

TALLINNA ÜLIKOOL

Haapsalu kolledž

Rakendusinformaatika

Juho Kalberg

HAAPSALU KOLLEDŽI E-ÕPPE PLATVORMI ESIRAKENDUSE  
LOOMINE KOOS DISAINI JA ESIRAKENDUSE KOODI  
SÜNKRONISEERIMISEGA

Diplomitöö

Juhendaja: Priidu Paomets, MSc

Haapsalu 2023

## SISUKORD

MÕISTED .....	4
SISSEJUHATUS .....	6
1. EELTÖÖD. PROBLEEMI SELGITAMINE JA ÜLESANNETE PÜSTITAMINE.	
EESMÄRK. ....	8
1.1. Diplomitööle eelnenud valikpraktika kursus. Prototüüp. ....	8
1.2. Uurimisküsimused .....	10
1.3. Diplomitöö eesmärk .....	11
2. NÕUDED RAKENDUSELE .....	12
2.1. Tellijapoolsed nõuded rakendusele .....	12
2.2. Kasutajauuringud .....	13
2.2.1. Lõppkasutajate intervjuud .....	13
2.2.3. Küsitlused .....	14
3. RAKENDUS .....	17
3.1. Valikpraktikas valminud osa rakendusest ja järgnevad ülesanded .....	17
3.2. Disainitootenid .....	18
3.3. Disaini visuaalse osa põhimõtted .....	19
3.3.1. Kujunduse stiili valik .....	21
3.4. Disainifaili loomise põhimõtted. Disainiprogramm. ....	24
3.4.1 Märkused spetsiifilisemate töökorralduslike muudatuste kohta .....	25
3.4.2. Abiprogramm – Figma Tokens Studio .....	26
3.4.3. Värvipaletti koostamine hue, ehk tooni baasil .....	28
3.5. Disainifailist koodi .....	30
3.5.1 Esirakenduses kasutatud tehnoloogiad .....	30
3.5.2 Disainitootenite koodiks muutmine omal jõul .....	32
3.5.3. Amazon Style Dictionary, Tailwindi töökorraldus .....	35
3.5.4 Tailwindi jaoks sobilikus formaadis koodi kasutamise põhimõtted ja kasutamine .....	36
3.5.5. Tulemus ja edasiarendus .....	38

KOKKUVÕTE .....	39
SUMMARY .....	40
KASUTATUD ALLIKAD .....	41
LISA 1. POOLSTRUKTUEERITUD INTERVJU LÕPPKASUTAJAGA	
LISA 2. KASUTATUD E-ÕPPEKESKKONDADE JA DISAINIEELISTUSTE KÜSITLUS	
LISA 3. KASUTAJA TEEKOND, JUTUSEIN	
LISA 4. IDEESEIN	

## MÕISTED

**Disaini arenduskomplekt** (*UI-kit*): disainifailis loodud stiilikomponentide ja näidislehtede komplekt (InVisionApp Inc, 2023).

**Disainisüsteem**: kasutajaliidese loomiseks mõeldud dokumentatsioon ja komponentide teek (Interaction Design Foundation, 2023).

**Disainitooten** (*Design token*): disaini alginformatsiooni ühik, millel on unikaalne nimetus ning väärtus ja võivad olla lisaks ka tüüp ning kirjeldus (Design Tokens Community Group, 2023).

**Esirakendus** (*Front-End*): rakenduse see osa, mida lõppkastaja ekraanil näeb ja läbi mille rakendusega suhtleb. Kliendi poolne osa rakendusest. (Lemonaki, 2022).

**Hüpertekst-märgistuskeel** (*HTML, HyperText Markup Language*): enimlevinud kodeerimissüsteem (tekstivorming) veebidokumentide loomiseks. HTML koodid ehk märgendid määravad ära selle, kuidas veebileht arvutiekraanil välja näeb (Vallaste, s.a.).

**Kaskaadlaadistik** (*CSS, Cascading Style Sheets*): veebilehtede valmistajatele ja kasutajatele mõeldud laadistik. Laadilehed (*style sheets*) kirjeldavad, kuidas HTML dokumente esitada kuvaril, printeril või kõnesüntesaatorist kostva kõnena. Laadilehed lubavad kasutajal muuta sadade dokumendilehtede väljanägemist üheainsa CSS faili muutmise teel (Vallaste, s.a.).

**Kasutajakogemus** (*UX, User Experience*): see on viis mida ettevõtte või toode pakub kasutajale või kogemus, mille kasutaja saab ettevõtte teenuse või toote kasutamisest (Nielsen Norman Group, s.a.).

**Kasutajaliides** (*UI, User interface*): rakenduse see osa, mille abil kasutaja suhtleb rakendusega (Vallaste, s.a.).

**Pistikprogramm** (*Plugin*): tarkvaramoodul, millega lisatakse rakendusele lisavõimalusi (Vallaste, s.a.).

**Raamistik** (*Framework*): struktuur, mis toetab loodavat platvormi või tarkvara (Vallaste, s.a.).

**Rakendus, rakendusprogramm** (*Application*): Rakendus on lõppkasutaja tarbeks kirjutatud iseseisev terviklik programm (Vallaste, s.a.).

**Rakendusliides** (*API, Application Programming Interface*): reeglistik ja liides, mille alusel rakendus kasutab teise rakenduse teenuseid (Vallaste, s.a.).

**Seadmetundlikkus**: erinevatel ekraanisuurustel sobivalt skaleeruva esirakenduse omadus.

**Sisuedastusvõrk** (*CDN, Content Delivery Network*): Internetis või suures intranetis laialipaigutatud sisuga süsteem, kus sisust tekitatakse palju koopiaid ja puhverdatakse neid üle kogu võrgu (Vallaste s.a.).

**Tagarakendus** (*Back-End*): kõik rakenduse sisemised ja taustal töötavad protsessid, mis salvestavad ja pärivad informatsiooni. Serveri poolne osa rakendusest. (Lemonaki, 2022).

**Tume režiim** (*Dark mode*): kujunduslahendus telefonil, arvutiekraanil jne, kus kasutatakse musta või tumedat tausta heledama teksti kuvamiseks (*Cambridge Dictionary*).

**Veebirakendus** (*Web Application*): Veebis paiknev tarkvara. Termin võib tähendada peaaegu kõike veebiga seonduvat, sh veebilehitsejat või muud klientprogrammi, mis on võimeline veebi poole pöörduma (Vallaste, s.a.).

## SISSEJUHATUS

21. sajandi ülikool ja õppimine on seoses tehnika arenguga jõudnud etappi, kus suur osa õppest toimub interneti vahendusel (Chernev, 2023) (Rungta, 2023). Eriti kiirendas sellise õppevormi kasutuselevõttu ja sellega harjumist Covid19 aegne olukord (World Economic Forum, 2020) (Swiss School of Business and Management Geneva, 2022).

Haapsalu kolledž ei ole selles osas erand. Tuntumad kolledži õpetajate poolt kasutatavatest e-õppe keskkondadest on Moodle<sup>1</sup>, Google Classroom<sup>2</sup> ja Cisco Academy<sup>3</sup> ning Haapsalu kolledži enda materjalide hoidla Sahtel<sup>4</sup>. Lisaks kasutavad üliõpilased omal valikul Udey<sup>5</sup>, Skillshare<sup>6</sup>, Pluralsight<sup>7</sup> jms.

Haapsalu kolledži jaoks on need keskkonnad kas liiga suured, nagu nt Moodle ja Google Classroom või liiga ainespetsiifilised, nagu Cisco Academy. Siinse diplomitöö ja sellele eelnenud valikpraktika kursuse raames läbiviidud intervjuudest ja küsitlustest järeldub, et kolledži senine e-õppe korraldus toimub moel, kus iga õppejõud valib ise, kuidas ta oma töö korraldab, milliseid abivahendeid, sh e-õppe keskkondi kasutab. Tagajärjeks on erinevate õpikeskkondade paljususe, mis muudab õpingute ja kursuste järgimise keeruliseks. Üliõpilastel peab olema konto mitmes erinevas keskkonnas, mistõttu järje hoidmine oma õpingutel on keeruline ja asjad võivad ununeda.

Samuti puudub kolledžil kontroll võimalike lahenduste rakendusliidese üle, mistõttu ei saa kursust koostada täielikult oma äranägemise järgi ega selle sisu integreerida ka muudesse või tulevikus loodavatesse rakendustesse.

Eelnevast lähtuvalt esitas kolledž 2022. aasta sügissemestril valikpraktika raames üliõpilastele tellimuse, töötada välja kolledžile sobiv ja ühtne e-kursuste platvorm. Selle töö käigus jõuti valmis Githubis hallatava ainekursuste süsteemiga ning selle kuvamisega eraldi veebilehel.

---

<sup>1</sup> <https://moodle.edu.ee>

<sup>2</sup> <https://classroom.google.com/>

<sup>3</sup> <https://www.netacad.com>

<sup>4</sup> <https://start.hk.tlu.ee/sahtelbeta/login.php>

<sup>5</sup> <https://www.udemy.com>

<sup>6</sup> <https://www.skillshare.com>

<sup>7</sup> <https://www.pluralsight.com>

Siinne diplomitöö on järgmine etapp, mille eesmärgiks on Haapsalu kolledži õppetöös kasutatav, sisselogimist võimaldav, kasutajate arengut salvestav ning kuvav ja üliõpilaste õppetöös kasutatav süsteem. Diplomitöö on osa tervikust, millega arendatakse valmis esirakendus ja disain. Esimesest etapist saadud kogemus ja valminud rakendusest saadud tagasiside põhjal on lisaeesmärgiks seatud kiiremat arendusprotsessi võimaldava disaini ja esirakenduse sünkroniseeritud süsteem.

Lähtuvalt eesmärgist on sõnastatud järgmised peamised uurimisküsimused:

- kuidas peab lõppkasutaja saama rakendust kasutada;
- millised on nõuded disainile;
- milline on parim esirakenduse ja disaini haldamise töökorraldus.

Uurimisküsimuste põhjal on püstitatud järgmised ülesanded:

- luua disainifail sellisel moel, et see ühtiks tehniliselt esirakendusega;
- luua esirakendus, mis on seadistatud võtma vastu disainist saadatud koodi;
- kirjeldada, milliseid tehnilisi abivahendeid on kogu protsessiks vaja;
- kirjeldada, milliselt võiks olla üles ehitatud disainifail, mida on võimalik automaatselt esirakenduse koodi viia;
- uurida välja kasutajate seisukohad disaini suhtes;
- luua esirakenduse disain;
- kirjeldada sellise töökorraldusega seotud kitsaskohad, millal selline korraldus sobib ja millal mitte.

Tellija soovide ja tingimuste välja selgitamiseks kasutati intervjuusid, lõppkasutajate eelistuste väljaselgitamiseks poolstruktureeritud intervjuusid ja küsitlusi. Samuti viidi läbi testid valikpraktika raames valminud rakendusega, selgitamaks välja lõppkasutaja hakkamasaamine rakenduses.

Diplomitöö koosneb sissejuhatusest, kolmest põhipeatükist, kahekümne ühest alampeatükist ning kokkuvõttest, milles esitatakse järeldused ja soovitused. Esimeses peatükis kirjeldan eeltöid ja seda, kuidas ja miks ma valitud eesmärgi seadsin. Teises peatükis kirjeldan tellija nõudmisi ja kasutajate eelistusi. Kolmandas peatükis kirjeldan disainivalikuid ning kuidas ma rakenduse lõin.

Kõik täpsema viite või kirjelduseta joonised on autori loodud.

### **3. EELTÖÖD. PROBLEEMI SELGITAMINE JA ÜLESANNETE PÜSTITAMINE. EESMÄRK.**

Selles peatükis annan ülevaate diplomitööle eelnenud tööst, selle põhjal formeerunud ülesande püstitusest ja täpsematest uurimisküsimustest.

#### **3.5. Diplomitööle eelnenud valikpraktika kursus. Prototüüp.**

Diplomitööle eelnes 2022. aasta sügisel kolmanda kursuse raames valikpraktika kursus, mis on siinse diplomitöö aluseks ja eelkäijaks. Valikpraktika käigus loodi töötav, kuid pooliku funktsionaalsusega rakendus – puudus kasutajate haldus, sisselogimine, õpingu edenemise jälgimine ja isikustatud töölaud<sup>8</sup>.

Kasutajakogemuse osas selgitati intervjuude ja küsimustike abil välja kasutajate soovid ja vajadused ning loodi kasutajateekonnad. Saadud informatsiooni alusel loodi UX- ja UI-kavandid (joonis 1), esialgne disain ning klikatav prototüüp. Kuigi disaini funktsionaalne ja kasutajakogemuslik külg sai lahendatud põhjalikult, jäi disaini esteetiline pool nõ tagasihoidlikuks<sup>9</sup> (joonis 2).

Tellija, ehk Haapsalu kolledži soovil baseerus rakendus Githubis<sup>10</sup> hallataval sisul, mida läbi rakendusliidese kuvatakse esirakenduses. Valikpraktika raames jäi lahendamata kasutajate sisselogimine ja nende edenemise salvestamine.

Esirakenduses oli kasutusel mallimismootor Handlebars<sup>11</sup> ja CSS-raamistik Tailwind CSS<sup>12</sup>. Arenduskeskkonnaks oli Node.js<sup>13</sup> platvorm.

Valikpraktika ajal tekkis ka küsimus disaini visuaalse poole valikutest – milline on hea, kuidas seda hinnata, kes hindab jne. Nendele küsimustele ei jõutud valikpraktika ajal vastata, kuid sellega seoses tekkis idee automatiseerida disaini viimine koodi, mis lubaks muuta koodi kiiremini ja seega muuta valmis rakenduse väljanägemist tavalisest kiiremini ja odavamalt, ehk vastavalt nõudmiste muutumisele.

---

<sup>8</sup> <https://github.com/tluhk/rif20-valikpraktika-1/tree/1.0.48.1>

<sup>9</sup> <https://github.com/tluhk/rif20-valikpraktika-1/tree/1.0.48.1/docs/content/ux>

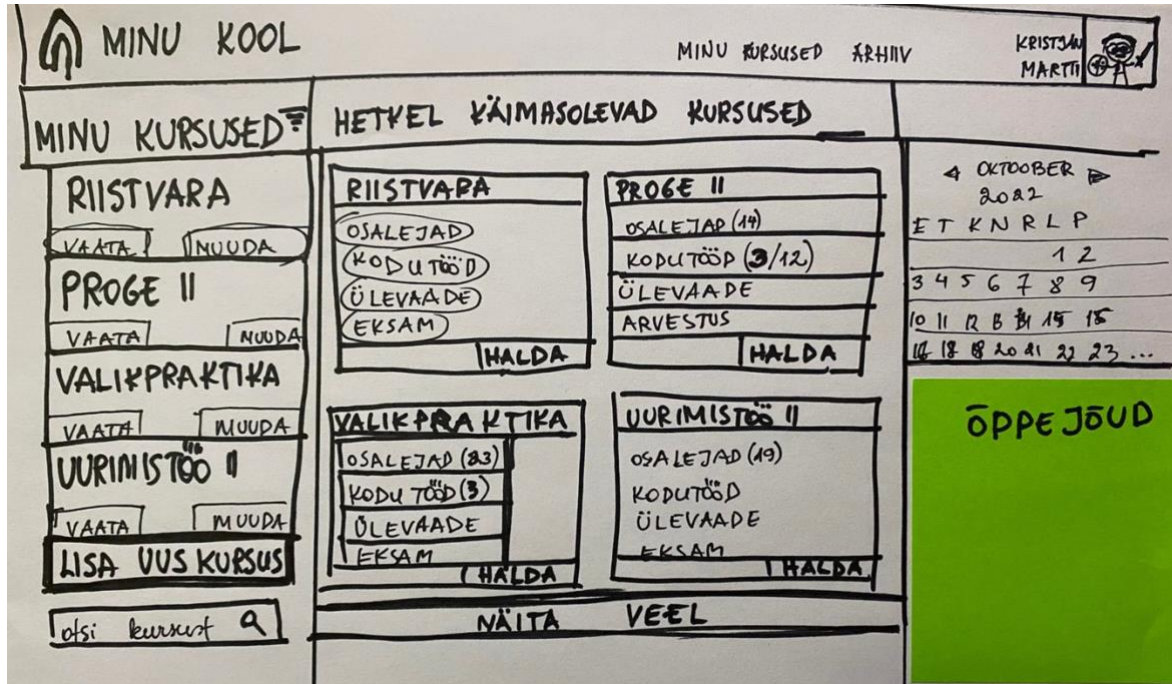
<sup>10</sup> <https://github.com>

<sup>11</sup> <https://handlebarsjs.com>

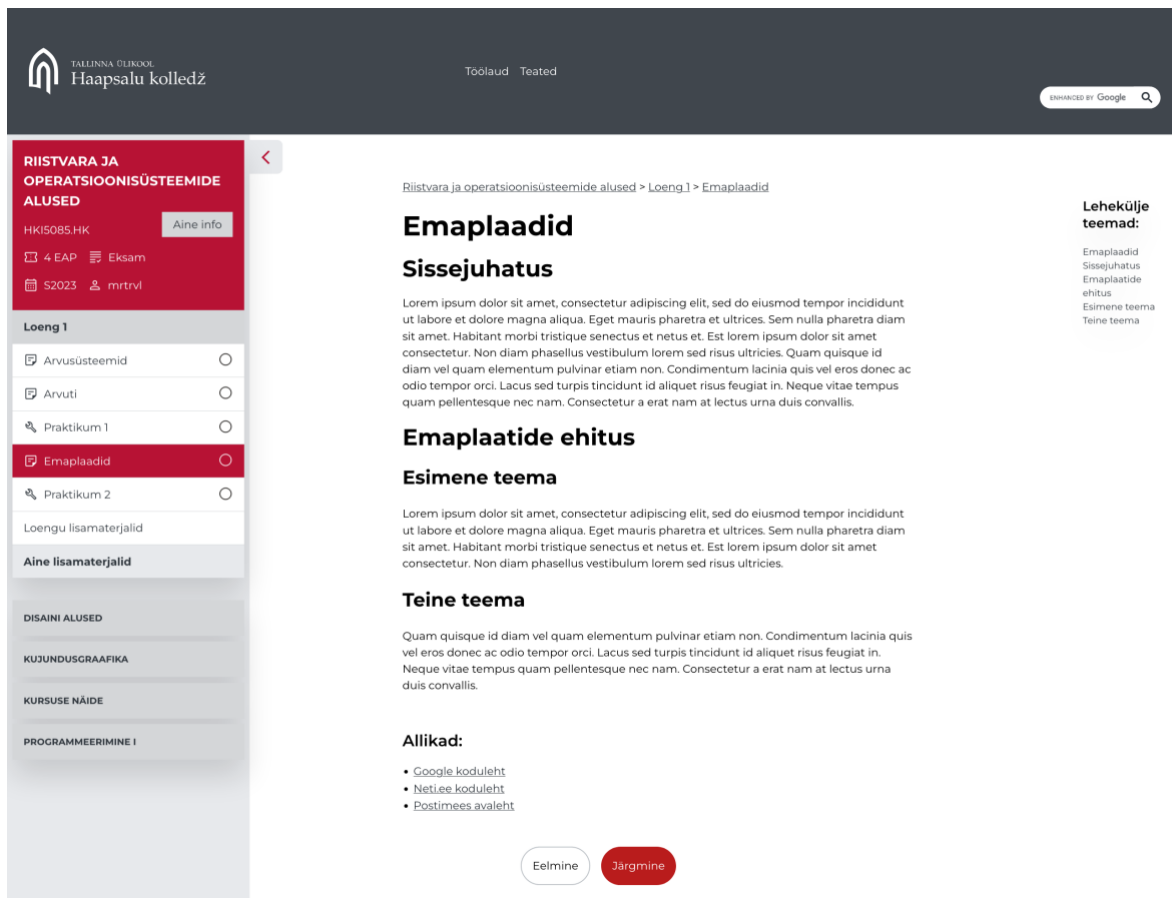
<sup>12</sup> <https://tailwindcss.com>

<sup>13</sup> <https://nodejs.org>





Joonis 1. Valikpraktika raames tehtud UX-prototüübi näidis. Tiina Kukli joonis.



Joonis 2. Valikpraktika raames valminud rakenduse kasutajaliides. Kuvatõmmis.

## 1.2. Uurimisküsimused

Siinne töö on üks osa tervikust, mis koosneb teoreetilisest tööst ja rakendusest, mille löid kaks Haapsalu kolledži üliõpilast. Praktiline osa jaotub omakorda tagarakenduseks, esirakenduseks, disaini arenduseks ja disaini ning esirakenduse koodi sünkroniseerimiseks. Minu teha tööst oli osa, mis puudutas esirakendust ja disaini – sellise disainifaili loomine, millise sisu saab võimalikult automaatselt esirakenduse koodi eksportida ja esirakenduse seadistamine moel, et see disaini võimalikult automaatselt vastu võtaks.

Ideaalse ja testitud disaini loomine ei olnud siinse töö eesmärgiks, küll aga keskkonna võimaldamine, et tulevased üliõpilased saaksid sellele keskenduda.

Sobiva lõpplahenduse leidmiseks püstitasime koos kaasarendajaga üldised uurimisküsimused:

- kuidas peab üliõpilane saama rakendust kasutada ja mida peab platvorm talle selleks pakkuma;
- kuidas peab õpejõud saama rakendust kasutada ja milleks peab platvorm sellest lähtuvalt võimeline olema;
- kas valikpraktika käigus tehtud rakendus vastab kasutajakogemuse seisukohast püstitatud eesmärkidele;
- kas valikpraktika käigus loodud visuaal on lõppkasutajale meeltemööda.

Lähtuvalt oma töö skoobist püstitasin ka järgnevad uurimisküsimused, millele diplomitööga vastuse leida soovisin:

- millistele tingimustele peab vastama disainifail, et disaini ja koodi sünkroniseerida;
- millise kompetentsiga peab olema disainer, et valmistada ette esirakenduse jaoks sobilik disainifail;
- kuidas seadistada esirakendus, et kood tuleks disainifailist rakendusse automaatselt;
- millised on sellise töökorralduse (disaini koodi automaatne saatmine esirakendusse) nõrgad kohad;
- milliste projektide puhul tasub selline töö ette võtta.

Üldistele küsimustele vastuste leidmiseks kasutasime intervjuusid, küsimustikke ja testimiseks valikpraktika raames valminud mittetäieliku funktsionaalsusega rakendust.

Minu skoobi jaoks püstitatud küsimustele vastuste saamiseks lõin disainifaili ja esirakenduse, mis olid teineteisega seotud ja käisin kogu protsessi läbi.

Disaini sobiva esteetika valik on üsna raske mitmel põhjusel:

- täieliku uuringu läbiviimine tähendaks suuremahulist uurimistööd, see ei tarvitse tihti ajaliselt võimalik olla;
- võiks lähtuda hetkemoest, mida on võimalik välja selgitada veebist leitavate erialaste trendiajakirjade abil, kuid mood vananeb ja sel juhul oleks mõne aastaga veebileht vananenud;
- võiks lasta kliendil otsustada, kuid sel juhul ei tarvitse see vastata lõpptarbija ootustele.

Seega lahendades disaini võimalikult automaatne ehk kiire viimine esirakenduse koodi, saaks vajalikke disainimuutusi rakendada märksa kiiremini. Ettevõtetele tähendaks see rahalist võitu, üliõpilaste poolt hallataval koolilehel mahuks arendusperiood ühte semestrisse.

### 1.3. Diplomitöö eesmärk

Eelnevast tulenevat sai diplomitöö eesmärgiks luua ühtne disaini ja esirakenduse süsteem, milles disain muudetakse esirakenduse koodiks automaatselt.

Eesmärgi täitmine sisaldas järgnevaid ülesandeid:

- luua esirakendusega tehnilises mõttes kooskõlas olev disainifail, anda soovitusel, milline see olema peaks, et see ühtiks tehniliselt esirakendusega;
- luua esirakendus, mis on seadistatud võtma vastu disainist saadetud koodi;
- kirjeldada, milliseid tehnilisi abivahendeid on kogu protsessiks vaja;
- kirjeldada, millisel võiks olla üles ehitatud disainifail, mida on võimalik automaatselt esirakenduse koodi viia;
- kirjeldada, sellise töökorraldusega seotud kitsaskohad, millal selline korraldus sobib ja millal mitte.

## 2. NÕUDED RAKENDUSELE

Selles peatükis annan ülevaate tellijapoolsetest nõuetest ja kasutajate eelistustest, mis sain teada intervjuudest ja küsitlustest. Samuti kirjeldan uurimismeetodid, mille abil vastused välja selgitasin.

### 2.1. Tellijapoolsed nõuded rakendusele

Tellimus Haapsalu kolledži e-koolitusplatvormi loomiseks tuli Haapsalu kolledži poolt siinsele diplomitööle eelnenud valikpraktika aine raames 2022 sügisel. Ülesandega tegeles lisaks siinkirjutajale neli kaastudengit. Koolipoolsete tingimuste väljaselgitamiseks viisime läbi kolm intervjuud tellija esindajatega<sup>14,15,16</sup>, mille käigus küsitlesime kahte IT õppejõudu ja saime selle alusel peamised nõuded, mida tellija soovib:

- vaja on rakendusliidesega lahendust, mille poole saab andmete pärimiseks pöörduda, mille üle on kolledžil kontroll ja milles asub kogu andmestik;
- näha peab olema üliõpilaste edenemine õppetöös;
- rakendus peaks olema lihtne;
- tagarakendus võiks kasutada Githubi võimalusi – versioonihaldust, andmete hoiustamist, ühist arendustööd ja läbi andmete kuvamist läbi api loodavas rakenduses.

Diplomitöö raames 2023 talvel, viisime koos tagarakenduse arendajaga läbi intervjuud kolledži Rakendusinformaatika õppesuuna juhiga<sup>17,18</sup>. Nendes intervjuudes täpsustusid nõuded:

- üliõpilastel on kasutajakontod. Kontosid loob esialgu õppejõud;
- üliõpilane näeb sisse logitult enda õppeaineid;
- üliõpilane saab aineid läbida ja nii tema ise kui õppejõud saavad tema edenemist õpingutes näha;
- rakendus peaks olema samal ajal ka õppevahend.

---

<sup>14</sup> <https://tinyurl.com/yldsiseend>

<sup>15</sup> <https://tinyurl.com/marttiraavel>

<sup>16</sup> <https://tinyurl.com/kristjantoots>

<sup>17</sup> <https://tinyurl.com/martti-esimene>

<sup>18</sup> <https://tinyurl.com/martti-teine>

Eelkõige on oluline, et üliõpilane saaks süsteemi kasutada, õppejõu kasutajaliides esimeses etapis on vähemtähtis. Kuna võtsime kasutusele Githubi rakendusliidese ja andmete hoiu, siis soovitas õppesuuna juht kasutada niipalju Githubi võimalusi kui vähegi võimalik. Seega toimub suurem osa rakenduse funktsionaalsusest kasutades Githubi – andmete salvestamine, loomine, gruppide haldamine, loengumaterjalide versioonide haldus.

Teises intervjuus täpsustasid disaini ja esirakenduse seisukohalt detailid:

- sisselogimine Githubi kasutajaga läbi loodava rakenduse;
- õppejõud saab uuendusi üliõpilaste tegevuse kohta (nt kui üliõpilane esitab kodutöö).

Sellest järeldotsid sisselogimisvaate, üldise kasutajavaate ja ainevaate vajadus.

Kolmandas intervjuus mängisime läbi kogu lõppkasutaja ja õppejõu teekonna veebilehel, millest selgus kõikide vaadete vajadus (Lisa 1)<sup>19</sup>. Teekonna läbimängimiseks kasutasime jutuseina tehnikat (Tromp, 2017).

## **2.2. Kasutajauuringud**

Kasutajauuringud toimusid nii diplomitööle eelnevas faasis kui diplomitöö raames poolstruktureeritud intervjuude ja küsitluste abil.

### **2.2.1. Lõppkasutajate intervjuud**

Eeltöö kasutajakogemuse leidmiseks tegime valikpraktika aine raames, kus kaardistasime kasutajate vajadused ning koondasime need kasutajalugude faili<sup>20</sup>. Diplomitöö raames, 2023. aasta talvel, intervjuerisime koos kaasarendajaga kolme lõppkasutajat – teise ja kolmanda kursuse Rakendusinformaatika üliõpilasit. Intervjuud toimusid Microsoft Teamsi<sup>21</sup> videokõne keskkonnas. Viisime läbi poolstruktureeritud intervjuud, milles uurisime esmalt intervjueritavate nõudeid ja ootusi e-õppe platvormile ja seejärel testisime valikpraktika raames valminud rakendust, andes kasutajatele ülesandeks jõuda kindla kursuseni, ehk soovisime näha, kas rakenduse kasutajakogemuse osa vajab olulisi täiendusi või on see arusaadav. Lisaks palusime hinnangut senise rakenduse disainile ja pärisime küsitletavate eelistusi disaini kohta üldisemalt (Lisa 2).

---

<sup>19</sup> <https://tinyurl.com/hkjutusein>

<sup>20</sup> <https://tinyurl.com/kasutajalood>

<sup>21</sup> <https://www.microsoft.com/et-ee/microsoft-teams/group-chat-software/>

### 2.2.2. Intervjuude tulemused ja analüüs

Vastajaid said etteantud ülesandega hakkama hästi, millest võib järeldada, et seni tehtu on kasutajakogemuse disaini osas hea. Saime mõned ettepanekud külghenüü lihtsustamiseks, mida ka kasutame rakenduse uues versioonis.

Vastajate jaoks olulised punktid üldises funktsionaalsuses olid järgmised:

- käimasoleva semestri ained eespool;
- õppejõudude poolsed teavitused;
- oluline on näha õpingute edenemist ehk progressiriba;
- loogiline navigatsioon;
- tähtjad ja teavitused, mis ajaks midagi esitada või millal millegagi tegeleda;
- õppejõudude nimed kursuste juures;
- ainepunktid aine juures.

Olulised punktid disainis olid järgmised:

- pigem ümarad vormid kui kandilised;
- selged vormid – paksud, tugevad pealkirjad;
- mobiilivaade;
- tume režiim;
- visuaalne pool, esteetika on oluline.

### 2.2.3. Küsitlused

Disainieelistuste olulisuse selgitamiseks viisin läbi küsitluse. Küsitlus toimus Google Forms keskkonnas 2023. aasta veebruaris. Küsitluses osales 57 Haapsalu kolledži tudengit, kellest 23 vastajat olid Rakendusinformaatika eriala, ülejäänud Tervisejuhtimise, Liiklusohutuse ning Käsitöötehnoloogijate ja disaini üliõpilased.

Küsimuste abil soovisin teada saada, milliseid e-õppe keskkondi kasutatakse ja kas kasutajatel on tekkinud tähelepanekuid rakenduste visuaalse poole kohta, et leida positiivseid näiteid, millest enda töös eeskuju võtta.

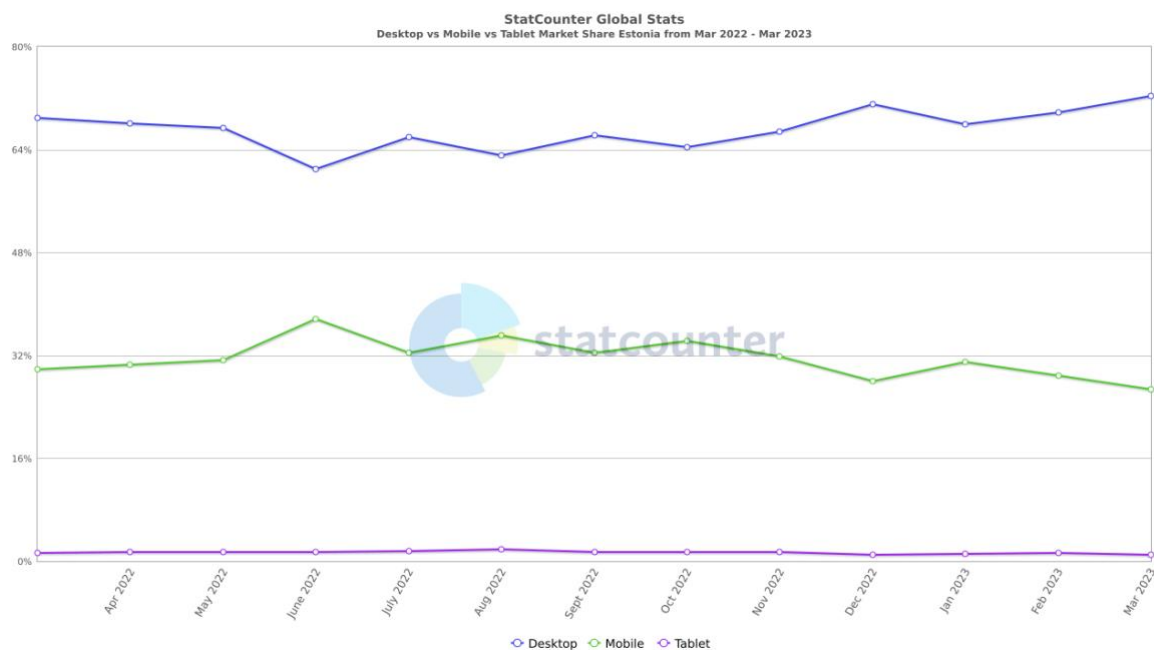
Samuti soovisin teada saada kui oluliseks peavad kasutajad disaini visuaalset poolt, kui oluline on tume režiim ja kuivõrd oluline on rakenduse kasutamise võimalus mobiilis.

## 2.2.4. Küsitluste tulemused ja analüüs

Kõik vastanud üliõpilased (n=57) kasutasid vähemalt ühte e-õppe platvormi (Moodle), mis tähendab, et Haapsalu kolledži õpe on juba korraldatud mingis osas e-õppe vormis. Kahte e-õppe platvormi kasutas 36 üliõpilast ja kolme või enam 19 üliõpilast. Kasutamise aktiivsus sõltus erialast, nii näiteks kasutasid kõik Rakendusinformaatika eriala üliõpilased enamat kui kolme e-õppe keskkonda.

Läbiviidud küsitlus jäi liiga nõrgaks, et selle baasil kujunduse jaoks järeldusi teha. Selle jaoks oleks pidanud koostama spetsiifilisema küsimustiku või looma eristuvamad näidislehed, mida kasutajad oleks saanud hinnata. Küll aga sai ülevaate selliste disainiaspektide nagu mobiilis kasutamine, hele ja tume režiim ning kujundustrende järgiva visuaali olulisusest. 55 vastajat andis hinnangu sellele kui oluline on veebilehe kujunduse moodsus või trendikus – 37 pidas seda oluliseks või väga oluliseks. 57-st vastanust 37 pidas e-õppe võimalust mobiilis oluliseks või väga oluliseks. Tumedat režiimi pidas oluliseks või väga oluliseks 22 üliõpilast 56-st sellele küsimusele vastajast.

Analüüsi põhjal võib öelda, et küsitlus oleks võinud sisaldada ka küsimusi kooli tuleku viisi kohta, mis oleks selgitanud paremini, kuivõrd keegi mobiilis õpet praktikas kasutaks. Näiteks kui tuled õppesessioonile kaugemalt bussiga, on sul ka tegelikult aeg mobiilis keskkonda kasutada. Teisalt on kaasajal esirakenduse arendus erinevatele seadmetele niivõrd elementaarne, et seda arvestamata ei võiks lihtsalt esirakendust enam arendada. Seda nii seetõttu, et nagu küsitlus näitas, eelistavad kasutajad kaasaegset disaini, kui ka seetõttu, et nutitelefonidega tarbitakse interneti Eestis juba ca 30% (joonis 3) (Statcounter, 2023) veebi kasutavatest seadmetest ja vaadates tõenäolisi suundumusi, siis USA-s lausa 50% (Statcounter, 2023). Ja viimaks, veebirakendus peaks olema mobiili jaoks arendatud ka seetõttu, et vastasel juhul ei pääse leht otsingumootorites enam nii hästi leitavale kohale (Google Search Central, 2023). Muidugi on siin ka erijuhud, nt kui vaid laua- või sülearvutile loodud arendus on vajalik või vältimatu.



Joonis 3. Seadmed, millega Eestis internetti kasutatakse. Joonis Statcounter, 2023.



### 3. RAKENDUS

Käesolevas peatükis kirjeldan esmalt, millise osa valikpraktikas valmis saime ja mis osa tuli diplomitööga juurde lisada. Kirjeldan, miks otsustasin suurema disainilahenduse kasuks, mis koosneb disainifaili ja esirakenduse sünkroniseeritud koodist. Seejärel selgitan millistest põhimõtetest lähtuvalt valmis disain, millest lähtusin disaini visuaalses pooles, millel baseerub disaini funktsionaalne pool, miks ja milline on disaini fail, millised valikud tegin esirakenduses ja miks.

#### 3.1. Valikpraktikas valminud osa rakendusest ja järgnevad ülesanded

2022. sügissemestril saime valikpraktika raames koos nelja kaasüliõpilasega Haapsalu kolledžilt tellimuse e-õppe rakendusele, millest jõudsime valmis arendada järgmised osad:

- kasutajakogemuse disaini põhimõtted ja kavandid;
- kasutajaliidese disain;
- prototüüpimistarkvaras Figma loodud prototüüp;
- Tailwind CSS stiiliraamistikku ja Hanlebars templiitismootorit kasutav esirakendus;
- Githubis asuv tagaraendus ning esi- ja tagarakenduse ühendus Node.js arenduskeskkonas kasutades Express.js raamistikku<sup>22</sup>.

Kuna Haapsalu kolledž soovib rakenduse kasutusele võtta, vajab rakendus lisaks järgmisi funktsionaalsusi (vt peatükk 2.1. Tellijapoolsed nõuded rakendusele):

- õpiteekonna jälgimine;
- kasutajate sisselogimine GitHubi kasutajakontodega;
- õpilane saab märkida sisuteemasid loetuks;
- õpetaja saab ülevaate kui kaugel on iga õpilane mõnel ainekursusel;
- andmebaas, kus hoitakse infot selle kohta, kui kaugel iga õpilane ainekursusega on;
- lisada õigused kasutajatele.

Valikpraktika raames valminud rakenduse disaini üheks nõrgaks kohaks oli visuaalne pool, kuna tööaeg oli piiratud. Samas tuli aga küsitlustest välja, et kaasaegsus ja esteetika on lõppkasutajale olulised. Eesteetika väärtust kasutajakogemuse osas tuuakse välja selle ala

---

<sup>22</sup> <https://expressjs.com>

spetsialistide uuringutes. Väidetavalt lepivad kasutajad väikeste kasutatavusvigadega, kui disain on visuaalselt meeldiv (Moran, 2017). Disaini visuaalse poole hindamine ise toimub ülikiirelt – vaid 50 millisekundit kulub aega visuaalse atraktiivsuse hindamiseks (Lindgaard et al., 2011).

Kuna tellijalt disainile suuri nõudmisi ei olnud, kuid disain, nagu eelnevalt viidatud, on oluline, siis otsustasin lahendada töö moel, luues süsteemi, mida on võimalikult lihtne disaini poolest täiustada. Tulevased üliõpilased saavad ise süsteemi atraktiivsusel ja trendikusel silma peal hoida, arendamata selleks uut versiooni. See tähendab seda, et disainiprogrammis hallatakse kujundust nõ disainitokenite abil, mis omakorda tuuakse esirakenduse koodi automaatselt.

### 3.2. Disainitokenid

Disainitokenid on mõiste, mille esmakasutust omistatakse tihti Salesforce'le (Bolton & Ferrua, s.a) Siiski on disainerid kasutanud disainisüsteeme luues mõistet ka varem. Suures plaanis on disainitokenid (Design Tokens Community Group, 2023) sarnased muutujatele, sisaldades deklaratsiooni ja väärtust, kuid nad on tehnoloogiateülesed ning neile saab lisada rohkem sisu, nt kirjelduse ja omakorda saab neist tuletada nt CSS-i või Javascripti muutujad (koodinäide 1). Seega, ehitades juba disainiprogrammi üles moel, et kujunduse põhikomponente – värve, kirjastiile, nurgaraadiuseid, veeriseid, puhvreid jms väärtusi hallatakse läbi disainitokenite, on lihtne need samad tokenid eksportida json faili ja sealt omakorda importida esirakendusse. Kui selline töökorraldus on valmis seatud, võib disainisüsteemis muuta tokenite väärtusi ja saades sellele kooskõlastus, eksportida need esirakenduse koodi, misjärel rakenduse visuaal muutub ilma, et esirakenduse arendaja omalt poolt midagi teeks. See muudabki arenduse kiiremaks.

```
„link-hover“: {  
  „value“: „{primary.600}“,  
  „type“: „color“,  
  „description“: „This is for link hover state“  
}
```

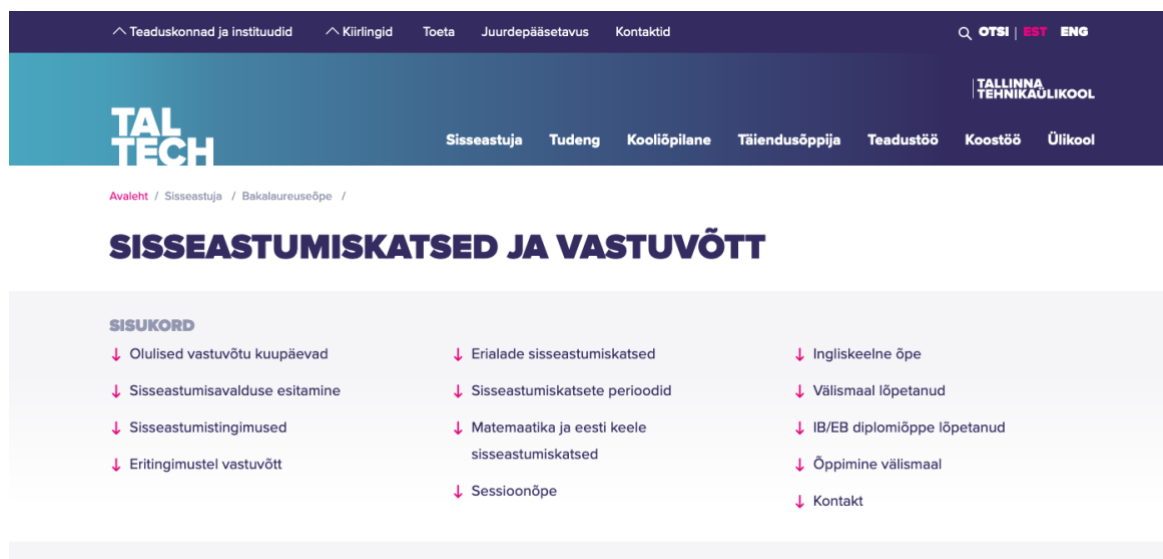
Koodinäide 1. Json failis asuva tokeni näidis. See token hoiab lingi „hover“ oleku värviinformatsiooni ja kasutab omakorda teises tokenis deklareeritud tokeni – primary.600 – väärtust.

### 3.3. Disaini visuaalse osa põhimõtted

Nagu küsitlustest selgus, peab lõppkasutaja ajakohast ja trende järgivat visuaali küllaltki oluliseks. Trendide järgimiseks leidsin Google otsingu abil märksõnadega „*design trends for webpages 2023*“ valiku soovitusi ja samuti vaatasin veebilehete awwwards.com ja dribbble.com trende ning koostasid endale nõu ideeseina näidistega (lisa 4).

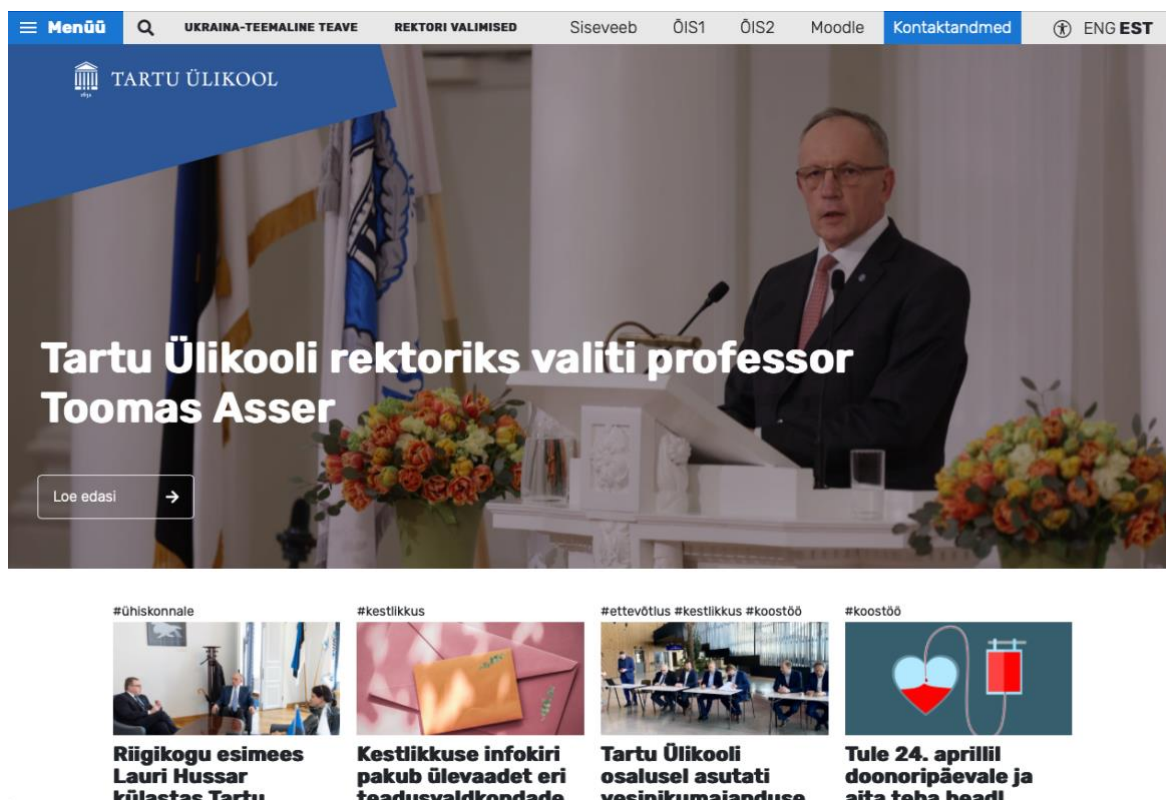
Lisaks trendi järgivatele lehtedele noppisin ideeseinale ka Eesti ülikoolide veebide ja alamlehekülgede näidised ning tuntumate koolituslehtede nagu Udemy, Skillshare, Pluralsight jms veebilehete näidised, et leida, kuivõrd akadeemilised ja kuivõrd muud koolituslehed lubavad endale trendide järgimist.

Esialgne ootus oli, et Eesti ülikoolide lehed on konservatiivsed, et jätta nõu akadeemilisemat muljet. Ent see pigem ei vasta tõele. On õige, et ülikoolide veebilehed ei hakka silma oma ekstravagantsusega, ent rõhutatud konservatiivsust sealt ka silma ei paista. Pigem vormib ülikoolide lehti hoopis samadest CSS-raamistikest tulenev üheülbalisus. Kui koodi uurida, võib järeldada, et näiteks nii Taltech'i kui ka Tartu ülikooli pealehed on mõlemad tehtud kasutades Bootstrap CSS-raamistikku. Bootstrap on Github Stars<sup>23</sup> programmi põhjal populaarseim CSS-raamistik (Silkals, 2023), millest tuleneb aga ka sarnasus oma eeldefineeritud komponentide tõttu ja seeläbi ka igavus, mis võib mõjuda konservatiivselt (joonis 4 ja joonis 5).



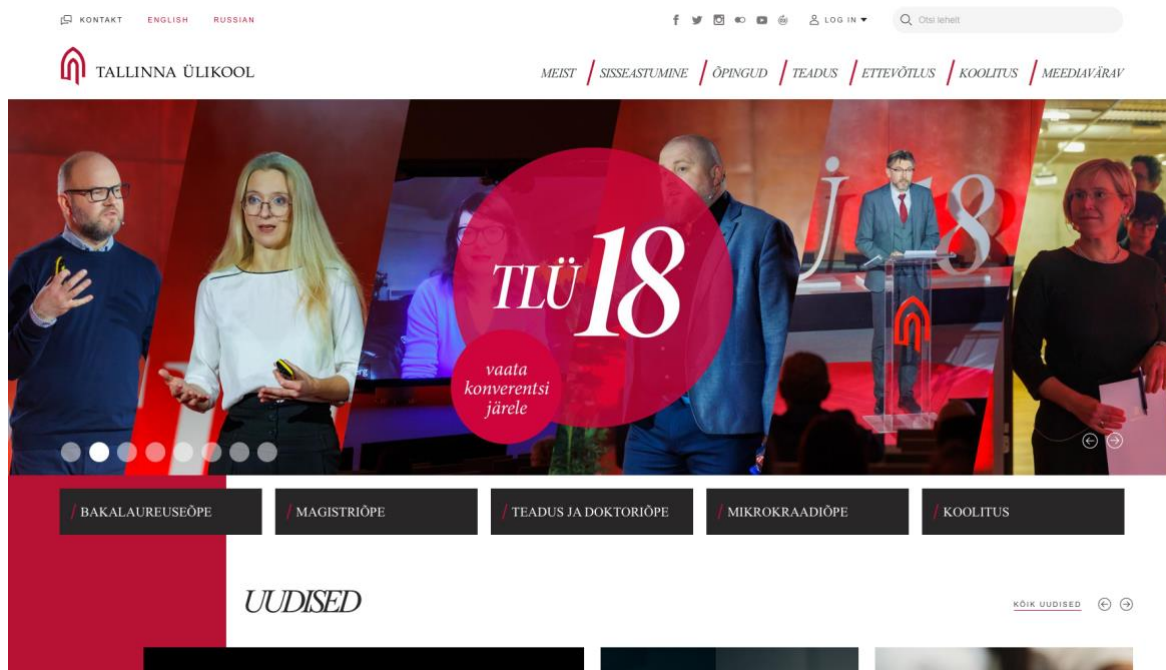
Joonis 4. Taltech veebileht. Kuvatõmmis.

<sup>23</sup> <https://stars.github.com/program/>



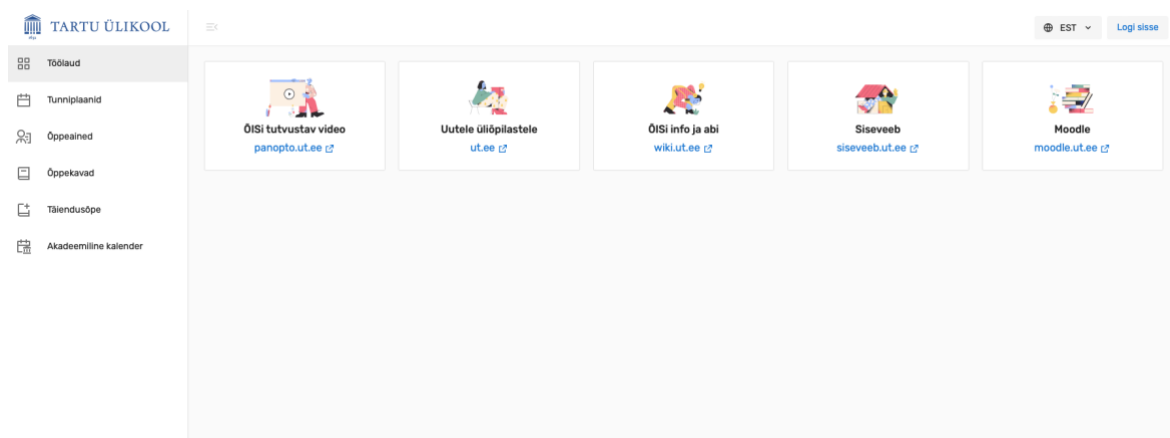
Joonis 5. Tartu ülikooli veebileht. Kuvatõmmis.

Siinjuures võib subjektiivse vahemärkusena lisada, et Tallinna ülikool on eelnevast kahest oma visuaali poolest märksa vabam ja julgem (joonis 6).



Joonis 6. Tallinna ülikooli veebileht. Kuvatõmmis.

Tartu ülikooli puhul aga, kui liikuda edasi alamlehtedele, muutub kujundusstiil juba üsnagi „vabaks“ ja võimalikust akadeemilisest käekirjast pole enam suurt midagi järel (joonis 7).



Joonis 7. Tartu ülikooli õppeinfosüsteem. Kasutusel on hetkel moes olevad joonistused. Kuvatõmmis.

Eelnevast järeldasin, et kuigi tegemist on akadeemilise rakendusega ja peaks säilitama teatud konservatiivsuse, võib hea maitse piires lubada kujundusse üksjagu julgust.

### 3.3.1. Kujunduse stiili valik

Kujunduse soovisin luua suure ja selge tüpograafiaga ja eristuva tekstihierarhiaga, et eri taseme tekstiosad teineteisest hästi eristuks ja sõnumid oleksid kiirelt haaratavad. Sobivaks tüpograafiliseks skaalaks, ehk iga tekstitaseme vaheks valisin pärast testimist kordaja 1.25, ehk *Major third*. Selline skaaladeks jaotamine, mis tagab ühtlase ja harmoonilise struktuuri kogu kujunduses, pärineb Šveitsi tüpograaf Josef Müller-Brockmanilt, kes lõi tüpograafiaskaalad muusikaliste intervallide eeskujul ja avaldas need esmakordselt aastal 1981 oma raamatus „Grid Systems in Graphic Design“ (Müller-Brockmann, 1981). Seega Eesti keeles peaks nimetama skaalat *Major third* Suureks tertsiiks. Tüpograafiaskaala tööriistad on leitavad internetist<sup>24</sup>.

Baasteksti suuruseks valisin parema lugemiskogemuse saamiseks veebilehitsejate vaikimisi suurusest (16px) veidi suurema, ehk 18px.

<sup>24</sup> <https://typescale.com>

Sisuartiklite laiuseks määrasin 60 tähemärki. Loetavuse seisukohalt peavad erinevad uurijad optimaalseks veeru laiust, mis jääb vahemikku 50–75 tm. Kui veerg, ehk rea laius läheb laiemaks, muutub lugejal uue rea alguse leidmine rakseks, lühema rea phul peab lugeja liiga sageli uue rea algusesse hüppama, mis lõhub lugemise rütmi (Scott, 2022).

Kirjatüüpideks valisin peale katsetamist esmalt Roboto Serif H1 ja H2 taseme ning suuremate pealkirjade esitamiseks ning Roboto madalama taseme pealkirjade ja põhikirjatüübi jaoks, ent hiljem vahetasin Roboto välja Montserrat vastu, mis on mahult sarnane Robotole (joonis 8). Roboto šerifi ja Montserrat panin paari nende sarnaste omaduste tõttu(joonis 9).

## **Roboto serif Montserrat**

Joonis 8. Roboto serif'i ja Montserat võrdlus

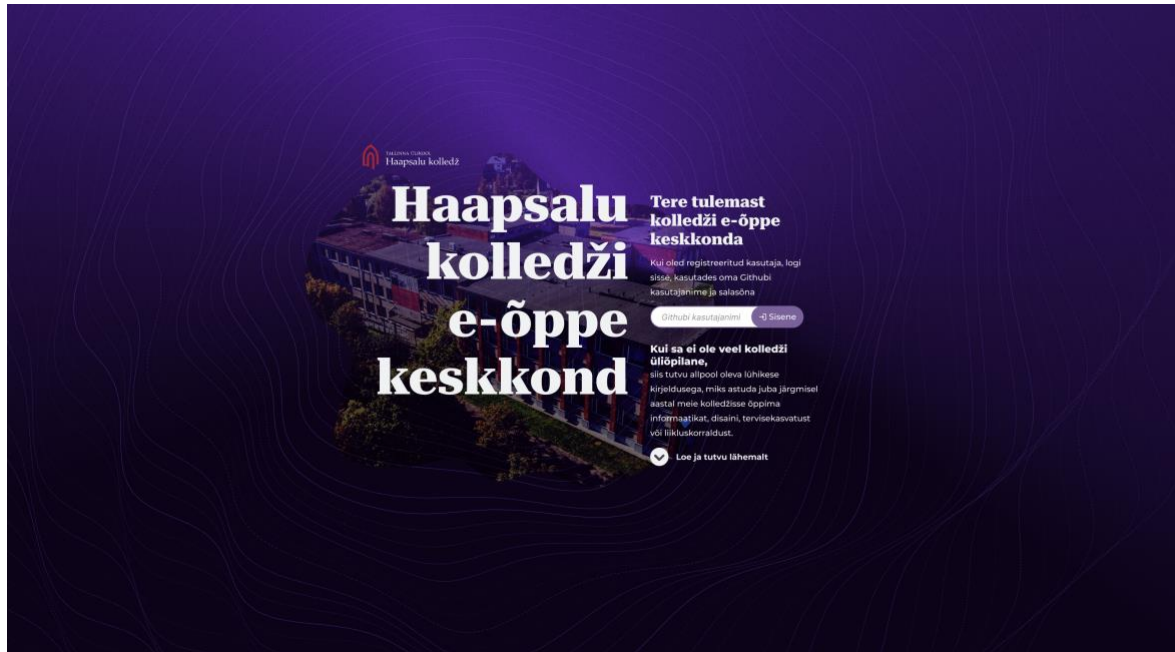


Joonis 9. Kirjatüüpide valik. Autori joonis.

Kuna küsitlustes selgus ka tumeda ja heleda režiimi valiku olulisus, on rakenduses kasutusel heleda ja tumeda vahel valimise võimalus.

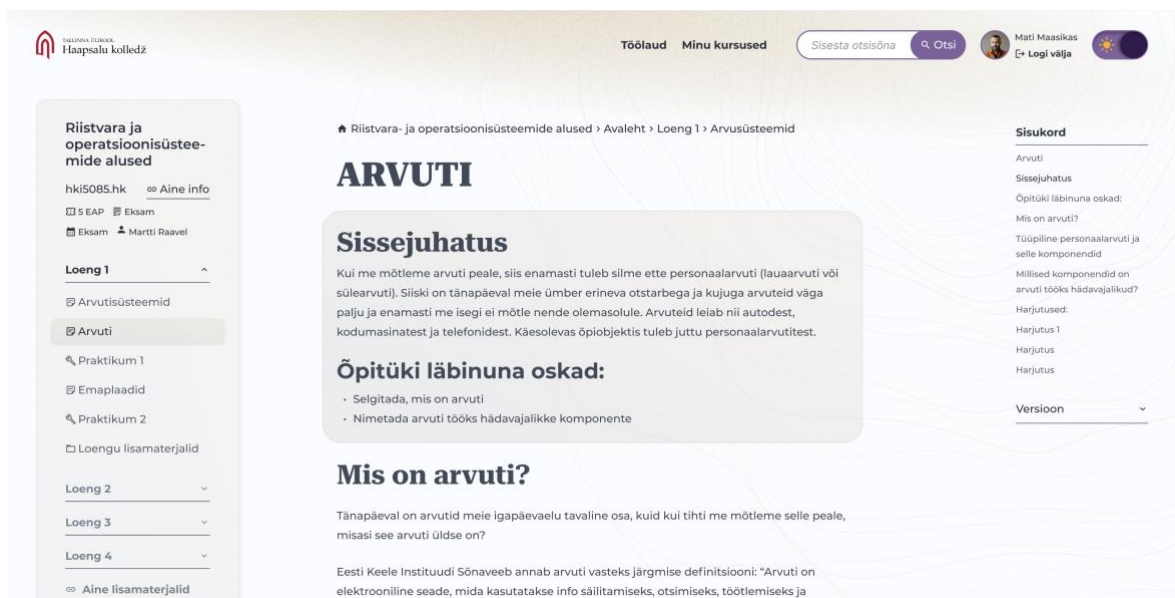
Värvivalikutest võtsin kasutusele sügavsinised/lillakad tumedad toonid, kombineerides need lilla aktsentvärviga, mis on kasutusel nuppude, joonte jms detailides. Subjektiivselt tundus valitud värvikombinatsioon sobituma hästi Tallinna ülikooli logopunasega (joonis 10).





Joonis 10. Rakenduse esileht. Autori joonis.

Sisulehtedel jätkub tumedas režiimis sama tumesinine/lilla ja heledas teemas üliõrna beeži lisatooniga sinakasvalge (joonis 11).



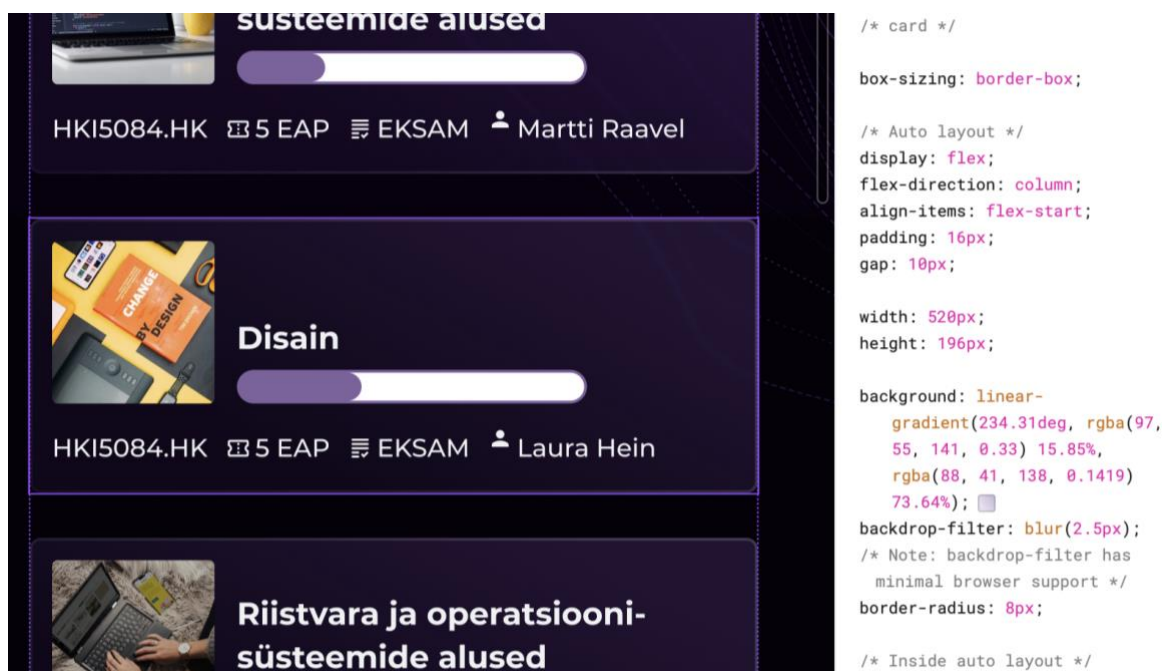
Joonis 11. Heleda teema sisulehe näidis. Autori joonis.

### 3.4. Disainifaili loomise põhimõtted. Disainiprogramm.

Disainifail on kättesaadav lingil: <https://tinyurl.com/e-platvorm-figma>

Disaini loomiseks kasutasin kujundus- ja prototüüpimistarkvara Figma<sup>25</sup>. Tegemist on hetkel maailmas enimkasutatava (Uxtools.co, 2022) veebirakenduste prototüüpimise tarkvaraga ja võib seega loota, et loodud disainifail on veel pikka aega kasutatav.

Nüüdseks, pärast arendus- ja disainiprotsessi läbitemmist võin öelda, et üks keerulisematest osadest disainitookenitega arendamisel on esirakenduse arenduse põhimõtteid ja tehnilisi tingimusi silmas pidav töökorraldus ka disainis – disainifail peab järgima samu printsiipe, mida kasutatakse esirakenduses. See eeldab, et disainer on kursis esirakenduse põhimõtetega ja mõistab, mis on muutujad, mis on veerised ja puhvrid, teksti- ja värvistiilid ja muud esirakenduses olevad osad. Selline nõue ei tarvitse olla kõikides ettevõtetes disainerile jõukohane. Kujundusprogrammides on võimalik luua pilt, millel ei ole koodiga mitte midagi ühist ja mille järgi esirakenduse arendaja peab iseseisvalt koodi kirjutama. Teine võimalus on, et disainer loob kujunduse ja esirakenduse arendaja loeb prototüüpimistarkvara poolt võimaldatud kuvadest vajalikud parameetrid välja ning kirjutab/kopeerib need koodi (joonis 12). Ka sel juhul ei pea disainer süüvima koodi nüanssidesse.



Joonis 12. Figma valitud objekti esirakenduse jaoks sobilikud parameetrid. Autori joonis.

<sup>25</sup> <https://www.figma.com>



Eelnevast tekib ka valikumoment, kas lasta disaineril ja esirakenduse arendajal teha kummalgi oma tööd teineteisega tugevamalt sidumata või püüda disaineri ja esirakenduse töökorraldused ühtlustada.

Diplomitöö saadud kogemuse najal võin pakkuda välja mõned mõtted, millal võtta kasutusele disaini ja koodi sünkroniseeritud töökorraldus:

- üldine kasutajakogemusest lähtuv loogika on paigas;
- soov on kasutada olemasolevat disainiraamistust ja muuta kujundusdetalle selle sees;
- luua suurem disainisüsteem, mida erinevad ettevõtted saavad oma ettevõtte stiili järgi kohendada;
- on olemas kompetents.

Ehk projekti läbiviijad peavad arvestama, et töökorraldus ei muutuks mõttetult keeruliseks antud projekti kontekstis. Kui on teada, et igakordsed muudatused on kui rätsepatöö, pole mõtet aega sedalaadi töökorralduse peale kulutada.

Haapsalu kolledži e-õppe platvorm on vahend, mida üliõpilased saavad õppe eesmärkidel tulevikus täiendada, seega selleks sobib selline töökorraldus hästi – siinse töö skoobist välja jäänud disaini esteetilisest poolt saavad järgmised üliõpilased lihtsasti, esialgu vaid Figma faili töödeldes muuta, parandada või täiendada ja disaini kooskõlastuse saamisel saavad selle automaatselt koodiks eksportida, ilma et esirakenduses väga põhjalikult tööd peaks tegema.

### 3.4.1 Märkused spetsiifilisemate töökorralduslike muudatuste kohta

Vaikimisi on Figma *auto-layout*<sup>26</sup> kasutuselevõtuga muudetud jätkuvalt ühtlase disaini rakendamine väga lihtsaks. Nii näiteks on erinevatest lõikudest koosneva ploki loomine väga lihtne, määrates *auto-layout*'iga lõikude vahed sama väärtusega. Ent koodi viimise puhul määrab Figma selle *gap* deklaratsioonina, mis kehtib vaid *display* deklaratsiooni *flex* ja *grid* väärtuste puhul. Koodis tahtsin ma aga kasutada deklaratsioone *250biil-top* ja/või *250biil-bottom*. Seetõttu muutsin töökorraldust selliselt, et lõikude või plokkide vahede määratlemisel lõpetasin *auto-layout*'i vahede kasutamise. Selle asemel võtsin kasutusele komponentidepõhise töökorralduse, nii, et elementide vahed määrان hoopis puhvriga, mille lisan elemendile üles või alla (joonis 13).

---

<sup>26</sup> <https://help.figma.com/hc/en-us/sections/13165750874519-Auto-layout>



Joonis 13. Vasakul *auto-layout* ja elementide vahed, paremal pealkirja ja sisuga komponent, milles nii pealkirjal kui sisuplokil all ja ülal puhver

Esimesel juhul on disaineri töö kiirem, kuid selle automaatne koodi viimine on võimatu, teisel juhul on koodi viimine automaatne.

Selliste nüansside arvestamist pole vaja, kui disainer lihtsalt disainib ega arvesta koodiga sünkroniseerimise vajadust.

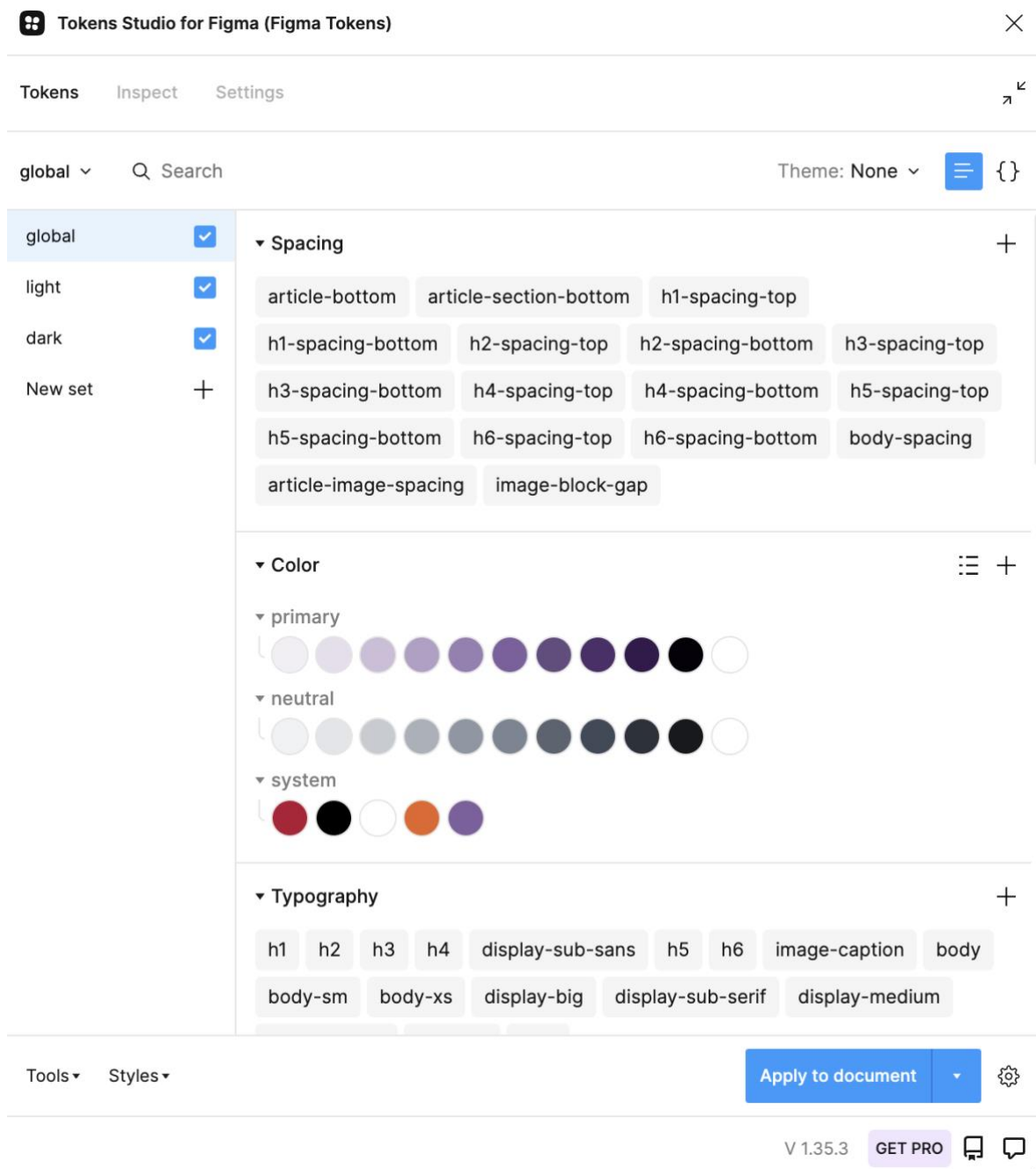
### 3.4.2. Abiprogramm – Figma Tokens Studio

Disainitookeneid vaikimisi genereerida ei saa, selleks on tarvilik mõni Figma pistikprogramm. Mina kasutasin Figma Tokens Studiot (joonis 14).

Arvestama Figma enda eripärade, Tokens Studio eripärade ja esirakenduse spetsiifikaga. Näiteks Tokens Studios saab loodud värvi- ja kirjastiilidest saab genereerida Figma enda värvi ja kirjastiilid, kuid puhvrite ja veeriste väärtusi samal moel luua ei saa. Seega on siin mitmetasandiline töökorraldus – kui oled loonud Tokens Studios tookenid, eksportinud need Figma stiilideks, siis genereeritud stiile saad rakendada otse Figma enda stiilide lehelt, kuid puhvreid ja veeriseid pead määrama objektile Tokens Studio aknas.

Väga hea funktsionaalsus on Figma Tokens Studioli heleda ja tumeda režiimi kasutus. See võimaldab samade tokenite nimede alla salvestada erinevate teemade väärtusi.

Tokens Studio saab ühendada erinevate git'i teenusepakkujatega, mis võimaldab salvestada tokenid näiteks Github'i ja tõmmata need sealt esirakenduse masinasse, mis asub hoopis teisel füüsilisel aadressil.



Joonis 14. Figma Tokens Studio. Kuvatõmmis.

Tokens Studio üks nõrkusi on, et see muudab arvuti töö aeglasemaks. Võib juhtuda, et suuremahuliste tööde puhul jääb arvuti hätta.

Tookenitega arendamisel peab arendustiim kokku leppima nimetamispõhimõtetes, et koodi jõudmisel oleks selge, mille jaoks miski token on. Hea soovitus on tutvuda nt BEM'i<sup>27</sup> või mõne sarnase nimetamissüsteemiga. Kuid kõige lihtsam on lähtuda arusaadavusest – nimeta token selle nimega, mida ta tegema peab. Nt nupu värv võiks olla „button-default“ või „button-hover“, tekstistiili nimetus, kas HTML-elementi nimetus nagu h1 või funktsionaalsuse järgi, näiteks „display-big“.

Kogu disainisüsteemi loomisel tasuks aga töötada välja või kasutada mõnd olemasolevat disainipõhimõtet. Hea soovitus siinjuures on Brad Frosti poolt välja töötatud Atomaarse disaini ideoloogia (Frost, 2013).

### 3.4.3. Värvipaletti koostamine hue, ehk tooni baasil.

Et disaini töövoogu lihtsustada, ehitasin kasutatavad värvipaletid Figma Tokens Studiot kasutades automaatselt varjundite palette koostama. Kui tavapäraselt luuakse värvipaletid käsitsi või kasutades mõnd programmivälist paleti genereerimise tööriista, siis Figma Tokens Studio abil on võimalik muuta töökorraldus automaatseks nõnda, et disainer peab muutma vaid põhitooni, värvivarjundid arvutab aga programm. Värvipaleti eesmärgiks on, et disaineril oleks piisav hulk sama tooni varjundeid (joonis 15). Näiteks kui disainis on vaja eristada interaktiivsete nuppude erinevaid olekuid – *default*, *hover*, *active*, *focus* – on stiililiselt korrektne kasutada selleks sama tooni erinevaid hele-tumedusi. Tavapäraselt koosneb ühe tooni skaala ca kümnest toonierisusest. Nt primaarvärvi astmed on saja astme kaupa primary-100 kuni primary-1000 või kümne astme kaupa primary-10 kuni primary-100. Need järgud, saja või kümne kaupa, ei näita kindlat protsentuaalset toonierisust, vaid lihtsalt järgmist astet. Iga disainisüsteemi looja võib ka oma järgusüsteemi luua ja ka rohkem või vähem tooniastmeid, aga oluline on, et see oleks kõikidele, ka järgmistele arendajatele ja disaineritele arusaadav.

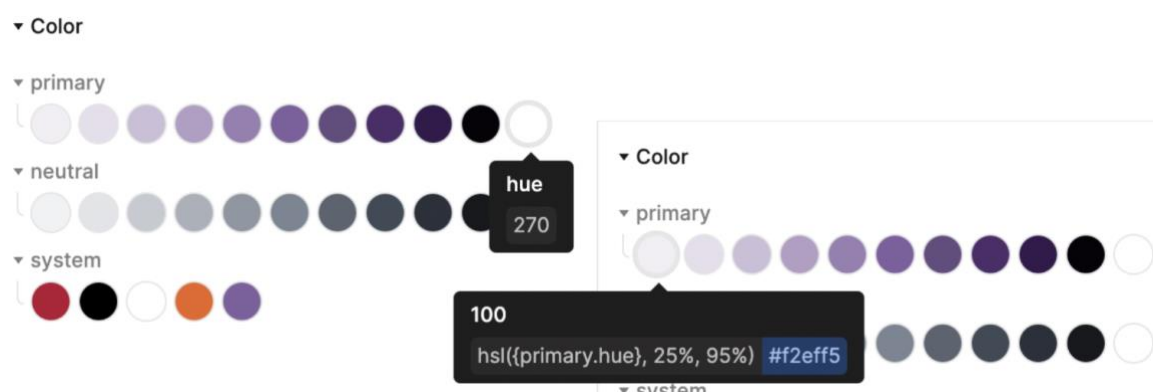
---

<sup>27</sup> <https://getbem.com>

Red	Pink	Purple
500 #F44336	500 #E91E63	500 #9C27B0
50 #FFEBEE	50 #FCE4EC	50 #F3E5F5
100 #FFCDD2	100 #F8BBD0	100 #E1BEE7
200 #EF9A9A	200 #F48FB1	200 #CE93D8
300 #E57373	300 #F06292	300 #BA68C8
400 #EF5350	400 #EC407A	400 #AB47BC
500 #F44336	500 #E91E63	500 #9C27B0
600 #E53935	600 #D81B60	600 #8E24AA
700 #D32F2F	700 #C2185B	700 #7B1FA2
800 #C62828	800 #AD1457	800 #6A1B9A
900 #B71C1C	900 #880E4F	900 #4A148C
A100 #FF8A80	A100 #FF80AB	A100 #EA80FC
A200 #FF5252	A200 #FF4081	A200 #E040FB
A400 #FF1744	A400 #F50057	A400 #D500F9

Joonis 15. Näide Material Design<sup>28</sup> värvipaleti osast. Nagu näha, kasutab Material Design rohkemat värvivarjundite skaalat. Kuvatõmmis.

Figma Tokens Studiot kasutades saab algselt defineerida varjundid tokenitena, mis sisaldavad valemit, mis arvutavad ise HSL<sup>29</sup> värvikoodi kokku. Minu töös on kasutusel 10-astmelised värviskaalad primary ja 29obiile<sup>29</sup>. Figma Tokens Studios on defineeritud toon (hue) ja 10 tooniastet, mis sisaldavad HSL koodivalemit ({toon muutuja}, tooni intensiivsuse protsent, heleduse protsent) (joonis 16).



Joonis 16. Figma Tokens Studios kasutatud värvide arvutamise süsteem. Sellisel skaalal on vaja muuta vaid tooni (hue) ja ülejäänud skaala muudab end vastavalt. Autori joonis.

<sup>28</sup> <https://m3.material.io>

<sup>29</sup> <https://www.w3.org/wiki/CSS/Properties/color/HSL>

### 3.5. Disainifailist koodi

Valikpraktika ajal tehtud rakenduse puhul sain õppetunni, et kui kohe disainist koodi asju viima hakata, võib tekkida üksjagu segadust kui disainifail ei lahenda ära kõiki võimalikke vaateid. Samuti võib juhtuda, et töö käigus on vaja muuta komponentide defineerimise põhimõtteid või asukohti. Selle haldamiseks tegin diplomitöö raames enne päris rakendusega ühendamist eraldi disainisüsteemi HTML-lehe. Ühest küljest on/oli tegemist testkeskkonnaga, kuid teisest küljest on tegu ka komponentide laoga, kust saab valmis kujundatud komponendid kopeerida päris rakendusse. Viimane muudab esirakendust luues töö korduvate komponentidega mugavamaks, kuna käsitsi kirjutamist on vähem.

Kopeeritavate näidiskomponentide lehekülg:

<https://tinyurl.com/tlu-design-system>

Repositooriumi aadress:

<https://tinyurl.com/tlu-design-system-git>

#### 3.5.1 Esirakenduses kasutatud tehnoloogiad

Valikpraktika ajal sai koos tellijatega ehk õppejõududega paika põhimõte, et rakendus võiks olla võimalikult lihtne. Kus võimalik, seal tuleks kasutada olemasolevaid komponente, raamistikke vms lahendusi, mis võiks protsessi haldamist lihtsama ja selgema hoida. Selguse saamine selles, kas selline põhimõte ka tegelikkuses protsessi lihtsamana hoiab, oli samas üks valikpraktika eesmärkidest. Sel põhjusel valisime valikpraktikas mallismootoriks Handlebars.js'i. Põhjus, miks üldse valida mallismootor on esirakenduse seisukohalt see, et hoida kokku aega korduvate elementide loomise pealt. Nt ei pea sel juhul igakordselt kopeerima päise või jaluse vms osa. Valikpraktika raames sai muuhulgas selgeks, et tulevikus oleks rakendus kiiruse huvides mõistlik teha ümber mõnele SPA<sup>30</sup>-le. Nt Reactile, mida Haapsalu kolledžis ka õpetama asutakse. Esialgu plaanisime seda lahendada ka siinse diplomitööga, kuid otsustasime selle plaani kõrvale lükata kuna ülesannete maht oleks liiga suureks läinud.

---

<sup>30</sup> <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

CSS-raamistiku valiku eelis võrreldes puhta CSS-iga on aja kokkuhoid, tänu eeldefineeritud komponentidele või utiliitidele<sup>31</sup> ehk koodiblokkidele. Samuti lähtuvad raamistikud oma kindlast struktuurist ja seeläbi kui näiteks vahetub arendaja, säilib stiil. Teisalt võib osade CSS-raamistike nõrkuseks minu hinnangul pidada ühelaadse stiiliga veebilehtede massilist levikut. Sellel põhjusel välistasime valikpraktika ajal ühe populaarseima CSS-raamistiku – Bootstrap<sup>32</sup> – kasutamise ja võtsime kasutusele Tailwind CSS-i, mis läheneb eeldefineeritud osadele veidi teisel moel. Nimelt ei ole Tailwind CSS-is eeldefineeritud mitte komponendid või objektid, vaid utiliidid ehk koodiblokid. Kui Bootstrap pakub kasutajale suures plaanis valmis lahendust, siis Tailwind CSS hulka eeldefineeritud detaile, millest kasutaja saab ise enda vajaduste järgi lahenduse ja komponendid luua. Kui Bootstrapis on nt nupp ja nupu erinevad variatsioonid valmis kujundatud, siis Tailwindis saab kasutaja, kasutades Tailwindi koodiosi, nupu alles luua (koodinäide 2 ja 3). Bootstrapis saab ka muidugi juba olemasoleva nupu üle kirjutada, ent sel juhul tekib koodi rohkem.

```
<button class="btn btn-primary">Primary Button</button>
```

Koodinäide 2. Nupu koodinäidis Bootstrapi puhul. Autori joonis.

```
<button class="bg-blue-500 hover:bg-blue-600 text-white font-bold  
py-2 px-4 rounded">  
  Primary Button  
</button>
```

Koodinäide 3. Nupu koodinäidis Tailwindi puhul. Autori joonis.

Mõlemal puhul on tulemuseks sarnane sinine nupp:



Joonis 17. Nupp. Kuvatõmmis.

Esmapilgul võib tunduda, et Tailwindi puhul on koodi rohkem. Ent vahe on selles, et Bootstrapi puhul on kood peidus Bootstrapi teekides. Kui nüüd hakata Bootstrapi kujundust oma stiili järgi ümber kujundama, peab lisaks olemasolevale Bootstrapi koodile, lisama oma stiili, ent Tailwindi puhul saab asendada muutujad enda muutujatega. Võib öelda nii, et Tailwindi abil on arendajal võimalik luua nõ oma Bootstrap. Vastupidi ei saa.

<sup>31</sup> <https://designsystem.digital.gov/utilities/>

<sup>32</sup> <https://getbootstrap.com>

Tailwindi saab kasutada nii CDN-ist, Node arenduskeskkonnas üksi, kui ka PostCSS pistikprogrammina. Viimane on kõige sujuvam viis kui on soov kasutada arenduses ka mõnda komplekteerimistööriista (*bundling software*) nagu näiteks Webpack'i<sup>33</sup>, Vite'i<sup>34</sup> või Parcelit<sup>35</sup>, aga ka PostCSS-i<sup>36</sup> ennast, nii nagu siinses diplomitöös kasutatud on.

PostCSS on siinses töös kasutusel koodi minimeerijana, samuti, kasutades autoprefixer pistikut, muudab ta CSS koodi automaatselt kõikidele veebilehitsejatele arusaadavaks.

Tailwindi lihtsamaks kasutamiseks on soovitatav kasutada koodiredigeerimisprogrammide pistikprogramme. Näiteks Visual Studio Code puhul Tailwind CSS IntelliSense't (Joonis 18), mis annab vihjeid kasutada olevate utiliitide, ehk koodiplokkide kohta.



Joonis 18. Tailwind CSS IntelliSense pistikprogramm Visual Studio Code'le. Kuvatõmmis.

### 3.5.2 Disainitokenite koodiks muutmine omal jõul.

Võimalusi, kuidas Figmast eksporditud json esirakenduse stiilimuutujateks lugeda on mitu. Esmalt otsustasin kirjutada ise vajalikud javascripti koodifailid.

Figma Tokens Studio ekspordib Figmast json faili kasutusel olevate disainitokenitega. Võimalik on eksportida lihtsalt hulk tokeneid, kuid on ka võimalik grupeerida tokenid vastavalt funktsioonile, nt üldised ja hele ning tumeda režiimi tokenid (koodinäide 4).

<sup>33</sup> <https://webpack.js.org>

<sup>34</sup> <https://vitejs.dev>

<sup>35</sup> <https://parceljs.org>

<sup>36</sup> <https://postcss.org>

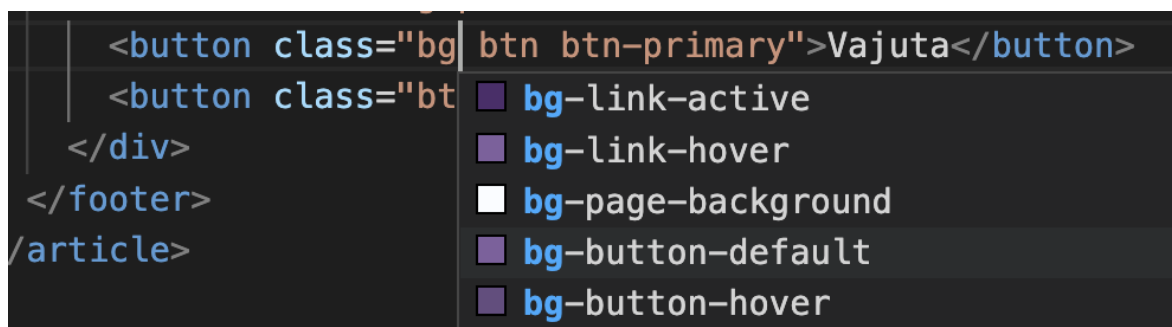


```
{
  „33obiile“: {
    „text-base“: {
      „value“: „1.125rem“,
      „type“: „fontSizes“,
      „description“: „Font sizes from TaylwindCSS“
    },
  },
  „light“: {
    „text“: {
      „value“: #000
      „type“: „color“,
    },
  },
  „dark“: {
    „text“: {
      „value“: #FFF
      „type“: „color“
    },
  },
  „$themes“: [],
  „$metadata“: {
    „tokenSetOrder“: [
      „33obiile“,
      „light“,
      „dark“
    ]
  }
}
```

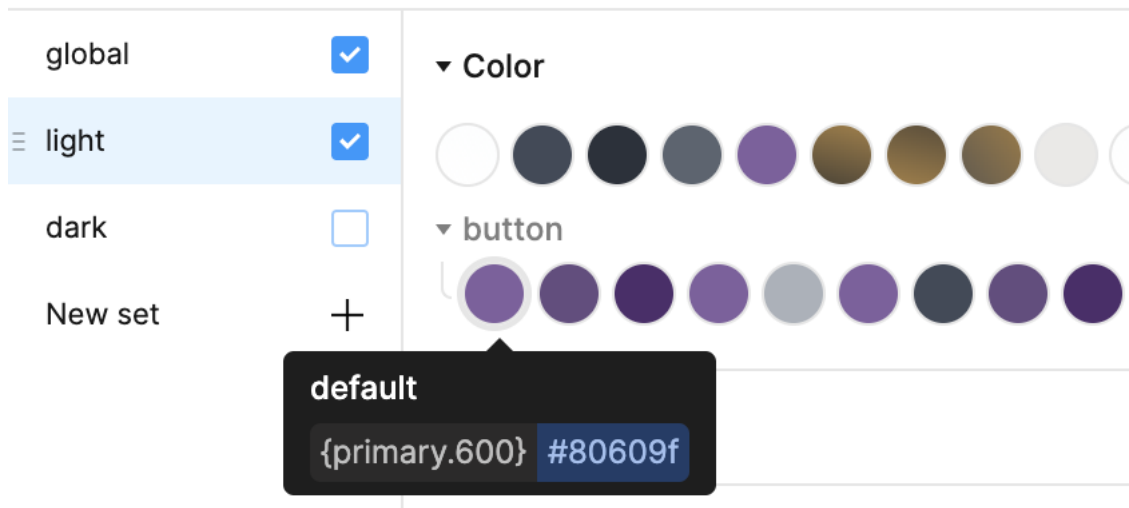
Koodinäide 4. Lihtsustatud tokenite json näide, milles on üldine, hele ja tume teema. Autori joonis.

Selleks, et json failis olevat tokenid muuta Tailwindi jaoks sobivateks muutujateks, on vaja luua skript, mis json failis oleva info läbi töötab ja tailwindi tõmbab.

Soovitud tulemus peab olema Tailwindis kasutatavad, kasutaja poolt loodud koodiblokid (Joonis 19), mis vastavad Figma Tokens Studio algmaterjalile (Joonis 20).



Joonis 19. Kasutaja poolt loodud Tailwindi koodiblokid. Autori joonis.



Joonis 20. Figma Tokens Studio nupu värviinformatsiooni sisaldav token. Autori joonis.

Kuna Figma Tokens Studios loodud värvid sisaldasid muutujaid, tuli skriptifaili luua funktsioonid, mis asendaks muutujate nimed muutujate väärtustega. Kui soovida kasutada värve CSS-muutujatena ning kasutada ka läbipaistvust ja omakorda defineerida need läbi Tailwindi, tuleks värvid `tailwind.config.js` failis defineerida läbi RGB kanalite<sup>37</sup>. Siinses töös ma asja nii keeruliseks ei ajanud, olgugi, et värvide konvertimise HSL-ist RGB-ks ja HEX-iks tegin läbi põhjusel, et lugesin Tailwindi juhendist ekslikult, et ma ei saa koodis HSL värviskeemi kasutada. See oli oma viga, mille tõttu kulutasin väga palju aega tühja töö peale.

Minu poolt kirjutatud skript kujunes poolautomaatseks, see tähendab, et ma kirjutasin javascripti faili, milles defineerisin värvitokenid ja muus osas luges skript json failist seal olevad minu tokenitele vastavad väärtused ning lõpuks luges Tailwindi konfiguratsioonifaili minu javascripti failist tulemused Tailwindi. Selline lahendus jäi poolikuks. Vaja oli skripti, mis teeb automaatselt kõik mis vaja, kuid töömaht hakkas muutuma veidi liiga suureks selles mõttes, et aega muude teha vajavate ülesannete jaoks hakkas nappima. Seetõttu võtsin kasutusele teiste poolt sama eesmärgi täitmiseks mõeldud lahendused.

<sup>37</sup> <https://tailwindcss.com/docs/customizing-colors>

### 3.5.3. Amazon Style Dictionary, Tailwindi töökorraldus

Tuntum disainitookeneid tõlgendav lahendus on ilmselt Amazon Style Dictionary<sup>38</sup>. Style Dictionary võtab json failist tokenite info ja muudab selle kas SCSS<sup>39</sup>, CSS või iOS või Android stiilifailideks. Figma Tokens Studiost eksporditava json faili struktuur on Style Dictionary nõuetest pisut erinev, kuid Tokens Studio loojad on omakorda loonud Node rakenduse, mis teeb vajalikud muudatused, mis võimaldavad Style Dictionarit kasutada<sup>40</sup>.

Kui ainult CSS-i või SCSS-i kasutada, sellest piisaks. Tailwindi puhul on aga hea tuua osad tokenid Tailwindi konfiguratsioonifaili sisse, et kasutada utiliite ehk koodiplokke nagu eelnevalt näidatud joonisel 19. Ka selle jaoks sain lahenduse Tokens Studio loojatelt<sup>41</sup>.

Lõppkokkuvõttes oli mul vajalik lahendus, milles Figmast eksporditakse json fail tokenitega, see konverditakse erinevateks Style Dictionaryle sobivateks json failideks – globaalse tokenitega, heleda režiimi tokenitega ja tumeda režiimi 35obiile35e35a, seejärel luuakse CSS failid muutujatega, mida saab kasutada esirakenduse koodis ja on olemas Tailwindi konfiguratsioonifail, mis loeb vajalikud muutujad Tailwindi süsteemi.

Amazon style Dictionary ja Figma Tokens Studio tööks vajalikud koodifailid asuvad rakenduse kataloogis:

- fns.js
- build.js

Failide sisus on kirjas ja saab määrata, kust võetakse lähteandmed ja kuhu kirjutatakse koodi jaoks sobilik lõpptulemus.

Lisaks lõin koodifaili, mis puhastab Style Dictionary poolt loodud json faili ebaolulisest või mittekasutatavast koodist: updateclean.js

---

<sup>38</sup> <https://amzn.github.io/style-dictionary>

<sup>39</sup> <https://sass-lang.com>

<sup>40</sup> <https://github.com/tokens-studio/figma-plugin/tree/main/token-transformer>

<sup>41</sup> <https://github.com/tokens-studio/plugin-example-tailwindcss>

### 3.5.4 Tailwindi jaoks sobilikus formaadis koodi kasutamise põhimõtted ja kasutamine

Kui sild disainifaili ja esirakendusega oli loodud, tuli otsustada, millised muutujad ja miks peaks olema Tailwindi süsteemis ning kas on vaja muutujaid kasutada ka tavapärasel ehk CSS-muutujate kasutamise teel.

Kuna ka valikpraktika raames kasutasin Tailwindi, sain sealt mõningase kogemuse ja ideed seekordseks töökorralduseks. Tailwindi süsteemi, ehk `tailwind.config.js` faili tõin sisse põhilised stiilid – tekstide suurus, kaalud ning skaala, kirjatüübid ja värvid. Kõiki muutujaid ei pidanud vajalikuks läbi `tailwind.config.js`-i süsteemi tuua, kuna see oleks muutnud `tailwind.config.js` faili pikaks ja segaseks ning osade stiilide sissetoomine läbi CSS-muutujate faili tundus ka lihtsam. Nii näiteks on kirjastiilide parameetrid, mis on seotud vaba ruumiga ümber teksti, toodud sisse läbi CSS-muutujate faili, kasutades `@import` reeglit (koodinäide 5). Koodinäites kuvatud muutujad on deklareeritud `36obiile.css` failis, mis on sinna omakorda toodud Figmast läbi varem kirjeldatud protsessi.

Üldise töökorraldusena lõin stiilifailides komponendid korduvatest elementidest, nagu näiteks nupud – *primary* ja *secondary* (koodinäide 6). Suuremad üksused, nagu näiteks päised, jalused, külgribad jms, stiliseerisin Handlebars'i taaskasutatavaid komponente (*partials*) kasutades (koodinäide 7).

```
@import 'output/36obiile.css';
section {
  36obiil-bottom: var(--article-bottom);
}
body {
  line-height: var(--line-height-body);
  36obiil: var(--body-spacing);
}
h1 {
  line-height: var(--h1-line-height);
  36obiil-top: var(--h1-spacing-top);
  36obiil-bottom: var(--h1-spacing-bottom);
}
```

Koodinäide 5. Muutujate sissetoomine läbi `@import`. Reegli.

```
@layer components {
  .btn {
    @apply h-auto px-6 py-3 rounded-full flex items-center gap-2
    justify-center min-w-[3.25rem] min-h-[3.25rem];
  }
  .btn-primary {
    @apply text-system-white bg-button-default hover:bg-button-
    hover active:bg-button-active focus-visible:outline-system-focus
    focus-visible:border-solid focus-visible:outline focus-
    visible:outline-2 focus-visible:outline-offset-2;
  }
  .btn-secondary {
    @apply text-button-secondary-label-default hover:text-button-
    secondary-hover border-2 border-button-secondary-default
    hover:border-button-secondary-hover active:border-button-
    secondary-active active:text-button-secondary-active focus-
    visible:outline-system-focus focus-visible:border-solid focus-
    visible:outline focus-visible:outline-2 focus-visible:outline-
    offset-2;
  }
}
```

Koodinäide 6. Tailwindis stiliseeritud nupud. Edaspidi ei ole vaja kogu koodi kopeerida, vaid piisab klassist .btn koos .btn-primary või .btn-secondaryga.

```
<nav
  class='nav bg-neutral flex flex-col lg:flex-row justify-between
  text-white p-8 relative z-10'
>
  <div class='flex justify-between items-center'>
    <a
      href='/'
      class='w-40 sm:w-64 h-auto'
    >
      <img
        src='/images/haapsalu-kolledzi-logo-mustal.svg'
        alt='Tallinna Ülikooli logo'
        style='max-height: 100px;'
      />
    </a>
    <span
      onclick='closeMenu()'
      class='material-symbols-outlined text-4xl cursor-pointer
      37obiile37e right-8 lg:hidden text-xxl'>
      menu
    </span>
  </div>
```

Koodinäide 7. Fragment Handlebars'i päise osast.

Kui tulevikus rakendus ümber ehitada React'i/Next'i või mõne teise javascripti raamistikule, soovitaksin sama loogikat, sel juhul siis React'i komponentide stiliseerimisega.

### 3.5.5. Tulemus ja edasiarendus

Diplomitöö raames valmis:

- lõppkasutajate vajadustel, visuaalsetel eelistustel ja hetke trendidel baseeruva visuaaliga disain;
- Figma prototüüpimistarkvaras arendatav disaini arenduskomplekt, ehk UI-kit;
- väiksemahuline disainisüsteem;
- disainifailiga üle githubi ühendatud esirakendus, mis kasutab TailwindiCSS stiiliraamistikku ja Handlebars.js mallimismootorit.

Kuidas täpsemalt toimib Figma disainifaili ja esirakenduse koodi sünkroniseerimine, on nähtav selles demovideos:

<https://tinyurl.com/disainitokenid>

Edaspidises töös peavad järgmised disainerid arvestama, et disaini muutes peab kasutama disainitokeneid, kasutades Figma Token Studio nimelist pistikprogrammi – see võimaldab disaini muundamise koodiks kasutades Amazon Style Dictionary nimelist Node moodulit ja skripte koos Figma Tokens Studio skriptidega.

Testimise käigus on ilmnenu, et Handlebars'i jaoks on selle rakenduse maht juba liiga suur ja tasub kaaluda React.js-i või Next.js peale esirakenduse ümbertegemist, mida selle töö raames ei jõudnud teha.

Rakendus on seadmetundlik ja loomisel on arvestatud WCAG<sup>42</sup> nõuetega.

---

<sup>42</sup> <https://www.w3.org/WAI/standards-guidelines/wcag/>

## KOKKUVÕTE

Haapsalu kolledž tellis e-õppe platvormi, mis valmis siinkirjutaja ja Krister Sepa ühistööna.

Rakendus vastab järgmistele kriteeriumitele:

- Rakendus on võimalik Haapsalu Kolledži poolt kasutusele võtta.
- Rakendus on sobilik järgmiste aastakäikude üliõpilastele edasiarendamiseks koolitööde raames.
- Rakenduse kood on kättesaadav githubis.

Rakenduse koodi githubi aadress on: <https://github.com/tluhk/HK-Ainekursus/tree/>

Rakenduse veebiaadress on: <http://ainekursus.hk.tlu.ee/login>

Disainitootenite tööpõhimõtet tutvustav video: <https://tinyurl.com/disainitootenid>

Rakenduse loomise üheks eesmärgiks oli ehitada disaini ja esirakenduse integreeritud süsteem, mille disaini saab muuta vastavalt vajadusele nii, et see esirakenduse koodi automaatselt, esirakenduse arendaja olulise panuseta üle kanda. Sellise süsteemi eeliseks on aja ja ressursi kokkuhoid. Sellise süsteemi ehitamise teeb võimalikuks disainitootenite kasutamine, millised eksporditakse disainiprogrammist ja muudetakse esirakenduses CSS-i parameetriteks.

Diplomitöös kogunenud teadmiste baasil võib disaini ja esirakenduse koodi sünkroniseerimise kohta väita järgmist:

- on kasulik, kui disainer on kursis HTML-I ja CSS-ga;
- disainer peab kasutama stiile ja disainitooteneid läbi Figma Tokens Studio või analoogse pistikprogrammi;
- esirakenduses peab kasutama Amazon Style Dictionary't või analoogset rakendust;
- sellise süsteemi loomine ei tasu ära väikeste projektide puhul – algse raamistiku paikapanek võtab rohkem aega kui rakenduse valmistamine ilma sünkroniseerimiseta;
- selline süsteem on väga hea disainisüsteemi loomiseks, mida saab kasutada erinevate projektide või klientide rakenduste loomiseks, nõ korduva mustri puhul;
- selline süsteem sobib hästi ülikooli enda rakenduse arenduseks, mida üliõpilased ise edasi arendavad.

## SUMMARY

Creation of the front-end application of Haapsalu College's e-learning platform with synchronization of design and front-end application code.

Haapsalu College commissioned an e-learning platform, which was created as a joint effort between the author and Krister Sepp.

The application meets the following criteria:

- the application can be adopted by Haapsalu College;
- the application is suitable for further development by students of the following cohorts as part of their coursework;
- the application code is available on GitHub.

Address of code repository: <https://github.com/tluhk/HK-Ainekursus/tree/>

Application web address: <http://ainekursus.hk.tlu.ee/login>

Demovideo about design tokens: <https://tinyurl.com/disainitookenid>

One of the goals of creating the application was to build an integrated system of design and frontend, where the design can be changed as needed without requiring the frontend developer to make significant contributions. The advantage of such a system is saving time and resources. The use of design tokens makes such a system possible, which are exported from a design program and converted into CSS parameters in the frontend.

Based on the knowledge gathered in the diploma thesis regarding the synchronization of design and frontend code, the following can be stated:

- it is useful for the designer to have knowledge of HTML and CSS;
- the designer must use style and design tokens, in this case, through the Figma Tokens Studio plug-in;
- the frontend must use Amazon Style Dictionary or a similar application;
- creating such a system is not cost-effective for small projects – setting up the initial framework takes more time than creating the application without synchronization;
- such a system is excellent for creating a design system that can be used for various projects or client applications, especially for recurring patterns;
- such a system is well-suited for developing the University's own application, which students can further develop themselves.



## KASUTATUD ALLIKAD

- Adhuham, M. (2021). A Complete Guide to Dark Mode on the Web. [2023, aprill 28].  
<https://css-tricks.com/a-complete-guide-to-dark-mode-on-the-web>
- Alam, R. (2022). Utility-first CSS framework? – Tailwind CSS. [2023, aprill 28].  
<https://xenox.dev/tailwind-css-utility-first-css-framework/>
- Bolton, J. Ferrua, B. Introducing The Salesforce Lightning Design System. [2023, märts29].  
<https://www.salesforce.com/video/192950/>
- Carlin, E. (2020). The Theory: A Semantic Color System. [2023, veebruar 14].  
<https://dev.to/ynab/a-semantic-color-system-the-theory-hk7>
- Chernev, B. (2023). 29 Astonishing E-learning Statistics for 2023. [2023, aprill 14].  
<https://techjury.net/blog/elearning-statistics/>
- Design Tokens Community Group (2023). Design Tokens Format Module. [2023, mai 19].  
<https://design-tokens.github.io/community-group/format/#design-token>
- Frost, B. (2013). Atomic design. [2023, veebruar 14].  
<https://atomicdesign.bradfrost.com>
- Google Search Central. (2023). Mobile site and mobile-first indexing best practices. [2023, aprill 20].  
<https://developers.google.com/search/docs/crawling-indexing/mobile/mobile-sites-mobile-first-indexing>
- Interaction Design Foundation. Design Systems. [2023, mai 19].  
<https://www.interaction-design.org/literature/topics/design-systems>
- InVisionApp. UI kit. [2023, mai 19].  
<https://www.invisionapp.com/defined/ui-kit>
- Nielsen Norman Group. The Definition of User Experience (UX). [2023, mai 21].  
<https://www.nngroup.com/articles/definition-user-experience>
- Lemonaki, D. (2022). Frontend VS Backend – What's the Difference? [2023, mai 20].  
<https://www.freecodecamp.org/news/frontend-vs-backend-whats-the-difference>
- Lindgaard, G. Fernandes, G. Dudek, C. Brown, J. (2011). Attention web designers: You have 50 milliseconds to make a good first impression! [2023, märts 25].  
<https://doi.org/10.1080/01449290500330448>
- Mangialardi, M. (2021). Integrating Design Tokens With Tailwind. [2023, veebruar 5].  
<https://www.michaelmang.dev/blog/integrating-design-tokens-with-tailwind>

- Moran, K. (2017). The Aesthetic-Usability Effect. [2023, märts 25].  
<https://www.nngroup.com/articles/aesthetic-usability-effect>
- Muldoon, K. (2022). How to name your colors in a Design System. [2023, veebruar 14].  
<https://uxdesign.cc/how-should-you-name-your-colors-in-a-design-system-3086513476df>
- Müller-Brockmann, J. (1981). *Grid systems in graphic design. A visual communication manual for graphic designers, typographers and three dimensional designers.* Niggli.
- Rungta, K. (2023). 100+ Must Know Online Learning Statistics in 2023. [2023, aprill 25].  
<https://www.guru99.com/online-learning-statistics.html>
- Scott, E. (2022). Readability: The Optimal Line Length. [2023, veebruar 14].  
<https://baymard.com/blog/line-length-readability>
- Silkalns, A. (2023). 12 Best Open-Source CSS Frameworks 2023. [2023, mai 20].  
<https://www.freecodecamp.org/news/frontend-vs-backend-whats-the-difference>
- Statcounter. (2023). Desktop vs Mobile vs Tablet Market Share Worldwide. [2023, aprill 25]  
<https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide>
- Swiss School of Business and Management Geneva (2022). Rise of E-Learning During the COVID-19 Pandemic. [2023, aprill 25].  
[https://www.ssbm.ch/blog/category/blog\\_mainportal/](https://www.ssbm.ch/blog/category/blog_mainportal/)
- Thalion. (2021). Building Design System: Part 3 Colors. [2023, veebruar 14].  
<https://uxmisfit.com/2021/07/12/building-design-system-part-3-colors/>
- Tromp, T. (2017). Kui seinad jutustavad. [2023, aprill 2].  
<https://blog.twn.ee/et/Kui-seinad-jutustavad>
- U.S. Web Design System (USWDS). Utilities. [2023, aprill 29].  
<https://designsystem.digital.gov/utilities/>
- Uxtools.co (2022), 2022 Design Tools Survey. [2023, mai 20].  
<https://uxtools.co/survey/2022/basic-prototyping/#basic-prototyping-tools-graph>
- Vallaste, H. (s.a.). E-teatmik. <http://www.vallaste.ee>
- World Economic Forum (2020). The COVID-19 pandemic has changed education forever. This is how. [2023, aprill 25].  
<https://www.weforum.org/agenda/2020/04/coronavirus-education-global-covid19-online-digital-learning/>

## LISA 1. POOLSTRUKTUEERITUD INTERVJU LÕPPKASUTAJAGA

### E-kursus

Üliõpilase töölaua vajadused

#### Sissejuhatus.

Järgnev uuring on Haapsalu kolledži lõputööks valmiva kolledži e-kursuse kasutajakogemuse uuring, mille abil selgitada välja üliõpilaste eelistused õppeplatvormi osas – kuidas tahab üliõpilane e-kursuseid kasutada, millised funktsioonid on e-kursuse töölaual/avakuval olulised, millised on üliõpilase nõudmised nii disainile kui ka üldisele väljanägemisele.

- Mida süsteem sulle kui õppijale ütlema peaks/võiks, millist abi osutama?
- E-kursusel on töölaud/avakuva – milliseid abistavaid viitse(“feature”), peaks sellel kuvama?

Kui ta ei oska vastata, siis alljärgnevalt abistavad küsimused.

- peab olema / oleks lahe / ei ole vaja
- õppejõudude nimekiri, kiired kontaktid
- abi
- õpingu edenemisriba (milline? %?) näiteid
- kas peaks nägema ühe aine järgmise tuleva teema nime?
- kas peaks nägema aine EAPsid? hindamisvormi?
- Mis on tehtud, mis tulemas, ehk, et mis järgmiseks korraks teha vaja
- viimati külastatud/õpitud aine
- järjestamine – kõige olulisem kõige ees?
- kas peaks nägema iseenda statistikat? viimati kursuste külastamise aega?
- kas peaks nägema avalehel enda kursuste viimaseid uuendusi õpetajate poolt? Või oleks eraldi leht, kus näeb neid, a’la notifications leht
- kas oleks oluline oma õpiaja graafiku nägemine (kaua ma lehel viibisin/õppisin)
- kas grupikaaslaste nimekirja nägemine on oluline, et githubi kontakte näha?
- Kas on oluline, et näed, et keegi grupikaaslane on sisse loginud:
- mitu õpilast on sisse loginud rohelise mummuga

- mitu õpilast vaatab praegu sinuga sama kursust
- anonüümselt: millal esitati viimati mingi kodutöö
- Kui githubis on projektigrupid, siis näha gruppide koosseise?

Kui eelmisest vastused käes, võiks küsida, kas ta teab häid näiteid a'la milline progressiriba on hea? Kus peaks abikontakt asuma?

- Mitme keskkonna progressi näited, mis mõtted tekivad?
- Mitme edenemise näited, milline on arusaadavam?
- Mitme töölaua näited, milline on selgem ja arusaadavam

Kasutuskogemuse definitsioon – siin soovime teada just seda osa, kui intuitiivne ja arusaadav on ühe ainekursuse õpiteekond.

- Milliseid e-õppe keskkondi tead? Nimeta.
- Milliseid kasutad?
- Millised neist on kasutuskogemuse poolest kõige paremad?
- Mis on see, mis kasutuskogemuse neis kõige paremaks teeb?
- Milline neist keskkondadest nägi välja kõige kenam, ilusam, moodsam? (kui pole kasutanud, võiks näidata stiile, millele hinnangut anda)
- Kui oluline see on?
- Kuivõrd sa jäta kasutamata keskkonna, mis on väljanägemiselt halb, kas on üldse nii halva väljanägemisega veebikeskkondi, mis hakkavad kasutamist segama?
- Kui oluline on tume teema vs hele teema (dark vs light mode)?
- Millised funktsioonid veebilehel lihtsustavad õppimist, muudavad õppimise mugavamaks?

Kõige lõpus Valikpraktika raames tehtud rakenduse etteandmine ja seal ülesande täitmine – liigu Riistvara kursusele ja leia loeng Arvutid.

- Kommenteeri, mis on hea, mis on halb, mis on loogiline, mis on ebaloogiline. Mida soovitad

## LISA 2. KASUTATUD E-ÕPPEKESKKONDADE JA DISAINIEELISTUSTE KÜSITLUS

1. Mitmenda kursuse tudeng sa oled?

- I
- II
- III
- Muu...

2. Millist eriala õpid?

3. Palun ütle ka, mis soost sa oled?

- Mees
- Naine
- Muu...

4. Ja kui vana?

Alla 20 (kaasaarvatud)

- 21-30
- 31-40
- 41-50
- 51-60
- 61-70
- 71-...

5. Milliseid e-õppe keskkondi kasutad või oled kasutanud?

6. Millistes e-õppe keskkondades on õppejõud oma materjale esitanud või õppetööd läbi viinud?

7. Millist ühte kasutatutest võid esile tõsta? (hakkas teistega võrreldes silma või kasutasid rohkem või oled vaid seda kasutanud)

8. Kirjelda vabas vormis, mis on see, mis eelmises punktis nimetatu heaks teeb või mis esiletõstmist väärrib?

9. Kuidas hindad eenevalt esile toodud keskkonna kujundust?

- Ei ole sedapidi asjale tähelepanu pööranud
- 1
- 2
- 3
- 4
- 5
- Kujundus oli üks olulistest komponentidest, mis selle keskkonna väga heaks tegi

10. Tekst (liigendatus) oli

- nagu oli, ei oska öelda
- 1
- 2
- 3
- 4
- 5
- väga hästi liigendatud, kerge oli lugeda

11. Tekst (kontrastsus) oli

- tavaline
- 1
- 2
- 3
- 4
- 5
- kontrastne, lihtsustas lugemist

12. Teksti suurus oli

- Normaalne

- Väike
- Tavalisest suurem, hakkas silma
- Muu...

13. Oskad sa kirjeldada, mis on see, mis muudab teksti hästi loetavaks?

14. Kirjatüüp ehk font oli

- tavaline, ehk ei pannud midagi erilist tähele
- seriifidega ja see on hea
- seriifideta ja see on hea
- õrnalt erikujuline, mitte tavaline Times või Arial, aga mitte ka ülepingutatud – otsitud
- Muu:

15. Pealkirjad olid

- suured ja rasvased
- no nagu ikka

16. Värvid olid

- mahedad
- kontrastsed
- Muu

17. Üldine stiil oli

- pigem selline ümar, pehme
- nurgeline
- Muu

18. Kui oluline on "dark mode" (hele tekst tumedal taustal)

- ei ole mulle seda vaja
- 1

- 2
- 3
- 4
- 5
- väga oluline

19. Kas eelpool nimetatud punktid liigituvad sinu jaoks kujunduse alla ja räägivad sellest, miks see kujundus oli hea või halb (vastavalt sellele kuidas sa vastasid küsimusele nr. 5)?

20. Kas on veel mingid komponendid, mida sa pead kujunduse juures oluliseks?

21. Kui oluline on e-õppe võimalus mobiilis?

- ei tunne puudust
- 1
- 2
- 3
- 4
- 5
- ülioluline, õpin kooli sõites

22. Kas e-õppe veebileht peab olema moodne, jälgima trende?

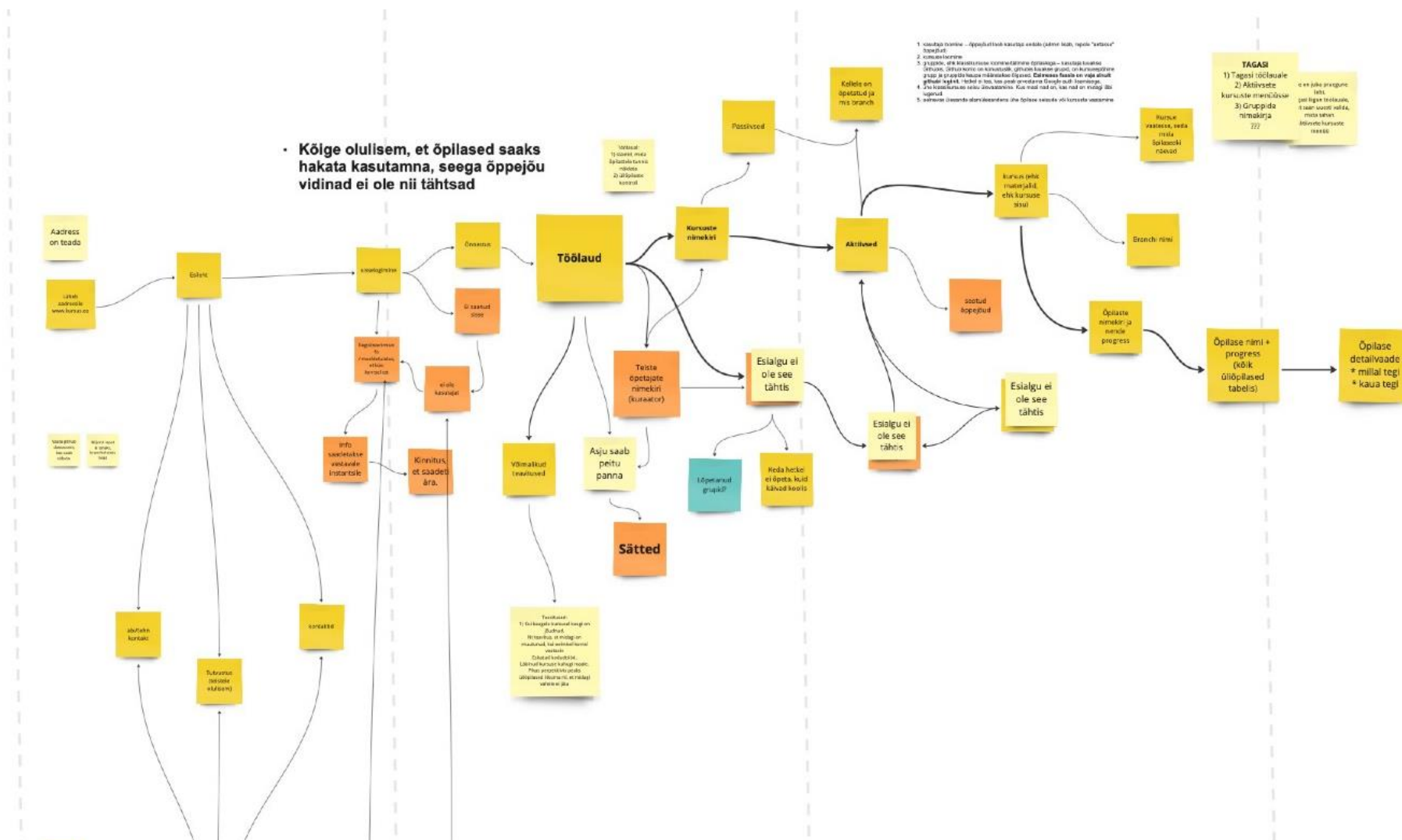
- Täiesti ebaoluline
- 1
- 2
- 3
- 4
- 5
- Ikka, ei taha kopitanud keskkonnas õppida

23. nimeta ka mõni halba kasutuskogemust pakkuv e-õppe keskkond ja kirjelda lühidalt, mis selle halvaks tegi?

24. Lisa siia kõik see, mida arvad, et siin küsitluses veel olema peaks



### LISA 3. KASUTAJA TEEKOND, JUTUSEIN.



## LISA 4. IDEESEIN

