid. Küsitluse tulemusi analüüsesid, selgus, et Java oli sellel aastal viiendal kohal. Seda eelistas kasutada 35.35% vastajatest [24].

Eesmärgiga arendusprotsessi lihtsamaks muuta, otsustas autor uurida ka erinevaid Java veebiarendusega seotud raamistike. Üheks peamiseks ja kuulsaimaks raamistikuks osutus Spring raamistik, mis lihtsustab koodi kirjutamist sõltuvuse süstümise mustril abil [24]. Selle kasulikkus peitub selles, et arendaja saab anda loodud moodulite kontrolli raamistiku kätte, mistõttu arendaja saab keskenduda rohkem rakenduse loogika peale ning ei pea kirjutama ülearust koodi [25]. Samuti osutus üheks populaarseimaks raamistikuks ka Hibernate. Antud raamistikku kasutatakse Java objektide salvestamiseks relatsioonilisse andmebaasisüsteemi. Selle eesmärk on lihtsustada arendatava rakenduse ja andmebaasi vahelist suhtlemist [26].

Autor on läbinud ülikooli õpingute käigus kolm Java programmeerimiskeelega seotud õppeainet, kus said selgeks mitmed Java ja Spring raamistikuga seotud põhitõed ning parimad tavad. Samuti on autor Java tehnoloogiatega kokku puutunud mitmetes projektides.

#### **4.2.2 C#**

C# on kaasaegne, objektorienteeritud ja tüübikindel Microsofti poolt arendatud programmeerimiskeel. C# võimaldab arendajatel luua mitut tüüpi turvalisi ja töökindlaid rakendusi, mis töötavad .NET-is (tuntud ka *dotnet* nime all). Peamiselt on C# seotud C, C++ ja Java programmeerimiskeeltega. [27]

Vaadates eelmises peatükis mainitud 2021. aasta StackOverflow küsitluse andmeid, selgub, et C# on kasutatavuse poolest kaheksandal kohal. Seda eelistas kasutada 27.86% arendajatest, kes küsitlusest osa võtsid. [24]

C# programmeerimiskeelega on seotud mitmeid raamistikke ja platvorme, mille eesmärgiks on muuta arendusprotsess mugavamaks. Üks kuulsamaid nendest on .NET, mis on tasuta, avatud lähtekoodiga arendajaplattform. See toetab mitmeid erinevaid



programmeerimiskeeli ning sellega on võimalik luua palju erinevaid tüüpe rakendusi [28]. See omab kõrget positsiooni ka arendajate poolt märgitud populaarseimate raamistike nimekirjas. Aastal 2021 eelistas kasutada .NET raamistiku 31.5% vastanutest [24].

C# programmeerimiskeelega tunneb autor ennast kõige kindlamalt sellepärast, et C# ja .NET raamistiku tutvustati ülikooli õpingute käigus väga põhjalikult. Lisaks on autor huvi olemasolu tõttu iseseisvalt uurinud ning katsetanud vastavate tehnoloogiatega mitmetes projektides.

### **4.2.3 PHP**

PHP on laialdaselt kasutatav avatud lähtekoodiga üldotstarbeline skriptimiskeel, mis on mõeldud eelkõige veebirakenduste loomiseks. PHP-d eristab kliendipoolsest programmeerimiskeelest nagu näiteks JavaScriptist, see, et kirjutatud kood käivitatakse serveris. Serveris genereeritakse koodi põhjal HTML (*HyperText Markup Language*) ja seejärel saadetakse see kliendile. Sellest tulenevalt saab klient skripti tulemuse kätte, teadmata, mis selle genereerimiseks vajalik kood oli. [29]

Analüüsidest taaskord StackOverflow küsitluse vastuseid selgub, et PHP on populaarsuse järjekorras 2021. aasta seisuga üheteistkümnend - seda eelistas kasutada 21.98% vastanutest. PHP raamistike kogumikust on arendajate seas kuulsaks saanud Laravel raamistik. Laravel on veebirakenduste raamistik, mis pakub rakenduse loomiseks struktuuri ning võimaldab tänu sõltuvuste süstimisele arendajal keskenduda rakenduse loogika peale ilma, et peaks mõtlema selle peale, kuidas rakenduste vahelisi ühendusi luua. Lisaks sõltuvuse süstimisele pakub Laravel ka võimast andmebaasi abstraktsioonikihti, integratsiooni testimist ning palju muud kasulikku [30]. Laraveli eelistas kasutada 2021. aastal 10.12% vastanutest, kes said küsitluses valida erinevate veebiraamistike vahel [24].

Kuigi autor on töötanud varasemalt PHP tehnoloogiaga ligikaudu aasta aega, tunneb ta siiski end seda kasutades ebakindlalt. PHP-ga on autor välja toodud programmeerimiskeeltest kõige vähem kokku puutunud. Samuti leiab autor, et PHP süntaks on võrreldes teiste programmeerimiskeelte süntaksitega veidi harjumatu.

#### **4.2.4 Autori tagarakenduse tehnoloogiate valiku otsus**

Peale tagarakenduse tehnoloogiate analüüsi, otsustas autor valida lõputöös arendatava rakenduse programmeerimiskeeleks C# ning abistavaks raamistikus .NET raamistiku. Peamine põhjus selleks leidub asjaolus, et autor omab kõige rohkem kogemusi vastava programmeerimiskeele ning raamistikuga. Samuti leidub nende tehnoloogiate populaarsuse tõttu internetis palju abistavaid materjale. Lisaks on Microsoft koostanud mõlema tehnoloogia kohta väga detailsed dokumentatsioonid, mis sisaldavad mitmeid juhendeid ja nippe veebirakenduste arenduse jaoks.

### **4.3 Klientrakendus**

Klientrakenduse eesmärk lõputöös valmiva veebirakenduse raames on määrata ära, kuidas rakenduse kasutajad erinevaid kuvatavaid elemente ekraanil näevad ning kuidas nendega suhelda saab. Seega on tähtis, et rakendusel oleks võimalikult esteetiline välimus ning lihtsasti kasutatav kasutajaliides. Enne klientrakenduse arendust valis autor välja selleks sobivad tööriistad – programmeerimiskeele ning raamistiku. Valiku langetamisele eelnevalt uuriti autorile tuntud programmeerimiskeelte ning raamistike üldtuntuse ning nende poolt pakutavate funktsionaalsuste kohta. Samuti võeti arvesse autori kogemusi vastavate tehnoloogiatega.

#### **4.3.1 JavaScript**

Tänapäeval üheks kõige populaarseimaks klientrakenduse programmeerimiskeeleks arendajate seas peetakse JavaScripti [31], [32]. JavaScript on programmeerimiskeel, mis võimaldab kasutada veebilehel kompleksseid funktsioone ning luua dünaamiliselt värskendavat sisu, juhtida multimeediumit, animeerida pilte ja palju muud [33]. JavaScripti kasutatakse peamiselt veebipõhiste rakenduste arenduse ja veebibrauserite jaoks, kuid tänu mitmetele teekidele ja raamistikele on seda võimalik kasutada ka näiteks tagarakendustes ning mobiilirakendustes [34].

#### **4.3.2 TypeScript**

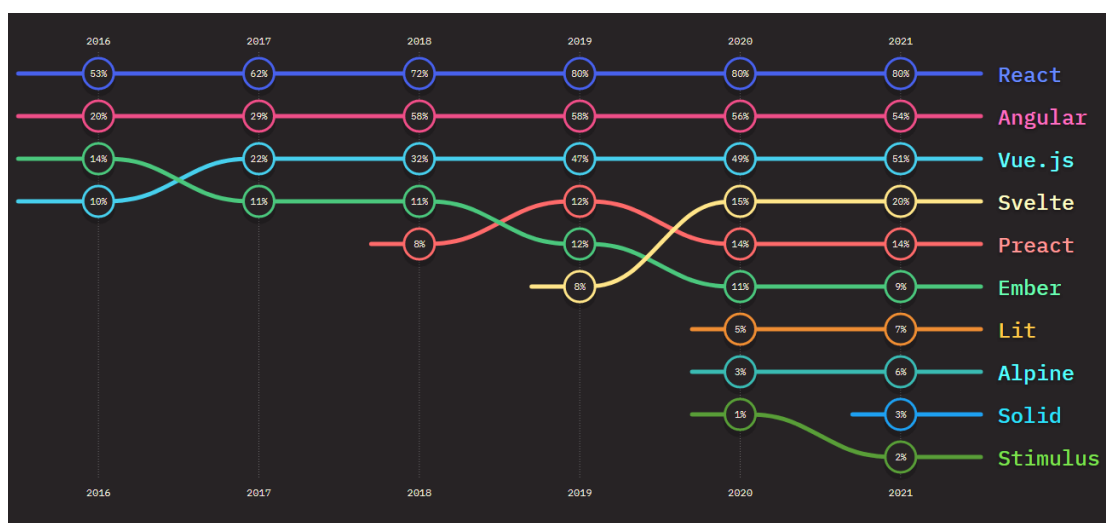
TypeScript on JavaScripti peale ehitatud programmeerimiskeel, mis sisaldab palju kasulike ning abistavaid funktsioone. TypeScriptiga tuleb kaasa võimas tüübisüsteemi, mis muudab arendamisprotsessi lihtsamaks. Lisades JavaScriptile täiendava süntaksi, et toetada tihedamat integreerimist arenduskeskkonnaga, aitab see arendajal leida varakult

koodis vead üles. TypeScripti koodi saab lihtsalt teisendada JavaScripti koodiks, teistpidi aga võimalik teisendust teha ei ole. Samuti võimaldab TypeScript arendajal koodi kirjutades kasutada objektorienteeritud programmeerimistehnikaid nagu näiteks liideste, klasside ja pärimise kasutamist [35].

### 4.3.3 Raamistik

Kuigi puhta JavaScriptiga on võimalik ehitada väga mitmekülgseid klientrakendusi, on sellel siiski omad miinused. Peamine probleem on see, et arendaja peab kirjutama palju korduvat koodi ning rakenduse käitumise muutmise korral peab samuti mitmes kohas muutma suurt hulka koodi. Kui sooritada tuleb palju muudatusi, siis on väga tõenäoline, et varsti on rakenduse kood arendaja jaoks raskesti navigeeritav, kui mitte et täiesti arusaamatu. Erinevad raamistikud pakuvad nendele probleemidele lahendust. Raamistik on eelnevalt kirjutatud koodipakett, mille põhilisteks eesmärkideks on kiirendada arendusprotsessi, kirjutada taaskasutatavat koodi ning mõnel juhul parandada ka rakenduse jõudlust. [36], [37]

Järgnevalt analüüsis autor kolme kõige populaarsemat klientrakenduse raamistiku, et leida kõige sobivam ning efektiivsem raamistik lõputööga seotud veebirakenduse loomiseks. Iga aasta sooritab State of JavaScript keskkond veebiarendajate seas küsitluse, et avastada praeguseid ja tulevaseid trende JavaScripti ökosüsteemis [38]. 2021. aasta küsitluse põhjal osutusid kõige populaarseimateks JavaScripti raamistikeks React, Angular ja Vue.js (Joonis 3). Joonisel on välja toodud raamistike ja teekide kasutatavuse suhted.

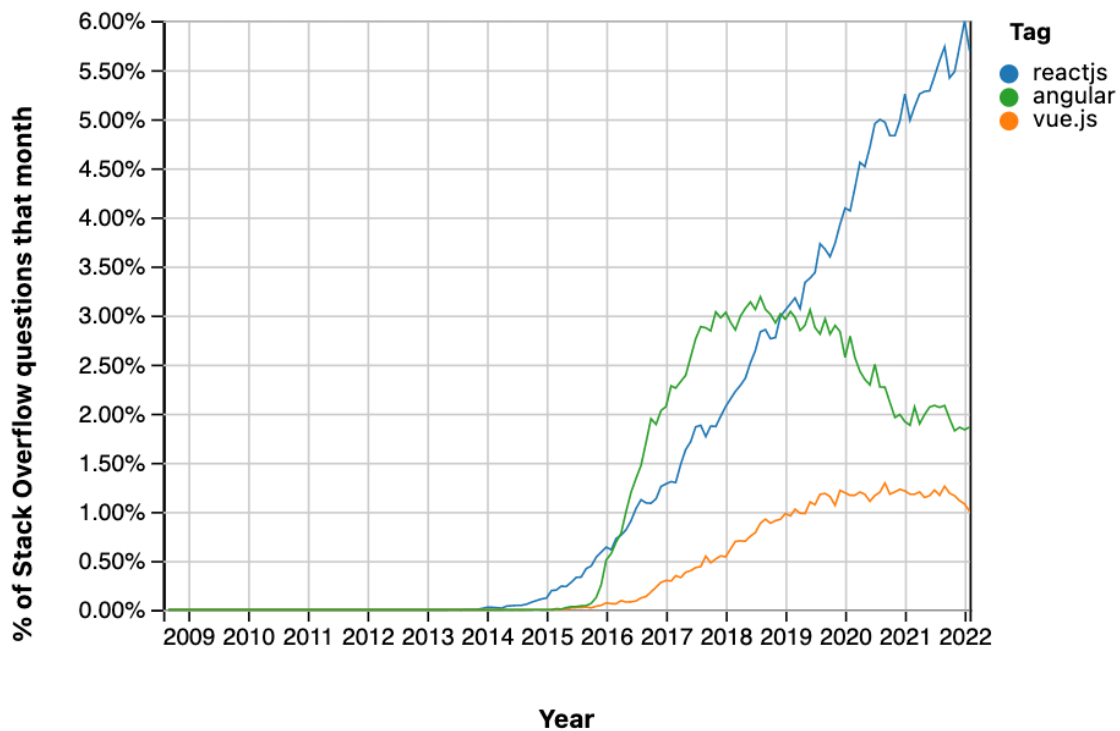


Joonis 3. Populaarseimad klientrakenduse raamistikud kasutatavuse poolest 2021. aasta seisuga. [38]

#### 4.3.4 React

React on avatud lähtekoodiga 2011. aastal Facebooki poolt loodud paindlik JavaScripti teek, millega saab ehitada kasutajaliideseid. See on ülesehitatud komponentide põhjal, mis sisaldavad endas tihtipeale JSX-i ehk JavaScripti süntaksi laiendit. Komponent on mingi eraldatud osa rakendusest, millest hiljem rakendus kokku pannakse. Komponentide koostamine tagab selle, et neid on võimalik rakenduses igal pool taaskasutada ilma, et peaks korduvat koodi kirjutama. React hoiab endas virtuaalset koopiat DOM-ist (*Document Object Model*) ning rakenduse muudatuste korral uuendatakse selles ainult vajalikke elemente, mistõttu teeb see ekraanil uuenduste protsessi kiiremaks. [39]

Reacti populaarsus on alates 2011. aastast tõusnud ja kasvab iga aastaga. 2021. aastal oli see kõige populaarseim klientrakenduse raamistik - kasutatavuse suhe teegil oli 80% (Joonis 3). Samuti on ka StackOverflow küsitluse kohaselt React 2021. aasta kõige rohkem kasutatav raamistik – 67 593 vastanutest, eelistas kasutada Reacti 40.14% [24]. Aja möödudes on hüppeliselt kasvanud ka React raamistikuga seotud StackOverflows esitatud küsimuste arv. 2022 aasta märtsi kuu seisuga on Reacti kohta postitatud joonise kohaselt kõige rohkem postitusi – ligikaudu 5.75% (Joonis 4).



Joonis 4. React, Angular ja Vue raamistikuga seotud StackOverflows postitatud küsimuste hulk protsendina. [40]

React erineb oma loomu poolest teistest JavaScripti raamistikest selle poolest, et tegelikult loetakse seda raamistiku asemel teegiks. See tähendab, et React ei sisalda endas kõiki arenduseks vajalikke tööriistu, vaid tööriistade valiku peab tegema arendaja ise vastavalt tema või kliendi poolt üles seatud nõuetele. Näiteks on vormide koostamiseks kaks peamist populaarset tööriista – Formi ja React Hook Form. Reacti koostajate poolt pole määratud mingeid kindlaid kohustuslikke nõudeid kumba kasutama peab, vaid selle valiku peab langetama arendaja ise. [41]

Autoril on Reactiga ligikaudu poolteist aastat kogemusi. Selle ajaga on autor koostanud vastava raamistikuga kolm hobiprojekti. Samuti on React autori praeguses töökohas igapäevaselt kasutusel.

#### **4.3.5 Vue**

Vue on 2014. aastal välja töötatud JavaScripti raamistik, mis on mõeldud kasutajaliideste loomiseks. See tugineb standardsele HTML-ile, CSS-ile (*Cascading Style Sheets*) ja JavaScriptile ning pakub deklaratiivset ja komponendipõhist programmeerimismudelit, mis aitab tõhusalt arendada kasutajaliideseid. Vue laiendab standardset HTML-i malli süntaksiga, mis võimaldab arendajal deklaratiivselt kirjeldada HTML-i JavaScripti oleku põhjal. [42]

Vue looja, Evan You, kes töötas Googles ning kasutas igapäevaseks tarkvara arenduseks AngularJS raamistikku, tuli idee peale teha raamistik, millel oleksid kõik Angularist tulevad vajalikud funktsioonid, mis talle meeldisid ning oleks samal ajal võimalikult kerge oma mahu poolest. [43]

Vue jääb oma populaarsuse poolest StackOverflows postitatud küsimustega alla nii Reactile kui ka Angularile – 2022. aasta märtsi seisuga oli kuus kõigest 1% küsimusi postitatud (Joonis 4). Sellest saab järeldada, et probleemide tekkimisel arenduse käigus ei saa oodata nii suurt tuge StackOverflow postitustest, kui teiste populaarsemate raamistike puhul. Vue on State of JavaScript lehekülje järgi kasutatavuse poolest kolmandal kohal ning selle kasutatavuse suhe on 51% (Joonis 3). Lisaks on Vue 2021. aasta StackOverflow läbi viidud küsitluse kohaselt viiendal kohal - seda eelistas kasutada 18.97% vastanutest [24].

Kuna töö autor pole antud JavaScripti raamistikuga ülikooli väliselt projektides osalenud, siis ei tunne ta selle raamistiku oskustega ennast kõige enesekindlamalt.

#### 4.3.6 Angular

Angular on TypeScriptiga ehitatud arendajatele mõeldud arendusplatvorm. Selle põhitoetaja on Google. Angulari arendusplatvorm on alates 2016. aastast jaotatud kaheks – AngularJS ning Angular. AngularJS on 2010. aastal välja lastud ning nende autoriteks on Misko Hevery ja Adam Abrons, kes olid Googles töötavad tarkvaraarendajad. 2016. aastal väljastas Google tänu kasvavatele modernsetele nõuetele Angular 2, mis oli täielik ümberehitus originaalsest Angularist. Põhjus, miks otsustati luua täiesti uus eraldiseisev versioon peitub selles, et erinevused vana Angulari ja uuema vahel olid liiga radikaalsed. Angulari on pärast esimest uuendust aastate möödumisel veel uuendatud. Hetkeseisuga on kõige uuem Angulari versioon 9. Selles alampeatükis on analüüsitud 2016. aastal välja ilmunud Angular versiooni, sest see on tänapäeval populaarsem kui vanem versioon. [24], [44]

Angulari platvorm sisaldab endas komponendipõhist raamistiku, millega on võimalik luua skaleeritavaid veebirakendusi. Lisaks sisaldab raamistik teekide kogumit, mis omakorda hõlmab endas paljusid erinevaid funktsioone arenduse lihtsustamiseks nagu näiteks vormide haldamine, navigeerimine ja kliendi-serveri suhtlemine. Samuti sisaldab platvorm juba eelnevalt valmiskirjutatud erinevaid tööriistade komplekte, mis aitavad kaasa koodi kirjutamisel, testimisel ja rakenduste ehitamisel. [45]

Angulari populaarsus on aastatega kahanenud - kui 2018. aasta keskpaigaks oli StackOverflows igakuine Angulariga seotud postituste suhe ligikaudu 3.25%, siis tänapäevaks on see langenud umbes 1.8% peale (Joonis 4). Samuti selgub ka State of JavaScript veebileheküljelt, et aina rohkem arendajaid on aja möödudes Angularist eemale pöördunud ning seetõttu on langenud ka selle raamistiku kasutatavuse suhe - 2021. aasta seisuga oli Angulari kasutatavuse suhe 54% (Joonis 3). 2021. aasta StackOverflows läbi viidud küsitluse kohaselt eelistas Angulari kasutada 22.96% arendajatest [24].

Angulari raamistikuga on autoril lisaks kooli õpingute kõrvalt kaheksa kuud töökogemust. Autor järeldeb oma kasutamise kogemustest Angulariga, et selle raamistiku

õppimisega ning ära harjumisega kulub rohkem aeg võrreldes teiste eelnevalt mainitud raamistikega.

#### **4.3.7 Autori klientrakenduse tehnoloogiate valiku otsus**

Autor otsustas analüüsi järgselt valida klientrakenduse raamistikuks Reacti. Selle valiku tegi autor seoses sellega, et tal on võrreldes Vue ja Angulariga, selle raamistikuga kõige rohkem kogemusi. Samuti osutus React analüüsi järgselt kõige populaarseimaks klientrakenduse raamistikuks, mistõttu ei pea autor muretsema, et selle raamistiku arendajad selle toetamise lähiajal lõpetama peaksid. Raamistiku suure populaarsuse tõttu saab ka leida internetist palju kasulikke ning abistavaid materjale.

Kuna TypeScript on pealisehitus JavaScriptile, mis sisaldab endas mitmeid lisafunktsionaalsusi ning milles on võimalik kirjutada määratud tüüpidega koodi, siis tegi autor klientrakenduse programmeerimiskeele valikul otsuse selle kasuks. Tuginedes sellele, et TypeScript on staatiliselt tüübitud keel, tagab see selle, et kompileerimise käigus kontrollitakse muutujate tüüpe ning seetõttu vähendab see potentsiaalsete loogikavigade ilmumist koodi kirjutamisel. Praeguseks on töö autor harjunud kõiki oma klientrakenduse projekte alustama TypeScripti kasutades.

### **4.4 Arenduskeskkond**

Arenduskeskkond on arendaja jaoks väga oluline tööriist. Arendaja peab tundma ennast selles võimalikult enesekindlalt ning mugavalt. Õige programmi valikul oleks mõistlik lähtuda järgnevatest kriteeriumitest: programmis eksisteerib võimalikult vähe häirivaid tegureid ning tundmatuid osasid, arenduskeskkond pakub arendajale mitmeid abistavaid lisasid ja programmi hind.

Eelnevalt mainitud kriteeriumite põhjal valis autor tagarakenduse arenduskeskkonnaks JetBrainsi poolt arendatud Rider tarkvara. Otsus langetati fakti tõttu, et autoril on vastava programmiga kõige rohkem arendustunde. Samuti omab Rider tarkvara palju mugavusfunktsionaalsusi, mis muudavad koodi kirjutamise võimalikult lihtsaks ning kiireks. Kuigi antud arenduskeskkonna täisversioon on tasuline, on see autorile siiski täismahus kättesaadav tänu ülikooli litsentsile.

Klientrakenduse arenduskeskkonna valikuks osutus Microsofti poolt välja lastud Visual Studio Code tarkvara. Sarnaselt tagarakenduse arenduskeskkonnaga on autoril selle tarkvaraga kõige rohkem klientrakenduse vallas arendustunde möödunud. Visual Studio Code sisaldab endas mitmeid sisseehitatud tööriistu, mis kiirendavad arendusprotsessi. Samuti on võimalik vajaduse korral alla laadida abistavaid laiendusi. Antud tarkvara on 2021. aasta StackOverflows läbi viidud küsitluse põhjal kõige populaarseim arenduskeskkond arendajate seas [24].



## 5 Rakenduse arendus

Käesolevas peatükis kirjeldatakse arendatava veebirakenduse arendusprotsessiga seotud otsuseid. Kõigepealt tutvustatakse rakenduse andmebaasi struktuuri ning seejärel nii tagarakenduse kui ka klientrakenduse ülesehitust.

### 5.1 Andmebaas

Relatsioonilises andmebaasis olevates tabelites peab igal väljal olema määratud tema andmetüüp. Enne andmebaasi struktuuri paika panemist tuleb autoril otsustada, mis on primaarvõtme välja andmetüüp. Primaarvõti on unikaalne väärtus relatsioonilises andmebaasis, mille abil on võimalik eristada tabelis olevaid kirjeid [46].

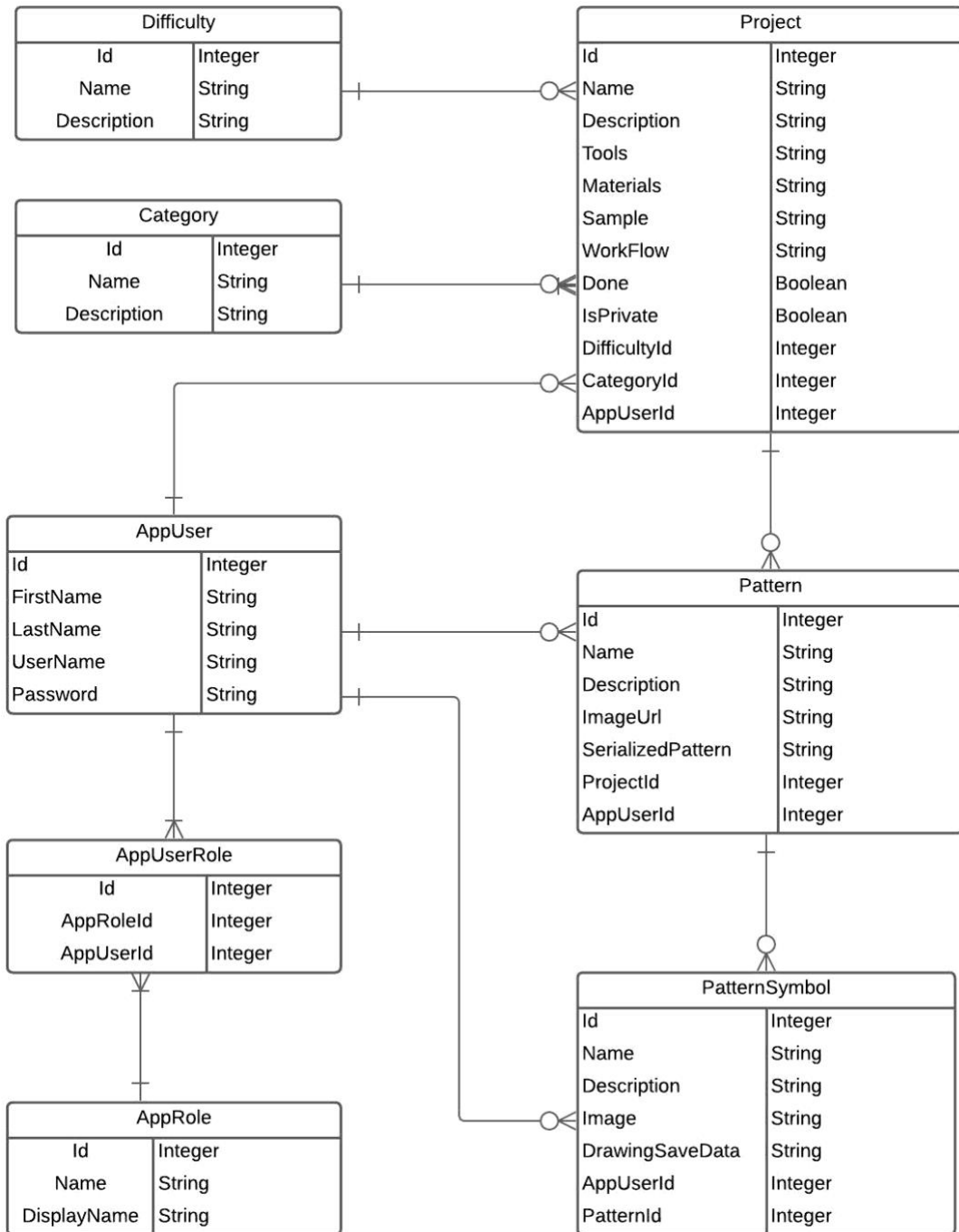
Primaarvõti võib koosneda ühest või mitmest väljast. Kui primaarvõti sisaldab mitut välja, koosneb see tavaliselt olemasolevate väljade kombinatsioonist, mis koos annavad tulemuseks unikaalsed kirjed tabelis [47]. Kuna arendatava rakenduse andmebaasis sisalduvad tabelid ei ole erilise keerukusega, siis ei pidanud autor vajalikuks valida mitmest väljast koosnevat primaarvõtit. Tänapäeval on peamiselt kaks üldtunnustatud valikut primaarvõtme andmetüüpide seas – *integer* ehk täisarv või GUID (*Globally Unique Identifier*) [48].

GUID on 128-bitine jada, mis on kanooniliselt esitatud sõne kujul. Neid on lihtne juurde genereerida, kuid pole oma pikkuse poolest ei kliendile ega arendajale kasutajasõbralikud. GUID on küll unikaalne üle kõigi serverite, kuid paraku on neid arenduse ja testimise ajal suure mahu tõttu ebamugav kasutada. Samuti on nende pikkuse tõttu neid ka rakenduse kasutajal ebamugav URL-ides (*Uniform Resource Locator*) kasutada. [48], [49]

*Integer* andmetüübi salvestusruumi suuruseks on 32 bitti. See tähendab, et toimingud nagu näiteks indekseerimine ja andmebaasi vastu tehtavad pärginud on üldiselt kiiremad kui GUID puhul. Samuti on neid arenduseprotsessis lihtne kasutada, kuna arendaja ei pea töötama pikkade väärtustega, mida on ebamugav üksteisest eristada. [48]

Lõputöös valmiva rakenduse väikese mahu tõttu ning arendusprotsessi lihtsustamiseks otsustas autor primaarvõtme andmetüübiks valida *integer* andmetüübi.

Enne arendusprotsessi algust koostas autor andmebaasis olevate tabelite kohta olemissuhte diagrammi (Joonis 5). Antud diagramm on valmistatud Lucidchart platvormi abil.



Joonis 5. Olemissuhte diagramm.

## 5.2 Tagarakendus

Tagarakenduse arendusel kasutati .NET versiooni 6.0 ning JetBrains Rider versiooni 2021.3.1. Andmebaasis olevate andmete töötlemiseks kasutati Entity Framework raamistiku.

Entity Framework on Microsofti poolt loodud .NET-i ORM (*Object-Relational Mapping*) raamistik. See võimaldab töötada relatsiooniandembaasi andmetega kui objektidega, kõrvaldades seejuures suure osa andmebaasiga suhtlemise koodist, mida peab tavaliselt arendaja ise kirjutama. Raamistiku kohaselt on C# objektid programmis, andmebaasis on objektidele vastavad tabelid ning nende vahel toimub kaardistamine. Andmebaasi päringute tegemiseks kasutati LINQ (*Language Integrated Query*) keelt, mis on struktureeritud päringu süntaks. LINQ keel võimaldab arendajal kirjutada C# programmeerimiskeele abil päringuid Entity Frameworki mudeli alusel. LINQ päringud teisendatakse lõpuks tavalisteks SQL lauseteks. [50], [51]

Rakenduse struktuuri puhul toetuti eelnevalt valmistatud projektidele ning samuti uuriti internetist erinevaid materjale, millest kõige kasulikumaks osutus .NET-i ametlik dokumentatsioon.

### 5.2.1 Struktuur

Rakenduse keerukuse haldamiseks jaotati rakendus erinevateks kihtideks. Kihiline struktuur aitab hoida koodi korrastatud kujul, mistõttu on arendajal kergem koodi hallata. Samuti saab kihilise arhitektuuriga rakenduse puhul kehtestada kihtidele piirangud. Piirates seda, millised kihid sõltuvad millistest kihtidest, saab teatud rakenduse kihi muudatuse korral leevendada muudatuse mõju nii et see muudatus ei mõjuta tervet rakendust vaid ainult konkreetset kihti. Lisaks sellele võimaldab kihiline arhitektuur lihtsalt välja vahetada rakenduse erinevaid osasid. [52]

Tagarakenduse struktuur on jaotatud kolmeks erinevaks osaks. Tagarakendus jaotub DAL (*Data Access Layer*) ehk andmekihiks, BLL (*Business Logic Layer*) ehk ärioloogika kihiks ning esitusloogika kihiks, mis suhtleb klientrakendusega REST (*Representational State Transfer*) API (*Application Programming Interface*) kontrollrite kaudu. Andmekihi ülesandeks on andmebaasiga suhtlemine päringute kaudu. Ärioloogika kihi ülesandeks on juhtida funktsionaalsust, töödeldes selleks andmekihist saadud andmeid vastavalt

esitusloogika kihist tulnud päringutele ning esitusloogika kihi ülesandeks on vahendada tagarakenduse ja klientrakenduse vahel liikuvat infot.

Rakenduse arendust alustati domeenmudelite defineerimisega. Domeenmudelid on kujutatud C# objektide kujul ning need sümboliseerivad andmebaasis olevaid olemeid. Peale seda alustati tagarakenduse kihtide loomisega. Järgnevalt tutvustatakse tagarakenduse iga kihi olemust.

### **5.2.2 Andmekiht**

Andmebaasiga seotud päringute ning toimingutega seotud koodi eraldas autor andmekihti. Selle kihi eesmärk on eraldada andmebaasiga seotud toimingud ülejäänud rakenduse koodist, et muuta koodi loetavamaks ning eraldada rakenduse erinevateks funktsioonideks mõeldud toimingud üksteisest [53].

Andmekihis kasutati *repository pattern* strateegiat. *Repository pattern* pakub andmetele abstraktsiooni, eraldades andmebaasi loogika muust rakenduse loogikast ja koodist [54]. Seda kasutatakse peamiselt rakenduse äri loogika ja andmetele juurdepääsu kihtide lahtisidumiseks. Tänu sellele on rakenduse kood loetavam, kergemini testitav ning rohkem avatud edasisteks arendusteks. Strateegia kohaselt loodi iga olemi jaoks eraldi hoidla.

Järgmiseks juurutas autor UOW (*Unit of Work*). Selle all mõeldakse ühte transaktsiooni, mis hõlmab endas mitut tüüpi operatsioone. Iga kasutaja tegevuse peale kogutakse kõik selle tegevusega seotud toimingud kokku üheks transaktsiooniks ning seejärel täidetakse see. Kuna mitme väikse transaktsiooni asemel tehakse üks suur, siis on võimalik sellega vähendada andmebaasi koormust ning muuta rakendust seejuures kiiremaks. [55]

### **5.2.3 Äri loogika kiht**

Andmekihi ning esitusloogika kihi vahele lõi autor äri loogika kihi. Äri loogika kiht võtab andmekihi käest tulevad andmed vastu, sooritab nendega vajalikud ärireeglite kohased toimingud või arvutused ning seejärel edastatakse andmed edasi vastavale REST API kontrolleri, mille poole klientrakendus pöördus. Äri loogika kiht garanteerib selle, et rakendusega seotud ärireeglid ning äri loogika on eraldatud teistest rakenduse kihtidest, mistõttu on kood palju loetavam ning arusaadavam [56].

#### 5.2.4 REST API kontrollid

Klientrakendusega suhtlemiseks kasutab tagarakendus autori loodud REST API kontrollereid. REST on arhitektuuristiil, mis seab paika standardid erinevate süsteemide vahel suhtlemiseks [57]. REST API on rakenduse programmeerimisliides, mis järgib REST-i arhitektuurilisi piiranguid ja mis võimaldab kahel tarkvarakomponendil definitsioonide ja protokollide komplekti kasutades üksteisega suhelda [58]. Enamik tänapäeva REST API-sid kasutavad transpordikihtina HTTP (*HyperText Transfer Protocol*) protokoll, kuna HTTP infrastruktuur, serverid ja klienditeegid on juba laialdaselt saadaval [59]. Seoses sellega otsustas ka autor kasutada loodud kontrollereid andmete haldamiseks HTTP päringuid.

Autor genereeris iga domeenmudeli jaoks eraldi REST API kontrolleri. Iga kontrolleri vastab kindlale lõpp-punktile ning igas kontrollereis on hulk meetodeid, mille vastu saab klientrakendus pöörduda. Igas kontrollereis on esindatud standardsed CRUD (*Create, Read, Update, Delete*) operatsioonidega meetodid ning vajadusel lisaks sellele hulk kohandatud meetodeid. Kui klientrakendus pöördub vastava kontrolleri poole, otsitakse kontrollereist õige lõpp-punktiga meetod üles ning käivitatakse see. Pärast toimingute sooritamist tagastatakse vastava meetodi vastus JSON (*JavaScript Object Notation*) kujul klientrakendusele tagasi.

Loodud kontrollid tagastavad andmed DTO (*Data Transfer Object*) objektide kujul. DTO-d on tavalised C# objektid, mis ei sisalda endas ärioloogikat vaid andmeid, mis on vajalikud erinevate rakenduse lülide vaheliseks suhtlemiseks [60]. DTO-de abil saab määrata ära, mis kujul ning milliseid andmeid järgmisele kihile saadetakse. Ebavajalikud väljad, mis ei ole järgmisele kihile ilmingimata vajalikud saab välja filtreerida ning asendada väljadega, mis omavad saadetavas kihis tähtsust.

#### 5.2.5 Turvalisus

REST API kontrollereid turvalisuse tagamiseks on vajalik kontrollida kontrollereid poole pöörduvate kasutajate ligipääsu õigusi. Selle lõputöö raames arendatava veebirakenduses kasutatakse autoriseerimiseks JWT-d (*JSON Web Token*). JWT on avatud standard, mis määratleb kompaktselt turvatud viisi teabe edastamiseks saatja ja vastuvõtja vahel. Kahe osapoole vahel turvaliselt edastatav teave on JSON formaadis ja selle autentisuse

kontrollimiseks allkirjastatakse see digitaalselt kasutades selleks spetsiaalset algoritmi. [61]

Antud veebirakenduses on registreeritud kasutajatel võimalik kuuluda ühte kahest rollist – *User* või *Admin*. Kasutaja registreerimiseks tuleb valida sobilik kasutajanimi, eesnimi, perekonnanimi ning meelepärane salasõna. Kõik registreeritud kasutajad on algselt *User* rollis. Rakenduse administraatoritel on võimalus muuta kasutajate rolle ning sellega ka nende õiguseid. Registreeritud kasutajatel on võimalik muuta enda eesnime, perekonnanime ning salasõna.

JWT genereerimine tagarakenduses toimub juhul, kui veebirakenduses registreeritakse uus kasutaja, logitakse kasutajana sisse või muudetakse olemasoleva kasutaja andmeid. Iga eelnevalt nimetatud toimingu peale genereeritakse uus JWT, et tagada kasutaja autentsus veebirakenduses (Joonis 6).

```
public static string GenerateJWT(IEnumerable<Claim> claims, string signingKey, string issuer, int expiresInDays)
{
    var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(signingKey));
    var signingCredentials = new SigningCredentials(key, algorithm: SecurityAlgorithms.HmacSha512);
    var expires:DateTime = DateTime.Now.AddDays(expiresInDays);
    var token = new JwtSecurityToken(issuer, audience: issuer, claims, notBefore: null, expires, signingCredentials);

    return new JwtSecurityTokenHandler().WriteToken(token);
}
```

Joonis 6. JWT genereerimise meetod.

JWT tähiste autentsust kontrollitakse REST API kontrolleriites sellel hetkel, kui nende poole pöörduetakse. Selleks on märgitud API kontrolleritele vastav annotatsioon: `[Authorize(AuthenticationSchemes = JwtBearerDefaults.AuthenticationScheme)]`. Kontrolleriite meetoditele, mis ei nõua päringus JWT olemasolu, saab märkida annotatsiooni `[AllowAnonymous]` ning meetoditele, mis nõuavad, et päringu teinud kasutaja oleks administraatori rollis on märgitud annotatsioon `[Authorize(AuthenticationSchemes = JwtBearerDefaults.AuthenticationScheme, Roles = „Admin“)]`.

### 5.3 Klientrakendus

Klientrakenduse arendusel kasutati TypeScript programmeerimiskeelt ning React teeki. Esialgse rakenduse struktuur sai genereeritud Create React App teegi abil, mis on ametlikult toetatud viis SPA (*Single Page Application*) React rakenduste loomiseks [62].

Üheleherakendused on rakendused, kus kasutaja tegevuse peale kirjutatakse veebilehe sisu dünaamiliselt üle, selle asemel, et serverist tervet uut lehte alla laadida. Kasutaja tegevuse peale jääb suur osa lehe teabest samaks ning värskendada tuleb vaid teatud osasid leheküljest. See tagab sujuvama kasutajakogemuse ilma pikkade laadimisaegadeta eri lehtede vahel. [63]

Klientrakenduse arendust alustati vajalike lehtede välja kaardistamisega. Järgmiseks koostati esimesed üldisemad komponendid, mis olid vajalikud mitmetel lehtedel nagu näiteks navigatsiooniriba ja jalus. Autor lähenes rakenduse arendusele eesmärgiga saada kõigepealt kõik rakendusega seotud funktsionaalsused töökorra. Kui kõik vajalikud funktsionaalsused said juurutatud, panustas autor aega kasutajaliidese disainile, et tulevastel veebirakenduse kasutajatel oleks võimalikult mugav ning lihtne veebilehel ringi navigeerida.

### 5.3.1 Struktuur

Toetudes autori eelnevate React projektide kogemustele jaotati rakendusega seotud failid eraldi kaustadesse, et tagada koodi parem hallatavus. Kaustade struktuur jaotati järgnevalt:

- components – sisaldab rakendusega seotud üldiseid React komponente, mis on kasutuses mitmel lehel;
- dto – sisaldab objekte, mille kaudu käib andmete vahetus tagarakendusega;
- assets - sisaldab rakenduse lehtedel leiduvaid pilte;
- context – sisaldab rakenduse olekuga seotud faile;
- helpers – sisaldab erinevate toimingute abistavaid funktsioone;
- pages – sisaldab rakendusega seotud erinevaid vaateid;
- services – sisaldab teenuseid, mille abil on võimalik suhelda tagarakendusega;
- styles – sisaldab rakenduse stiiliga seotud faile;
- types – sisaldab TypeScripti jaoks vaja olevaid tüüpe.

Rakenduse lähtefailiks on `index.tsx`, mis sisaldab endas `App` komponenti. `App` on rakenduse peamine komponent, mis toimib kõigi teiste komponentide kontainerina.

Rakenduse globaalsete andmete hoidmiseks kasutati `React Context` teeki. Üldiselt kasutatakse `Context`ti sellisel juhul kui rakenduses olevatele erinevatele komponentidele on vaja teha kättesaadavaks mingisugune hulk globaalseid andmed. Arendatava rakenduse korral on oluline hoida sisselogitud kasutajaga seotud informatsiooni globaalse muutujana. `Context`ti kasutusele võtmiseks installiti teek `NPM (Node Package Manager)` abil, deklareeriti esialgne kasutajaandmete olek ning seejärel loodi selle põhjal `Context`, selle varustaja ning tarbija. Loodud `Context`ti varustaja paigutati `App.tsx` komponendis juurelemendiks, mis tagab selle, et kõikidele lapskomponentidele oleks `Context` kättesaadav. [64]

Erinevate lehekülgede vahel navigeerimiseks kasutati `React Router` teeki. `React Router` abil on võimalik määrata igale komponendile kindel URL, kuhu navigeerides kuvatakse vastava komponendi sisu [65]. Teegi kasutusele võtmiseks laeti see alla `NPM` abil ning seejärel deklareeriti kõik rakenduse erinevad vaated `App.tsx` failis vastavate URL-idega ning nende vaadete vanemkomponendiks määrati `Switch` komponent [66]. See tagab selle, et `Router` kuvab esimese komponendi, mis vastab aadressiriba asukohale. Juhul kui kasutaja üritab navigeerida asukohale, mida rakenduses ei eksisteeri, kuvatakse talle vastava sõnumiga vaade, mis annab teada, et sellist lehte ei ole olemas.

Tagarakendusega suhtlemiseks valiti `Axios`. `Axios` on JavaScripti teek, mis võimaldab teha `HTTP` päringuid `Node.js` keskkonnas või brauseris [67]. Pärast teegi allalaadimist `NPM`-ga loodi `services` kausta failid `Axios` konfiguratsiooni saamiseks, uue `Axios` eksemplari loomiseks ja tagarakenduse `REST API`-dega kommunikatsiooniks. Peale seda oli võimalik erinevates komponentides sooritada `Axios` kliendi abil `HTTP` päringuid, et saada kuvamiseks tagarakendusest andmeid.

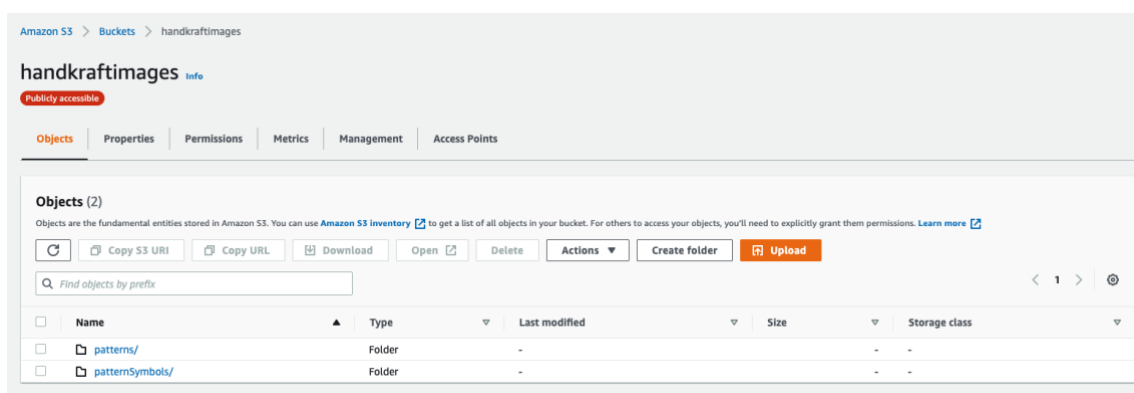
### **5.3.2 Piltide hoiustamine**

Kuna klientrakenduses peab olema võimalik registreeritud kasutajatel luua uusi kohandatud tingmärke ning mustriskeeme ja kuna neid salvestatakse pildi kujul, siis on vaja piltide jaoks hoidlat. Hoidlasse peab olema võimalik salvestada uusi pilte, neid sealt kustutada, muuta ning neid ka kuvamiseks hoidlast välja küsida.



Piltide hoiustamiseks otsustas autor kasutada AWS S3 (*Amazon Web Services Simple Storage Service*) teenust. See on objektide salvestusteenus, mis pakub kerget andmete kättesaadavust ning andmete turvalisust [68]. S3 teenuseid saab kasutada, et salvestada mis tahes andmemahuga faile. Üleslaetud failid paigaldatakse kasutaja poolt loodud konteineritesse, kus määratakse igale üleslaetud failile külge võti, mis käitub kui selle faili unikaalne identifikaator.

AWS S3 teenustele ligipääsemiseks registreeris autor ennast selles keskkonnas kasutajaks ning koostas uue konteineri. Loodud konteineris koostati kaks kausta – üks üleslaetavate muustriskeemide jaoks ning teine tingmärkide jaoks (Joonis 7).



Joonis 7. AWS S3 konteiner muustriskeemide ning tingmärkide kaustadega.

Loodud hoidla ühendamiseks klientrakendusega loodi klientrakenduses .env fail, kuhu salvestati AWS hoidlaga seotud konfiguratsiooni sätted, mille abiga on võimalik klientrakendusel AWS teenustega suhelda. Samuti laeti NPM kaudu alla react-aws-s3-typescript teek, mille eesmärgiks on lihtsustada AWS S3 teenusega suhtlemist.

Mustriskeemi või tingmärgi salvestamisel klientrakenduses, laetakse kõigepealt loodud muustriskeemi või tingmärgi pilt AWS S3 konteinerisse ning sealt tagastatakse URL, mis viitab üleslaetud pildile. Seejärel seotakse antud URL salvestatava JSON objekti külge, mispeale pöördutakse vastava objektiga HTTP päringut kasutades tagarakenduse poole, et salvestada uus kirje andmebaasi.

### 5.3.3 Abistavad teegid ja komponendid

Lisaks eelnevalt mainitud teekidele kasutas autor mitmeid abistavaid teeke ning olemasolevaid komponente. Kuna komponentide ja teekide ise juurutamine oleks võtnud palju aega ning kuna see ei ole antud lõputöö põhieesmärk, siis otsustas autor kasutada

juba olemasolevaid lahendusi. Järgnevalt tuuakse välja peamised teegid ning komponendid, mis aitasid rakenduse arendusele kaasa:

- Formik – Formik on avatud lähtekoodiga lihtne teek Reacti jaoks, mis aitab kaasa eelkõige vormi loomisele, oleku haldamisele, väljade valideerimisele ja vormi esitamisele [69]. Kui Reactis võib vormide loomine ja juhtimine nõuda märkimisväärset hulgal koodi, siis Formikut kasutavate Reacti vormidega on võimalik säästa palju aega koodi kirjutamisest;
- Bootstrap – Bootstrap on populaarne CSS raamistik reageerivate lehekülgede arendamiseks [70]. Antud raamistik sisaldab käepäraseid korduvkasutatavaid koodijuppe, mis on koostatud HTML, CSS ja JavaScripti abiga. Bootstrapi kasutades on võimalik säästa aega CSS koodi kirjutamisest;
- React Canvas Draw – React Canvas Draw on HTML canvas elemendi põhjal koostatud komponent [71]. Kuna rakenduse nõuete kohaselt peab olema võimalus kasutajal luua uut tingmärki, siis otsustas autor valida tingmärgi kujutamise viisiks vabakäelise joonistuskomponendi. Antud komponent toetab mitmeid funktsionaalsusi nagu näiteks joonistatud kujutise tagasivõtmine, joonistuse salvestamine ning laadimine ja palju muud;
- JWT-decode – JWT-decode on teek, mis aitab dekodeerida Base64URL kodeeringuga JWT tähiseid [72]. Seda teeki kasutas autor tagarakenduse poolt saadetud JWT tähiste dekodeerimiseks, et määrata ära kas sisselogitud kasutaja on administraatori rollis või tavalise kasutaja rollis;
- React-pdf – React-pdf on teek, mille abil on võimalik koostada Reactis PDF faile [73]. PDF-ide koostamiseks on eksporditud suur hulk Reacti komponente, mis võimaldavad koostada PDF dokumente. Dokumente saab kujundada erinevate CSS atribuutidega. Antud teeki kasutati mustriskeemide ning projekti kavandite PDF kujul allalaadimise toetamiseks.

## 6 Testimine

Antud peatükis kirjeldatakse lõputöö praktilises osas valminud veebirakenduse testimisprotsessi. Enne kliendipoolset testimist, testis autor iseseisvalt rakendust, kontrollides, kas kõik funktsionaalsused on täidetud ning töökorras ja et rakenduse disainis ei esineks ootamatuid vigu.

### 6.1 Autoripoolne testimine

Autoripoolne testimine viidi läbi kasutades MacOS operatsioonisüsteemi, mis töötab Monterey versiooni peal. Veebirakendust testiti järgnevates brauserites: Google Chrome (versioon 99.0.4844.74), Microsoft Edge (versioon 99.0.1150.46), Safari (versioon 15.2) ja Mozilla Firefox (versioon 98.0.1). Autor testis rakendust enne kliendipoolset testimist, et tagada rakenduse toimivus ning avastada võimalike vigu rakenduse käitumises ning väljanägemises.

Testiti järgmisi funktsionaalsusi:

- Kasutaja registreerimine ning sisse logimine;
- Kasutaja andmete muutmine;
- Muustriskeemide valmistamine, salvestamine, muutmine, kustutamine;
- Muustriskeemi ruudustiku mõõtude muutmine;
- Tingmärkide loomine, muutmine, kustutamine;
- Projektide loomine, salvestamine, muutmine, kustutamine;
- Muustriskeemi ning projekti kavandi salvestamine ning allalaadimine PDF formaadis;
- Administraatori rollis andmebaasi kirjade lisamine ning erinevate olemasolevate kirjade muutmine, kustutamine.

Testimise käigus suutis autor leida Mozilla Firefox brauseris probleemi, kus ükski API päring ei läinud läbi. Brauseri arendaja konsoolist selgus, et päring ebaõnnestus, kuna CORS (*Cross-Origin Resource Sharing*) oli blokeeritud. Probleemi lahendamiseks uuris autor internetist vastava probleemi kohta ning leidis ka lahenduse. Nimelt selgus, et Firefox ei usalda vaikumisi Windowsi sertifikaadipoes olevaid sertifitseerimisasutusi ning see põhjustas ka API päringute ebaõnnestumisi. Autor lahendas probleemi määrates Firefox brauseri konfiguratsioonis *security.enterprise\_roots.enabled* väärtuse tõeseks.

Pärast eelnevalt kirjeldatud vea parandamist testis autor uuesti Firefox brauseris rakenduse erinevaid sektsioone. Seekord läksid kõik API päringud ootuspäraselt läbi ning rakenduse vaadete väljanägemises hälbeid ei esinenud. Google Chrome ning Microsoft Edge brauserites ei suutnud autor testimise järgselt tuvastada ühtegi viga.

## **6.2 Kliendipoolne testimine**

Valmis saadud veebirakendust anti testida kliendile, kes on kogeenud silmuskuduja ning on tegelenud käsitööga mitmeid aastaid. Kliendipoolne tagasiside on äärmiselt kasulik, sest tulevikus hakkavad valminud rakendust kasutama samade eesmärkidega käsitöölised, seega on oluline, et rakendus oleks kliendi tahtmiste ning soovide järgi üles ehitatud.

Rakendust anti kliendile kasutada autori arvutis. Kliendi testimine sisaldas endas kõikide samade funktsionaalsuste testimist nagu autoripoolse testimise peatükis, välja arvatud administraatori rollis kasutaja toimingud. Kokkuvõttes lõppes kliendipoolne testimine positiivsete tulemustega ehk ühtegi probleemi ei õnnestunud tuvastada.

## **6.3 Rakenduse edasised arendused**

Kliendipoolse testimise järgselt said üles märgitud rakenduse potentsiaalsed edasised arendused:

- Projekti kavandisse võiks olla võimalik lisada pilte tööproovidest ja valminud töödest;
- Projekti loomisel võiks olla võimalus lisada visandeid;

- Projekti väljundi keel teha valitavaks;
- Tingmärkide loomisel võiksid olla toetatud erinevad värvid.

Lisaks kliendi tagasisidele toetuvate edasiste arenduste mõtetele on plaanis autoril muuta tulevikus veebirakenduse disaini veelgi ilusamaks ning lisada tugi ka inglise keelele. Samuti on plaanis autoril lisada juurde erinevaid tingmärke. Hetkeseisuga on rakenduses esindatud minimaalne hulk tingmärke, millega on võimalik luua kõige algelisemaid mustriskeeme. Lisaks sellele on autoril plaanis tulevikus uurida ka selle kohta, kuidas integreerida rakendust käsitööd õpetavatesse koolidesse, et lihtsustada õpilaste õppimisprotsessi.

## 7 Tulemused

Käesoleva lõputöö tulemusena valmis veebipõhine rakendus, mis abistab ning kiirendab käsitöölise mustriskeemide ja projektide tegemise protsessi. Kõik rakendusega seotud funktsionaalsed nõuded said edukalt täidetud.

Kliendiga arutades kalkuleeriti loodud lahenduse ligikaudne kasutegur välja. Selgus, et ühe projektiga seotud mustriskeemi peale läheb tavaliselt umbes 1-3 A3 paberit ja projekti kavandi peale ligikaudu 1-3 A4 lehte. Kuna lõputöö raames valminud veebirakenduses toimub mustriskeemide ning projektide kirjeldamine elektrooniliselt, siis on selles keskkonnas paberi kulu olematu. Samuti arutas autor kliendiga ka rakenduse ajalise võidu üle. See faktor on oluline, kuna mida vähem käsitööline saab kulutada aega käsitööga seotud eelnevate protsesside peale, seda rohkem aega jääb reaalse käsitöö jaoks. Ajaliselt kiirendab veebirakenduses mustriskeemide ja projektide loomine ligikaudu 1/3 võrra reaalsele käsitööle eelnevat protsessi. Kui ühe korra on mustriskeem ja projekt juba veebirakenduses disainitud, siis on järgmistel samadel käsitöö tellimustel seda hea taaskasutada. Samuti lahendab valminud rakendus ka arhiveerimis ja jagamis probleeme, mistõttu jääb käsitöölistel rohkem aega reaalse käsitöö jaoks.

Valminud veebirakenduse peamised vaated on välja toodud lisas 2.

Valminud rakenduse lähtekood asub GitHubi hoidlas, mille link on leitav lisas 3.

## 8 Kokkuvõte

Antud lõputöö eesmärgiks oli muuta tekstiilkäsitöö projektide ja mustriskeemide loomisprotsess elektrooniliseks seoses sellega, et nende kujutamine paberile on ebaefektiivne. Selle eesmärgi täitmiseks loodi veebipõhine rakendus, kus on võimalik selle kasutajatel luua ja jagada tekstiilkäsitöö projekte ja mustriskeeme.

Selle lõputöö raames keskenduti põhjalikumalt silmuskudumise käsitöötehnikale ning sellega seotud tingmärkide kohta. Sellest tulenevalt uuriti enne veebirakenduse loomist põhjalikult silmuskudumise protsessi ning mustriskeemide kohta. Samuti võrreldi juba olemasolevaid sarnase eesmärgiga lahendusi, kaardistati ära nende positiivsed ning negatiivsed küljed ning märgiti ära kliendiga kooskõlas uue lahenduse jaoks nõuded. Järgnevalt uuriti ning analüüsiti erinevaid tarkvara arendusega seotuid tehnoloogiaid ning nende tulemuste põhjal sai valitud kõige efektiivsemad tööriistad, millega lõputöö praktilist osa asuti tegema. Tagarakenduse arenduse jaoks valiti analüüsi järgselt C# programmeerimiskeel ja .NET raamistik. Andmebaasi valikuks osutus PostgreSQL ning klientrakenduse programmeerimiskeeleks valiti TypeScript ja raamistikuks React. Arenduskeskkonna valikul tagarakenduse puhul osutus sobivaks programmiks JetBrains Rider ja klientrakenduse puhul Visual Studio Code.

Rakenduse esialgse prototüübi valmimise järel testiti seda kõigepealt autori poolt ning seejärel kliendi poolt. Rakenduse autoripoolse testimise järgselt tuvastati brauseri spetsiifiline probleem, millele leiti lahendus. Kliendipoolne testimine lõppes positiivsete tulemustega. Testimise järgselt sai autori ja kliendi poolt kaardistatud ka rakenduse potentsiaalsed edasised arendused, millega autor kavatseb tulevikus edasi tegeleda.

## Kasutatud kirjandus

- [1] A. Ojaste, „Etnograafilised koed - kirjamine ja roosimine,“ 2010. [Võrgumaterjal]. Available: [https://www.tlu.ee/opmat/ku/MLT6008/silmuskudumise\\_ajaloost.html](https://www.tlu.ee/opmat/ku/MLT6008/silmuskudumise_ajaloost.html). [Kasutatud 07 01 2022].
- [2] A. Pink, S. Reimann ja K. Jõeste, Eesti silmuskudumine I. Tavad ja tehnikad, Türi: Saara Kirjastus, 2014.
- [3] H. Koger, Kudumismustrid, Tallinn: Eesti Raamat, 2021.
- [4] Profistyle70, „Background patterns with knitting needles. Simple raised knitting patterns,“ [Võrgumaterjal]. Available: <https://profistyle70.ru/en/recepty/fonovye-uzory-spicami-prostye-relefnye-uzory-spicami-kletka-iz-vytyanutyh/>. [Kasutatud 07 01 2022].
- [5] Stitchfiddle, „About Stitch Fiddle,“ [Võrgumaterjal]. Available: <https://www.stitchfiddle.com/en/company/about>. [Kasutatud 14 01 2022].
- [6] American Graphics Institute, „What is Adobe Illustrator,“ 17 03 2022. [Võrgumaterjal]. Available: <https://www.agitraining.com/adobe/illustrator/classes/what-is-adobe-illustrator>. [Kasutatud 14 01 2022].
- [7] Stitchmastery. [Võrgumaterjal]. Available: <https://stitchmastery.com/>. [Kasutatud 14 01 2022].
- [8] Lucidchart, „What is a Database Model,“ [Võrgumaterjal]. Available: [https://www.lucidchart.com/pages/database-diagram/database-models#section\\_0](https://www.lucidchart.com/pages/database-diagram/database-models#section_0). [Kasutatud 2022 01 22].
- [9] IBM, „Defining primary key-foreign key relationships,“ 26 03 2021. [Võrgumaterjal]. Available: <https://www.ibm.com/docs/en/doc/3.9.0.0?topic=model-defining-primary-keyforeign-key-relationships>. [Kasutatud 22 01 2022].
- [10] Matillion, „The Types of Databases (with Examples),“ 18 11 2020. [Võrgumaterjal]. Available: <https://www.matillion.com/resources/blog/the-types-of-databases-with-examples>. [Kasutatud 22 01 2022].
- [11] C. Curator, „What Are Object-Oriented Databases And Their Advantages,“ 06 09 2019. [Võrgumaterjal]. Available: <https://www.c-sharpcorner.com/article/what-are-object-oriented-databases-and-their-advantages2/>. [Kasutatud 22 01 2022].
- [12] PostgreSQL, „About,“ [Võrgumaterjal]. Available: <https://www.postgresql.org/about/>. [Kasutatud 23 01 2022].
- [13] S. Bhatia, „PostgreSQL vs MySQL: Everything You Need to Know in 2022,“ 07 04 2022. [Võrgumaterjal]. Available: <https://hackr.io/blog/postgresql-vs-mysql>. [Kasutatud 23 01 2022].
- [14] H. Kathuria, „The Most Popular Databases for 2022,“ 11 01 2022. [Võrgumaterjal]. Available: <https://learnsql.com/blog/most-popular-databases-2022/>. [Kasutatud 23 01 2022].



- [15] T. Graham, „6 Things for Developers to Know about Postgres,“ 15 11 2019. [Võrgumaterjal]. Available: <https://thenewstack.io/6-things-for-developers-to-know-about-postgres>. [Kasutatud 23 01 2022].
- [16] Oracle Corporation, „What is MySQL?,“ [Võrgumaterjal]. Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. [Kasutatud 27 01 2022].
- [17] Talend, „What is MySQL? Everything You Need to Know,“ [Võrgumaterjal]. Available: <https://www.talend.com/resources/what-is-mysql/>. [Kasutatud 27 01 2022].
- [18] S. M. M. Tahaghoughi ja H. E. Williams, „Learning MySQL,“ [Võrgumaterjal]. Available: <https://www.oreilly.com/library/view/learning-mysql/0596008643/ch01s01.html>. [Kasutatud 27 01 2022].
- [19] GeeksforGeeks, „Introduction of MS SQL Server,“ 06 06 2021. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/introduction-of-ms-sql-server/>. [Kasutatud 27 01 2022].
- [20] SQLServerTutorial, „What is SQL Server,“ [Võrgumaterjal]. Available: <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/>. [Kasutatud 27 01 2022].
- [21] Y. Yehuda, „6 Reasons Why Microsoft SQL Is Alive and Well,“ 03 05 2018. [Võrgumaterjal]. Available: <https://dzone.com/articles/6-reasons-why-microsoft-sql-is-alive-and-well>. [Kasutatud 27 01 2022].
- [22] Oracle Corporation, „What is Java technology and why do I need it?,“ [Võrgumaterjal]. Available: [https://www.java.com/en/download/help/whatis\\_java.html](https://www.java.com/en/download/help/whatis_java.html). [Kasutatud 10 02 2022].
- [23] J. Hartman, „What is Java? Definition, Meaning & Features of Java Platforms,“ 16 04 2022. [Võrgumaterjal]. Available: <https://www.guru99.com/java-platform.html>. [Kasutatud 10 02 2022].
- [24] StackOverflow, „Developer Survey 2021,“ 2021. [Võrgumaterjal]. Available: <https://insights.stackoverflow.com/survey/2021/>. [Kasutatud 10 02 2022].
- [25] R. Shankar, „10 Best Java Frameworks to Use in 2022,“ 30 11 2021. [Võrgumaterjal]. Available: <https://hackr.io/blog/java-frameworks>. [Kasutatud 10 02 2022].
- [26] P. Pedamkar, „What is Java Hibernate,“ [Võrgumaterjal]. Available: <https://www.educba.com/what-is-java-hibernate/>. [Kasutatud 10 02 2022].
- [27] Microsoft, „A tour of the C# language,“ 18 03 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>. [Kasutatud 10 02 2022].
- [28] Microsoft, „What is .NET? Introduction and overview,“ 11 03 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/core/introduction>. [Kasutatud 10 02 2022].
- [29] The PHP Group, „Introduction,“ [Võrgumaterjal]. Available: <https://www.php.net/manual/en/intro-what-is.php>. [Kasutatud 12 02 2022].
- [30] Laravel LLC, „Installation,“ [Võrgumaterjal]. Available: <https://laravel.com/docs/9.x>. [Kasutatud 12 02 2022].
- [31] Medium, „Top 5 Programming languages for Web development in 2022,“ 13 01 2021. [Võrgumaterjal]. Available: <https://medium.com/javarevisited/top-5->

- programming-languages-for-web-development-in-2021-f6fd4f564eb6. [Kasutatud 19 02 2022].
- [32] Buzz Interactive, „10 Best Front End Development Languages To Develop Your Website,“ [Võrgumaterjal]. Available: [buzzinteractive.co/best-front-end-development-languages/](https://buzzinteractive.co/best-front-end-development-languages/). [Kasutatud 19 02 2022].
- [33] MDN contributors, „What is JavaScript?,“ 23 03 2022. [Võrgumaterjal]. Available: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript). [Kasutatud 19 02 2022].
- [34] Hack Reactor, „What is JavaScript used for?,“ 26 08 2021. [Võrgumaterjal]. Available: <https://www.hackreactor.com/blog/what-is-javascript-used-for>. [Kasutatud 19 02 2022].
- [35] J. Hartman, „TypeScript Vs JavaScript: What is the Difference?,“ 12 03 2022. [Võrgumaterjal]. Available: <https://www.guru99.com/typescript-vs-javascript.html>. [Kasutatud 19 02 2022].
- [36] C. Dugas, „Pros and Cons of Using a Front-End Library (or Framework),“ 01 09 2021. [Võrgumaterjal]. Available: <https://bluemodus.com/articles/pros-and-cons-of-using-a-front-end-library-or-framework>. [Kasutatud 19 02 2022].
- [37] T. Patel, „Does Your Web Application Need A Front-End Framework,“ 26 05 2020. [Võrgumaterjal]. Available: <https://www.thirdrocktechkno.com/blog/does-your-web-application-need-a-front-end-framework/>. [Kasutatud 19 02 2022].
- [38] State of JS, „Front-End Frameworks,“ [Võrgumaterjal]. Available: <https://2021.stateofjs.com/en-US/libraries/front-end-frameworks/>. [Kasutatud 19 02 2022].
- [39] Meta Platforms Inc., „Tutorial: Intro to React,“ 2022. [Võrgumaterjal]. Available: <https://reactjs.org/tutorial/tutorial.html>. [Kasutatud 19 02 2022].
- [40] StackOverflow, „Stack Overflow Trends,“ [Võrgumaterjal]. Available: <https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangular>. [Kasutatud 19 02 2022].
- [41] R. Barger, „Is React a Library or a Framework? Here's Why it Matters,“ 12 04 2021. [Võrgumaterjal]. Available: <https://www.freecodecamp.org/news/is-react-a-library-or-a-framework/>. [Kasutatud 19 02 2022].
- [42] Vue, „Introduction,“ [Võrgumaterjal]. Available: <https://vuejs.org/guide/introduction.html>. [Kasutatud 20 02 2022].
- [43] V. Cromwell, „Between the Wires: An interview with Vue.js creator Evan You,“ 30 05 2017. [Võrgumaterjal]. Available: <https://www.freecodecamp.org/news/between-the-wires-an-interview-with-vue-js-creator-evan-you-e383cbf57cc4/>. [Kasutatud 20 02 2022].
- [44] Altexsoft, „The Good and the Bad of Angular Development,“ 25 03 2020. [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/>. [Kasutatud 20 02 2022].
- [45] Angular, „What is Angular?,“ [Võrgumaterjal]. Available: <https://angular.io/guide/what-is-angular>. [Kasutatud 20 02 2022].
- [46] J. Rabelo, „Primary Key,“ 14 08 2020. [Võrgumaterjal]. Available: <https://www.techopedia.com/definition/5547/primary-key>. [Kasutatud 05 03 2022].

- [47] Microsoft, „Introduction to tables,“ [Võrgumaterjal]. Available: <https://support.microsoft.com/en-us/office/introduction-to-tables-78ff21ea-2f76-4fb0-8af6-c318d1ee0ea7>. [Kasutatud 05 03 2022].
- [48] M. Jones, „Integers vs GUIDs - The Great Primary Key Debate,“ 2016. [Võrgumaterjal]. Available: <https://exceptionnotfound.net/integers-vs-guids-the-great-primary-key-debate/>. [Kasutatud 05 03 2022].
- [49] InformIT, „The Cost of GUIDs as Primary Keys,“ 08 03 2002. [Võrgumaterjal]. Available: <https://www.informit.com/articles/article.aspx?p=25862&seqNum=5>. [Kasutatud 05 03 2022].
- [50] Microsoft, „Entity Framework 6,“ 14 10 2020. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/ef/ef6/>. [Kasutatud 11 03 2022].
- [51] Microsoft, „LINQ to Entities,“ 15 09 2021. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/language-reference/linq-to-entities>. [Kasutatud 11 03 2022].
- [52] Microsoft, „Common web application architectures,“ 14 04 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>. [Kasutatud 11 03 2022].
- [53] E. Reitan, „Create the Data Access Layer,“ 05 05 2022. [Võrgumaterjal]. Available: [https://docs.microsoft.com/en-us/aspnet/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/create\\_the\\_data\\_access\\_layer](https://docs.microsoft.com/en-us/aspnet/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/create_the_data_access_layer). [Kasutatud 11 03 2022].
- [54] Microsoft, „Design the infrastructure persistence layer,“ 13 04 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-design>. [Kasutatud 11 03 2022].
- [55] J. Singh, „Unit of Work in Repository Pattern,“ 30 11 2018. [Võrgumaterjal]. Available: <https://www.c-sharpcorner.com/UploadFile/b1df45/unit-of-work-in-repository-pattern/>. [Kasutatud 11 03 2022].
- [56] S. Mitchell, „Creating a Business Logic Layer (C#),“ 05 05 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/aspnet/web-forms/overview/data-access/introduction/creating-a-business-logic-layer-cs>. [Kasutatud 12 03 2022].
- [57] Codecademy, „What is REST?,“ [Võrgumaterjal]. Available: <https://www.codecademy.com/article/what-is-rest>. [Kasutatud 12 03 2022].
- [58] Red Hat, „What is a REST API?,“ 08 05 2020. [Võrgumaterjal]. Available: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>. [Kasutatud 12 03 2022].
- [59] L. Fischer, „A Beginner's Guide to HTTP and REST,“ 13 02 2022. [Võrgumaterjal]. Available: <https://code.tutsplus.com/tutorials/a-beginners-guide-to-http-and-rest--net-16340>. [Kasutatud 12 03 2022].
- [60] A. Zang, „,NET Basics: DTO (Data Transfer Object),“ 02 02 2022. [Võrgumaterjal]. Available: <https://www.telerik.com/blogs/dotnet-basics-dto-data-transfer-object>. [Kasutatud 12 03 2022].
- [61] J. Kanjilal, „Implementing JWT Authentication in ASP.NET Core 5,“ 08 06 2021. [Võrgumaterjal]. Available: <https://www.codemag.com/Article/2105051/Implementing-JWT-Authentication-in-ASP.NET-Core-5>. [Kasutatud 13 03 2022].

- [62] Facebook Inc., „Getting Started,“ [Võrgumaterjal]. Available: <https://create-react-app.dev/docs/getting-started/>. [Kasutatud 19 03 2022].
- [63] K. Lawson, „What are Single Page Applications and Why Do People Like Them so Much?,“ 19 07 2018. [Võrgumaterjal]. Available: <https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application>. [Kasutatud 19 03 2022].
- [64] Meta Platforms Inc., „Context,“ [Võrgumaterjal]. Available: <https://reactjs.org/docs/context.html>. [Kasutatud 19 03 2022].
- [65] M. Manjunath ja M. Wanyoike, „React Router v5: The Complete Guide,“ 27 10 2020. [Võrgumaterjal]. Available: <https://www.sitepoint.com/react-router-complete-guide/>. [Kasutatud 19 03 2022].
- [66] React Training, „Switch,“ [Võrgumaterjal]. Available: <https://v5.reactrouter.com/web/api/Switch>. [Kasutatud 19 03 2022].
- [67] Axios, „Getting Started,“ [Võrgumaterjal]. Available: <https://axios-http.com/docs/intro>. [Kasutatud 19 03 2022].
- [68] Amazon Web Services Inc., „What is Amazon S3?,“ [Võrgumaterjal]. Available: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>. [Kasutatud 19 03 2022].
- [69] Formium Inc., „Overview,“ [Võrgumaterjal]. Available: <https://formik.org/docs/overview>. [Kasutatud 20 03 2022].
- [70] A. Ouellette, „What is Bootstrap: A Beginner's Guide,“ 27 12 2021. [Võrgumaterjal]. Available: <https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/>. [Kasutatud 20 03 2022].
- [71] M. Beierling-Mutz, „React Canvas Draw,“ [Võrgumaterjal]. Available: <https://github.com/embiem/react-canvas-draw>. [Kasutatud 20 03 2022].
- [72] Auth0, „Jwt-decode,“ [Võrgumaterjal]. Available: <https://github.com/auth0/jwt-decode>. [Kasutatud 20 03 2022].
- [73] React-pdf, „React-pdf,“ [Võrgumaterjal]. Available: <https://react-pdf.org/>. [Kasutatud 20 03 2022].

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

Mina, Marko Gordejev

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose Veebirakendus tekstiilkäsitöö mustriskemide ja projektide loomiseks, mille juhendaja on Toomas Lepikult
  - 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

---

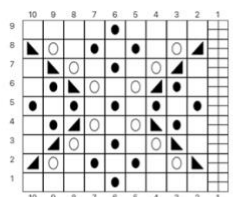
<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud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

## Lisa 2 – Rakenduse peamised vaated



### Loo oma mustrid...

Siin saad luua projektide jaoks mustreid.  
Kui tunned, et tingimärke on vähe, ära muretse, registreerudes kasutajaks saad luua ise omale meelepärased!



### Sinu projektid kõik ühes kohas...

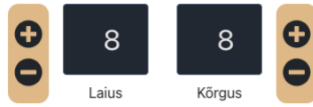
Vihkad paberimajandust? See jääb nüüdsest minevikku!  
Siin saad hoida oma projektid kõik turvaliselt ja mugavalt ühes kohas.

### Registreeri kasutajaks!

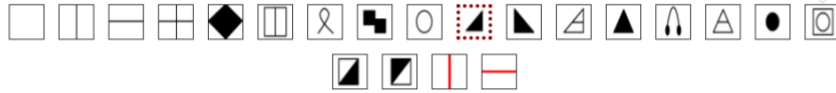
Registreeri kasutajaks ja saad luua tingimärke, projekte ning vaadata ka teiste kasutajate projekte.  
See võtab kõigest mõned sekundid aega!



## Ruudustiku mõõdud



LISA UUS TINGMÄRK

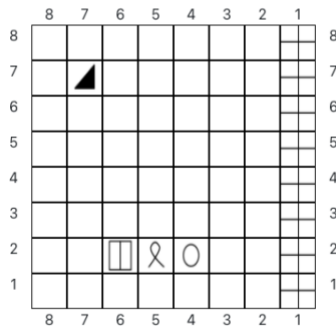


## Patendisilmus

Vardale jääb kudumata üle tõstetud silmus koos lõngakeeruga

LÄHTESTA

LOHISTA



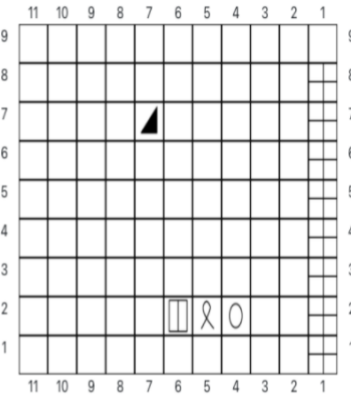
LAE ALLA PDF

SALVESTA MUSTRITESSE

Joonis 9. Mustrite koostamise lehekül.

## Minu mustrid

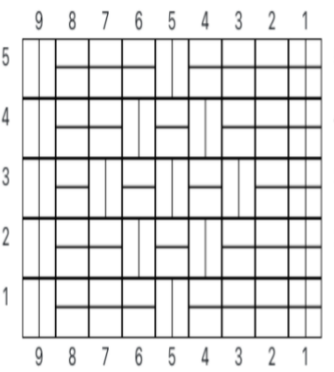
Lisa uus muster



**Seeliku muster**  
Pooleli

Muuda

Kustuta



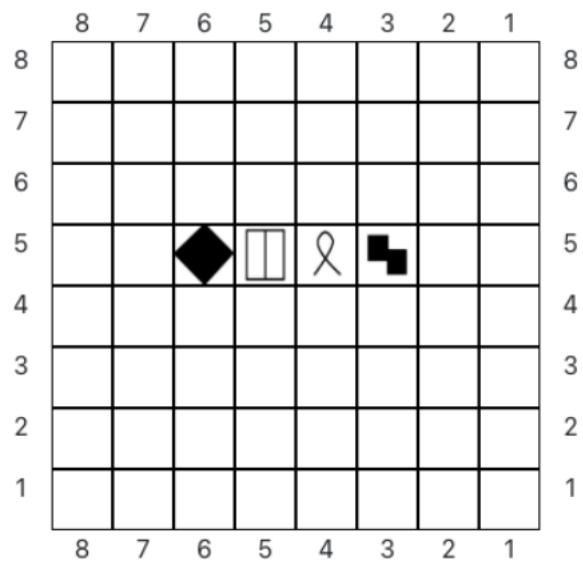
**Muster 2**  
VALMIS

Muuda





Kustuta

Joonis 10. Minu mustrite lehekülj.





### Tingmärgid

-  Eelmise rea silmusest kootakse parempidi juurde uus silmus
-  Kahe silmuse vahelt keerdsilmusena välja kootud uus silmus
-  Pahempidise võttega üle tõstetud silmus - Lõng jääb töö taha
-  Parempidine keerdsilmus - Kudumiseks pistetakse varras sisse silmuse tagumisest poolest

Joonis 11. PDF väljund valmistatud mustri skeemist.

## Administraatori koobas

[Kategoriad](#)[Raskusastmed](#)[Mustrid](#)[Tingmärgid](#)[Projektid](#)

Joonis 12. Administraatori lehekülg.

## Kategoriad

Kampsunid	▼
Kleidid	▼
Sallid	▲
Nimi	
<input type="text" value="Sallid"/>	
Kirjeldus	
<input type="text" value="Sallid, kaela soojendamiseks"/>	
<a href="#">Muuda</a> <a href="#">Kustuta</a>	
Mütsid	▼
Kindad	▼

[Lisa kategooria](#)

Joonis 13. Kategooria muutmine administraatori leheküljel.

## Minu projektid

[Lisa uus projekt](#)

						Q Otsi projekte	X
Nimi	Autor	Kategooria	Raskusaste	Valmis	Privaatne		
Roheline müts	TestUser1	Mütsid	Kerge			<a href="#">Vaata</a>	<a href="#">Muuda</a>
Soojad kindad	TestUser1	Kindad	Keskmine			<a href="#">Vaata</a>	<a href="#">Muuda</a>
Mugav kampsun	TestUser1	Kampsunid	Keskmine			<a href="#">Vaata</a>	<a href="#">Muuda</a>
Sinine sall	TestUser1	Sallid	Kerge			<a href="#">Vaata</a>	<a href="#">Muuda</a>

5 kirjet |< < 1-4-st > >|

Joonis 14. Minu projektide lehekülg.

### Sinine sall

Kustuta projekt

Salvesta muudatused

Valmis

Privaatne

Projekti nimi

Sinine sall

Kirjeldus

Laius: 20cm  
Pikkus: 150cm

Kategooria

Sallid

Raskusaste

Kerge

Töövahendid

Ringvarras 5.0 40cm

Materjalid

KATIA CONCEPT Cotton-Merino (50g - 105m) 2 tokki  
70% puuvill, 30% merinovill

Tööproov

10 cm = 28 silmust  
10 cm = 13 muustrida

Töökäik

Loon 55 silmust ja koon muustrit 200 muustrida

### Mustrid

Loo uus muster

Lisa olemasolev muster

Eemalda muster

	8	7	6	5	4	3	2	1	
4		□				□			4
3				□				□	3
2		□				□			2
1				□				□	1
	8	7	6	5	4	3	2	1	

Valepööratud muster



Joonis 15. Uue projekti loomise lehekülg.

# Sinine sall

## Kirjeldus

Laius: 20cm  
Pikkus: 150cm

## Kategooria

Sallid

## Raskusaste

Kerge

## Töövahendid

Ringvarras 5.0 40cm

## Materjalid

KATIA CONCEPT Cotton-Merino (50g - 105m) 2 tokki  
70% puuvill, 30% merinovill

## Tööproov

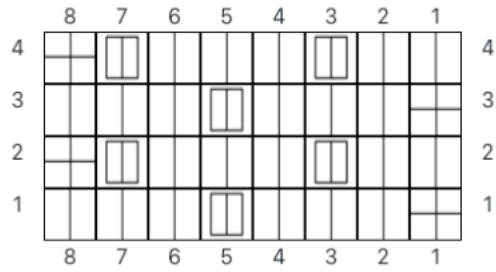
10 cm = 28 silmust  
10 cm = 13 mustirida

## Töökäik

Loon 55 silmust ja koon mustrit 200 mustirida




Joonis 16. Projekti PDF väljund 1. leht.

## Mustrid



Valepatent muster

## Tingmärgid

-  Ületõstetud ääresilmus - Ületõstetud ääresilmus
-  Parempidine silmus - Parempidine silmus
-  Pahempidise võttega üle tõstetud silmus - Lõng jääb töö taha

Joonis 17. Projekti PDF väljund 2.leht.

## Minu kasutaja

Kasutajanimi

Eesnimi

Perekonnanimi

Praegune salasõna

Uus salasõna

Joonis 18. Minu kasutaja lehekül.

## **Lisa 3 - Valminud rakenduse lähtekood**

Link lähtekoodile: <https://github.com/markogo/HandKrafts>