

Tallinna Ülikool
Digitehnoloogiaste instituut
Informaatika

Kujundusmalli loomine sisuhaldussüsteemile Magento

Bakalaureusetöö

Autor: Raul Gordejev

Juhendaja: Romil Rõbtšenkov

Autor:””2017

Juhendaja:””2017

Instituudi direktor:””2017

Tallinn 2017

Autorideklaratsioon

Deklareerin, et kaesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Koik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt parinevad andmed on viidatud.

.....

(kuupäev)

.....

(autor)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Raul Gordejev (sünnikuupäev: 08.06.1995)

1. annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Kujundusmalli loomine sisuhaldussüsteemile Magento“, mille juhendaja on Romil Rõbtšenkov, säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositooriumis.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tallinnas, _____

allkiri ja kuupäev

Sisukord

Sissejuhatus	5
1 E-kaubandus.....	7
2 Magento	9
2.1 Magento versioonid	11
2.2 Magento 2 ja tulevik.....	13
2.3 Magento 1 ja Magento 2 erinevused	13
2.4 Magento kujundusmallid	17
3 Kujundusmalli arendus	18
3.1 Magento 2 seadistamine	18
3.2 Kujundusmalli loomine ja paigaldamine.....	19
3.3 Arendustööriistad	19
3.4 Kujundusmalli arendus.....	22
4 Analüüs ja valideerimine	30
4.1 Lehe kiiruse mõõtmine	30
4.2 Mobiilsed vaated	31
Kokkuvõte.....	32
Kasutatud kirjandus.....	33
Summary	35

Sissejuhatus

E-kaubandus on tänapäeval muutunud väga populaarseks ja e-poodide loomiseks iga päevaga aina rohkem juurde. See muudab inimeste ostlemise mugavamaks ja praegusel hetkel saab juba peaaegu kõike eluks vajalikku osta e-poodidest ning lasta see endale koju kohale toimetada. E-poodid annavad klientidele võimaluse ka enne poodi minekut valida internetist endale huvi pakkuvad tooted välja, mis muudab ostlemist jällegi mugavamaks ja efektiivsemaks. E-poodide puhul ei ole sarnaseid barjääre nagu füüsilistel poodidel. Nagu näiteks kindel asukoht, mis limiteerib klientide arvu või määratud lahtioleku ajad. E-poodidega on võimalik hea turunduse korral meelitada kohale kliente üle kogu maailma ja väga suurtes massides (Shah, kuupäev puudub).

Käesoleva bakalaureusetöö käigus antakse ülevaade ühest populaarseimast e-kaubanduse sisuhaldussüsteemist Magento¹. Töö eesmärgiks on luua Magento 2 versioonile kujundusmall (ingl *template*), mille läbi antakse ülevaade põhilistest sisuhaldussüsteemi võimalustest ning eripäradest Magento 1 ja Magento 2 vahel. Loodava malli puhul ei keskenduta ilusale disainile, vaid olulistele tähelepanekutele, mida jälgida ise kujundusmalli luues. Tutvustatakse ka kahe versiooni sarnasusi. Teema valiku põhjuseks on autori varasem kokkupuude Magento 1 versiooniga ning tööalane vajadus õppida selgeks kujundusmallide loomine Magento 2 versioonis. Samuti on Magento 2 väga hiljuti avaldatud ning arvatavasti jääb tulevikus üheks enim kasutatavaks e-poodide loomise platvormiks.

Käesolev töö on suunatud arendajatele, kellel on varasem kokkupuude Magento 1 süsteemiga ning soov tutvuda Magento 2 versiooniga. Samuti ka neile, kellele pakub huvi e-kaubandus ning e-poe kujundusmalli loomise protsess Magento sisuhaldussüsteemis.

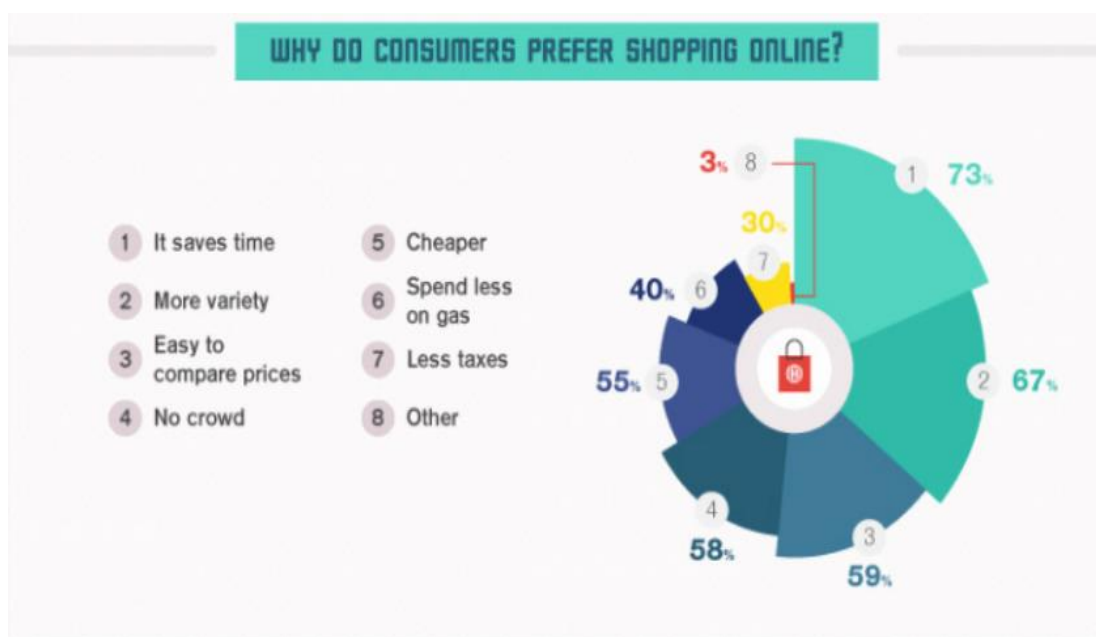
Töö on jagatud mitmeks osaks: esimeses peatükis on üldist informatsiooni Magento sisuhaldussüsteemi kohta ja selle ajaloost. Samuti on seal välja toodud ka suurimad erinevused Magento 1 ja Magento 2 vahel ning ülevaade kujundusmallidest. Teises peatükis antakse ülevaade Magento 2 seadistamisest, arendustööriistadest ning kujundusmalli arendusest ja paigaldamisest. Kolmandas peatükis analüüsitakse

¹ www.magento.com

valminud kujundusmalli nii kiiruse kui ka skaleerumise ja mobiilsete vaadete poole pealt.

1 E-kaubandus

E-poodidest ostmine on tänapäeval muutunud täiesti tavaliseks ostlemise viisiks ning paljud inimesed eelistavadki pigem tellida tooteid endale koju e-poodidest, selle asemel, et minna ise poodi kohale ja osta sealt. Tänapäevaks on võimalik e-poodidest kätte saada igast valdkonnast tooteid. Näiteks on Selverist võimalik virtuaalselt osta toitaineid ja lasta need koju tuua. Põhjuseid selleks on mitmeid (vt Joonis 1), kuid põhiliselt on inimeste jaoks e-poodides ostlemine vähem aeganõudev tegevus kui füüsiliselt poodi kohale minek. Samuti on e-poodides üldiselt suurem kaubavalik ja kergem tooteid omavahel võrrelda (Francis, 2016).



Joonis 1. Internetist ostmise eelistamise peamised põhjused (Francis, 2016)

E-kaubandus hõlmab endas igat tehingut või ülekannet, mis toimub elektrooniliselt, interneti vahendusel, kus ühe osapoole ülekande tegemisel saadab teine osapool talle soovitud toote. Alates toidukaupade ostmisest internetist kuni digitaalselt allalaetavate mängude ostmiseni (Francis, 2016).

Enamus e-poode on sarnase ülesehitusega: neil on virtuaalne ostukorv ja kassasüsteem (ingl *checkout system*). Tänu nendele, saavad kliendid lisada tooteid ostukorvi ning hiljem ilma suurema vaevata nende eest maksta ja seejärel oodata kauba kohalejõudmist (Francis, 2016).

Kõige esimene e-kaubanduse alla kuuluv tehing toimus aastal 1994, kus üks inimene tellis endale e-poe vahendusel koju pitsa. Suurem e-kaubanduse kasv algas aastal 2005, kui tekkisid siiani ühed suurimad e-pood eBay² ja Amazon³. Selleks ajaks oli tehtud üle interneti 109,4 miljardi dollari väärtuses makseid. Kogu maailma inimestest 15,7% kasutas e-poodide võimalusi sellel ajal. Nüüdseks on pool rahvastikust vähemalt ühe korra midagi tellinud e-poest. Kogu tehingute summa on kasvanud suuremaks kui 1,55 triljonit dollarit (Francis, 2016).

E-poodide kasvu algusega aastal 1994, hakati arendama ka SSL (Secured Socket Layers) krüpteeringut. Esimesena võttis selle vastu internetibrauser Netscape⁴. Krüpteeringu kiire vajaduse põhjuseks oli turvaliste ostude tagamine ja krediitkaartide andmete varastamise vähendamine. Ilma krüpteeringuta oli väga lihtne saada kellegi teise personaalsed andmed. 1998 aastal, PayPal⁵ tekkimisega, kaasnes interneti maksete standardiseerimine, mis muutis ostlemist mingil määral mugavamaks ja veel turvalisemaks. 2004 aastal loodi krediitkaarte pakkuvate ettevõtete poolt PCI (Payment Card Industry) turvalisusstandardid. Tänapäevaks on erinevaid maksemeetodeid väga palju: alates mobiilsetest rahakottidest (ingl *mobile wallets*) kuni krüptovaluutadeni nagu näiteks Bitcoin⁶ (Francis, 2016).

² www.ebay.com

³ www.amazon.com

⁴ Netscape – aastal 1994 loodud veebibrauser. Populaarne 90ndatel kuid nüüdseks välja surnud.

⁵ www.paypal.com

⁶ www.bitcoin.com

2 Magento

Magento on vabavaraline e-kaubanduse (ingl *e-commerce*) sisuhaldussüsteem, mis on peamiselt loodud programmeerimiskeeles PHP. Magento tarkvara loodi esialgselt ettevõtte Varien Inc. poolt, kus suure panuse andsid ka vabatahtlikud arendajad (bssadmin, 2015).

Võrreldes teiste tuntud e-kaubanduse platvormidega (nt Shopify⁷) on Magento olnud alati vabavaraline. Lisaks on võimalik Magentot integreerida paljude teiste (majandus)tarkvaradega. Magentot peetakse väga mahukaks süsteemiks mis võib esialgu tunduda nii arendajatele kui ka sisuhalduritele arusaamatu ja keeruline. Magento sisuhaldussüsteemis on peale sisseehitatud e-poe funktsionaalsuste võimalus luua ka täiesti tavalisi sisulehti, sarnaselt WordPressiga⁸.

Esimest Magento versiooni hakati arendama 2007. aasta alguses ja esimene avalik beeta versioon anti välja 31. august 2007. Kui aasta peale seda anti välja esimene terviklik versioon (ingl *full release*), saatis Magentot suur edu ja seda hakkasid kasutama suured ettevõtted, nagu näiteks tehnoloogiaettevõtte Samsung, riiete bränd The North Face ja kontoritarvete jaemüüja Office Max. Magento'st on arendatud eraldi ka veel täiustatud versioon nimega Magento Enterprise. Selle tegemise põhjuseks võib pidada suurte ettevõtete huvi Magento vastu (Morrioni, 2015).

Aastal 2010 ostis eBay 49% Varienist ning ülejäänud 51% osteti 2011 aasta veebruaris. Peale Varieni ära ostmist liidestati see ka Paypaliga, mis andis veel suurema eelise teiste e-kaubanduse sisuhaldussüsteemide ees. Samuti muudeti siis ka nimi Varien ümber Magentoks (Morrioni, 2015).

2015 aastal läbiviidud uurimises selgus, et Magento kasutamine moodustub 29,8% kogu e-kaubanduse platvormidest. Uurimises oli ligikaudu 30 erinevat platvormi. 2015 aasta novembris avaldati Magento 2.0, millega kaasnes palju uuendusi platvormi struktuuris kui ka poe halduse juures (aheadworks, 2015). Hetke seisuga on erinevate Magento süsteemide peale kokku ehitatud üle 250000 e-poe ülemaailmselt (Magento, kuupäev puudub).

⁷ www.shopify.com

⁸ www.wordpress.org

On palju põhjuseid, miks valida e-poe platvormiks just Magento. Saab välja tuua järgmised (Imre & Gordons, 2016):

- Magento Community Edition on alati olnud ja jääb alatiseks vabavaraliseks, mille tõttu on seda võimalik igapidi kohandada ja edasi arendada.
- Võimalik hallata mitmeid e-poode ja veebilehti ühe ainsa sisuhaldussüsteemi paigaldusega.
- Kujundusmallide taandumise süsteem (ingl *theme fallback*).
- Magentot on kerge laiendada lisades juurde mooduleid.
- Hea optimiseerida otsingumootorite jaoks (ingl *Search Engine Optimization*)
- Suur kommuun [Magento Stackexchange](https://www.stackexchange.com/questions/tagged/magento)⁹, kus on üle 53000 Magento teemalise küsimuse.
- Magento on skaleeruv, mis tähendab seda, et selle peale saab teha nii väiksemaid kui ka suuri (st. kümnete tuhandete toodetega) e-poode.
- Sisseehitatud kasutajate loomine ja haldamine, arvete ja tellimuste saatmise süsteem, mitmekeelsus, toodete kerge sisestamine ja haldamine, ülemaailmsete valuutade tugi, turundusvahendid, kerge maksesüsteemi liidestuse võimalus ja kõrge turvalisus.

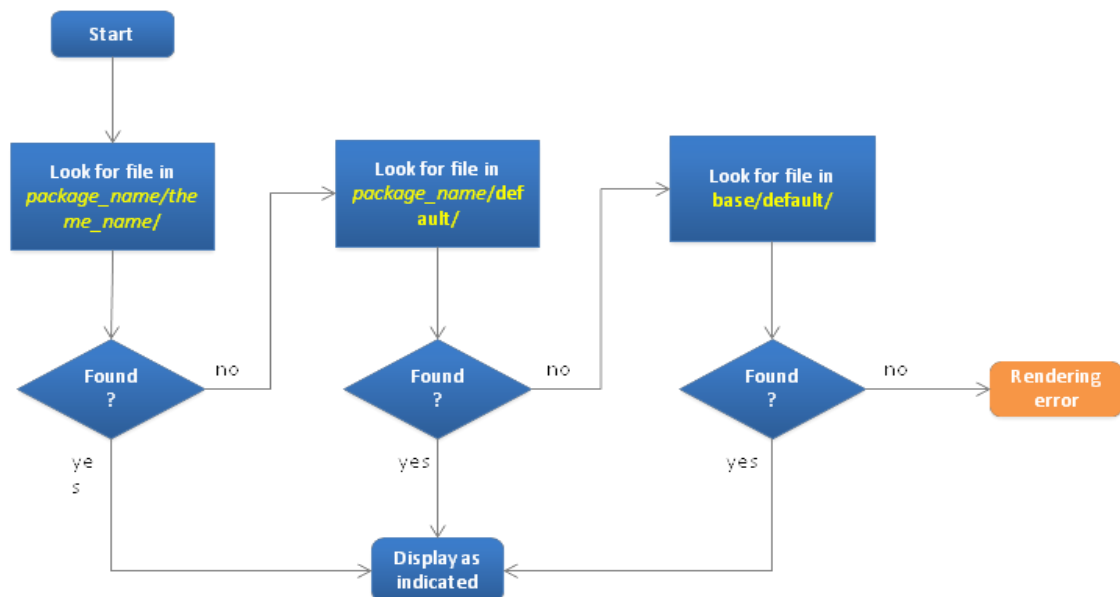
Kujundusmalli luues on välja toodud punktidest kõige olulisem kujundusmallide taandumise süsteem, mis võimaldab uue kujundusmalli aluseks võtta mingi olemasoleva malli ning seda täiendada. See muudab malli struktuuri loovate failide (laiendid phtml ja xml) haldamise lihtsaks, sest uues mallis muudatuste tegemiseks tuleb ainult muudetavad failid uue malli kausta tõsta. Ülejäänud tuleb taandumisega automaatselt, juhul kui tegu pole ainsa malliga. Sisuhaldusest saab määrata peamise kujundusmalli ja sellele järgneva. Samuti saab ka valikuliselt kujundusmalli osasid asendada uutega. Näiteks võib asendada ainult CSS failid ja jätta struktuur samaks või vastupidi.

⁹ www.stackexchange.com

Iga kujundusmall koosneb paketi nime (ingl *package name*) kaustast, kus omakorda võib olla mitmeid erinevaid mallide nimelisi kaustasi (ingl *theme name*). Peamiseks mallide kaustaks on *base/default*, millele taandutakse viimasena.

Taandumise hierarhia on järgmine (vt Joonis 2):

1. Otsitakse kohandatud kujundusmalli, mis on määratud sisuhaldusest aktiivseks.
2. Kui ei leita, siis otsitakse selle sama paketi *default* malli.
3. Kui seda ka ei leita, siis taandutakse *base* paketti *default* mallile, mis sisaldab Magento installeerimisega kaasa tulevaid malli faile.
4. Kui ei leita ka sealt, siis kuvatakse välja veateade.



Joonis 2. Kujundusmalli taandumise süsteem (Nikunj, 2013)

2.1 Magento versioonid

Magento esimene versioon anti välja 2008 aastal, ning Magento 1 viimane versioon hetke seisuga 2017. aastal, milleks on Magento 1.9.3.2. Aastal 2015 tuli välja ka Magento 2.0 ning aasta peale seda Magento 2.1. Neid kolme versioone arendatakse paralleelselt edasi. Suurte struktuursete muutuste pärast ei sobi Magento 1 kujundusmallid Magento 2.0 ega 2.1 versiooni peale. Küll aga sobivad Magento 2.0 mallid 2.1 peale ja vastupidi.

Tabel 1 on välja toodud Magento 1 erinevad versioonid alates algusest kuni praeguse hetkeni, kus on kirjas ka suurimad muudatused mis on mingit moodi seotud kujundusmallide loomisega (Whitby, 2015). Välja on toodud eraldi Mageno Community ja Magento Enterprise versioonid. Kõige olulisemad muudatused mallide loomise koha pealt on kujundusmallide taandumine ning skaleeruv algne kujundusmall.

Tabel 1. Magento versioonid läbi aegade (Whitby, 2015)

Community	Enterprise	Aasta	Muutused
1		2008	
1.1		2008	Magento API, virtuaalsed tooted
1.2		2008	Allalaetavad tooted, otsingu-funktsiooni täiustamine
1.3		2009	Flat catalog
1.4	1.8	2010	Kolme astmeline kujundusmallide taandumine, WYSIWYG editor, vidinad (ingl <i>widgets</i>), compiler
1.6	1.11	2011	Persistent cart, mitme andmebaasi tugi
1.7	1.12	2012	Mobiilvaate kujundusmall, REST API, CAPTCHA, kliendigruppide hinnad, VAT ID valideerimine
1.8	1.13	2013	Redis vahemälu, täiustatud maksude arvutamise süsteemi
1.9	1.14	2014	Lõpmatu kujundusmallide taandumine, skaleeruv kujundusmall, PHP5.4 tugi, PHP5.5 tugi, MySQL5.6 tugi

Tabel 2 on välja toodud iga Magento põhiversiooni kohta hetkel uusimad alamversioonid. Magento 2.0 ja 2.1 versioonidele tuleb sageli juurde uusi alamversiooni ja need arenevad pidevalt (Magento, 2017).

Tabel 2. Hetkel uusimad Magento versioonid

Versioon	Alamversioon
Magento 2.1	2.1.5
Magento 2.0	2.0.13
Magento 1	1.9.3.3

2.2 Magento 2 ja tulevik

Kuna Magento 2 avaldati alles üsna hiljuti ja Magento 1 on paljudele väga tuttavaks saanud, siis pole veel kõik üle läinud Magento 2 peale ning endiselt tehakse poode ka Magento 1 peale. Hetkel ei soovitata kiirustada Magento 2 peale üle minemisega, juhul kui just ei ole vaja parandada poe jõudlust. Kui praegune e-pood töötab ilma probleemideta Magento 1 peal, siis võib vabalt paar aastat veel oodata ja alles siis hakata mõtlema uuendamise peale. Magento toetab oma vanemat versiooni veel paar aastat ning teeb selle aja jooksul pidevalt uuendusi. Ilmselt küll mitte midagi suurt kuid enamasti turvaaukude tekkimisel nende parandusuuendusi (ingl *security patch*) (Stanic 2015).

Uuenduse ja ülemineku teeb raskeks ka varem mainitud asjaolu: kujundusmallid ja vidinad ei kohandu uemale versioonile, mis tähendab, et tuleks ehitada pood ülesse sisuliselt nullist. Teisest küljest annab see võimaluse uut e-poodi tehes rakendada uusi ideid ja teha suuremaid muudatusi tulevikuks (Stanic, 2015).

2.3 Magento 1 ja Magento 2 erinevused

Kujundusmallide loomise koha pealt on erinevusi palju. Ühed suuremad on kindlasti hüperteksti märgistuskeel HTML5, kaskaadlaadistiku CSS3 tugi ja kaskaadlaadistiku eeltöötlusvahendi LESS-i¹⁰ integratsioon kujundusmallidega. See tähendab, et enam ei ole Magentoga kaasa tulevad kujundusmallid kujundatud CSS-iga, vaid LESS-iga.

JavaScripti poolelt on vahetatud Prototype raamistik, mida kasutas Magento 1, jQuery¹¹ vastu. Rohkelt on kasutusel ka jQuery UI¹². JavaScripti laadimine toimub Magento 2-s kõik läbi requireJS-i, mis tähendab, et enam ei pea lisama *script* elemente käsitsi ja ei pea muretsema kas JavaScripti failid laevad ära õiges järjekorras, sest requireJS-s¹³ saab ära määrata vajalikud lisad ja failide laadimise järjekorra (Thompson, 2016).

Magento 2-ga kaasa tulev kujundusmall on juba algusest peale skaleeruv (vt Joonis 3). See tähendab, et seda saab mugavalt vaadata nii arvutist kui ka väiksema ekraaniga seadetelt. Magento üritab sellega saavutada täielikult skaleeruvaid e-

¹⁰ www.lesscss.org

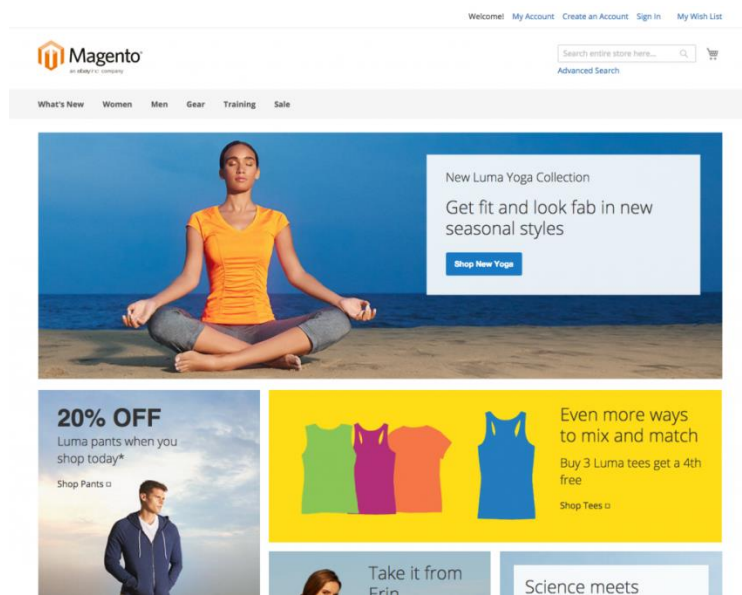
¹¹ <https://jquery.com/>

¹² <https://jqueryui.com/>

¹³ <http://requirejs.org/>

poodide kujundusi selle asemel, et teha ühest poest eraldi mobiilne ja arvuti versioon (Thompson, 2016).

LESS failide muutmine toimub nagu ka teiste failide muutmine taandumis süsteemiga. Muudetav fail tuleb lisada sama nimega enda kujundusmalli, järgides samat kaustade struktuuri. Magento 1-ga võrreldes on see lihtsam, sest Magento 1-s pidi CSS faili lisama kas läbi xml-i või käsitsi HTML *head* elemendi sisse (Thompson, 2016).



Joonis 3. Magento 2 kaasasolev kujundusmall.

Suuresti on muutunud ka failide struktuur võrreldes Magento 1-ga. All oleval joonisel (vt Joonis 4) on kõrvuti välja toodud Magento 1 ja Magento 2 algne struktuur.

Esmapilgul ei pruugi tunduda muutus nii suur, kui see tegelikult on. Magento 2-s ei ole enam *skin* kausta, kus võis leida staatilised CSS ja JavaScript failid ning pildid ja fondid. Magento 2-s on need paigutatud *app* kausta alla, kokku mallidele struktuuri andvate failide juurde. Lisaks on Magento 2 muutunud väga moodulite põhiseks: nüüd koosnevad moodulid enda struktuuri failidest ja staatilistest failidest, võrreldes Magento 1-ga kus võeti need endiselt eraldi kaustast. Suur erinevus on ka Magento enda failide (ingl *core files*) paiknemine. Magento 1-s olid nende enda kujundusmallide failid *app* kausta all kohe algselt, kuid Magento 2-s asuvad need *vendor* kaustas. Näiteks, kui on soov kasutades failide taandumise süsteemi muuta *checkout* mooduli faili *onepage.phtml*, siis tuleb võtta see fail

`vendor/magento/module-checkout/view/frontend/templates/onepage.phtml` asukohast ja asetada enda kujundusmalli asukohta `app/design/frontend/<paketi nimi>/<malli nimi>Magento_Checkout/templates/onepage.phtml`.

Oluline on siinkohal tähele panna, kuidas moodulite nimed ümber sõnastatakse. Kui mooduli nimi on algselt `module-checkout`, siis muutub see `Magento_Checkout`-ks. Tuleb ära võtta eest `module` ning asendada see `Magento`-ga. Kui nimi sisaldab rohkemat kui kahte sõna, mis on sidekriipsuga eraldatud, siis tuleb need sõnad kokku kirjutada ja suure algustähega.

Magento 1		Magento 2
..		..
app		.git
dev		app
downloader		bin
errors		dev
includes		lib
js		muu
lib		node_modules
media		phpserver
shell		pub
skin		setup
var		update
.htaccess	7 KB	var
.htaccess.sample	6 KB	vendor
api.php	4 KB	.htaccess
cron.php	3 KB	.htaccess.sample
cron.sh	2 KB	.php_cs
favicon.ico	2 KB	.travis.yml
get.php	6 KB	CHANGELOG.md
index.php	3 KB	composer.json
index.php.sample	3 KB	composer.lock
install.php	7 KB	CONTRIBUTING.md
LICENSE.html	11 KB	CONTRIBUTOR_LICE...
LICENSE.txt	11 KB	COPYING.txt
LICENSE_AFL.txt	11 KB	Gruntfile.js
mage	3 KB	index.php
php.ini.sample	1 KB	LICENSE.txt
RELEASE_NOTES.txt	580 KB	LICENSE_AFL.txt
		nginx.conf.sample
		package.json
		php.ini.sample

Joonis 4. Magento 1 ja Magento 2 failide struktuur

Alates Magento 2.0 versioonist on võimalik siduda kujundusmallid Composeriga¹⁴. Selleks tuleb lisada malli kausta `composer.json` (vt Joonis 5) fail ning luua Composeri konto. See tegevus pole vajalik kujundusmalli loomiseks, aga on hea lisavõimalus. Composer võimaldab luua paketi, kus on määratud kõik kujundusmalli sõltuvad osad (ingl *dependencies*). Loodud paketti saab jagada teistega või kasutada teiste loodud pakette (Thompson, 2016).

```
{
  "name": "magento/theme-frontend-luma",
  "description": "N/A",
  "require": {
    "php": "~5.5.0|~5.6.0",
    "magento/theme-frontend-blank": "1.0.0-beta6",
    "magento/framework": "1.0.0-beta6",
    "magento/magento-composer-installer": "*"
  },
  "type": "magento2-theme",
  "version": "1.0.0-beta6",
  "license": [
    "OSL-3.0",
    "AFL-3.0"
  ],
  "extra": {
    "map": [
      [
        "*",
        "frontend/Magento/luma"
      ]
    ]
  }
}
```

Joonis 5. Näide blank kujundusmalli `composer.json` failist.

Muutunud on ka lehe plokkide asetuse muutmine (ingl *layout*). Magento 1-s oli *layout* kausta võimalik luua `layout.xml` fail, kus sai teha kõik lehe kujunduse muudatused (ingl *page layout changes*). Magento 2-s enam seda võimalust pole, vaid kõik xml failid asuvad moodulite sees. Kui on soov muuta mingi mooduli asetust, siis tuleb teha selle mooduli xml failist koopia enda kujundusmalli ning teha muudatusi seal. Magento 2-ga tuli juurde *move* käsklus, mis võimaldab lehel plokk ühe koodireaga ümber tõsta. Magento 1-s pidi selleks esiteks otsima üles koha kus on plokk lisatud ja selle eemaldama (ingl *unset*). Seejärel lisama selle uuesti soovitud kohta (Thompson, 2016).

Konteinerid ja blokid (ingl *containers and blocks*) on teise tähendusega Magento 2-s kui see oli Magento 1-s. Magento 2-s annab see võimaluse luua HTML elemente ja neile atribuute lisada läbi xml failide. Näiteks saab lisada HTML elementidele klassi ja id nime, mis võivad muuta elemendi välimust või funktsionaalsust.

¹⁴ getcomposer.org

Magento 2 toetab brauseritest Internet Explorer 9 ja sellest uuemat, Mozilla Firefox'i ja Google Chrome kahte viimast versiooni, Safari 5 ning iOS6 Safarit (Vdovin, 2015).

2.4 Magento kujundusmallid

Üheks suurimaks keskkonnaks, kust on võimalik osta Magento kujundusmalle, on themeforest¹⁵. Praeguse seisuga on seal keskkonnas Magentole müügil 763 erinevat kujundusmalli. Paraku pole seal võimalust välja filtreerida malle ainult Magento 2 jaoks, kuid üldiselt kuulsamad ja ostetuimad (ingl *bestseller*) kujundusmallid sisaldavad nii Magento 1 kui ka Magento 2 peale optimeeritud malle. Mallide hinnad varieeruvad alates 29\$ kuni 299\$. Väga paljud kujundusmallid sisaldavad endas omakorda mitut eri vaadet, mis on küll sarnase kujundusega kuid erinevad struktuuri poolest. Kõige suurema müügiga on hetkel Ultimo¹⁶ kujundusmall, mida on müüdud kokku ~17000 korda ja mis maksab 99\$. Sellel on peaaegu poole rohkem oste kui teisel kohal oleval kujundusmallil. Populaarsed kujundusmallid on üldiselt väga kvaliteetsed ja hästi tehtud, neid saab muuta oma soovi järgi lihtsalt sisuhaldusest. Näiteks saab valida mitme erineva otsinguriba vahel või vahetada värve üle kogu kujundusmalli. Pea kõikide kujundusmallidega on võimalik osta ka aastaks ajaks tugi (ingl *support*) ning vajadusel lasta kujundusmalli installeerida nende poolt. Kvaliteetsemate mallide puhul on hinna sees juba kolme kuni kuue kuuline tugi. Autori enda kogemuste põhjal on jäänud mitme populaarse kujundusmalli loojate kasutajatoest halb arvamus, kuna enamus oskavad väga minimaalsel tasemel inglise keelt ja seetõttu pole alati soovitud abi saanud. Küll aga on mallidega kaasa tulevad dokumentatsioonid põhjalikud ja sellest peaks tavaliselt piisama, et see ise edukalt paigaldada ja muudatusi teha.

¹⁵ <https://themeforest.net>

¹⁶ https://themeforest.net/item/ultimo-fluid-responsive-magento-theme/3231798?s_rank=1

3 Kujundusmalli arendus

Magento 2 toetab alates Apache 2.2 või sellest uuemat veebiserverit. Andmebaasiks sobib MySQL alates versioonist 5.6.x ning serveripoolseks skriptimiskeeleks (ingl *server-side scripting language*) PHP¹⁷ 5.5.x, 5.6.x ja ka PHP7. Kindlasti on vajalikud ka järgmised PHP lisapaketid (ingl *extensions*) (Vdovin, 2015):

- PDO/MySQL;
- Mbstring;
- Mcrypt;
- Mhash;
- simpleXML;
- curl;
- xsl;
- gd, ImageMagick 6.3.7 või uuem, või mõlemad nendest;
- SOAP;
- Intl.

Vajalik on ka minimaalselt 512Mb muutmälu (ingl *ram*). Võrreldes Magento 1-ga, on seda täpselt poole rohkem.

3.1 Magento 2 seadistamine

Magento 2 seadistamiseks on mitmeid erinevaid võimalusi. Veebimajutusteenuse pakkuja DigitalOcean¹⁸ pakub väga lihtsalt ühe klikiga Magento paigaldust (ingl *one-click install*). Selle miinuseks on vaid see, et hetkel nad ei paku kõige uuemat versiooni ja peab vähemalt 10\$ kuus maksma veebimajutuse (ingl *hosting*) eest. Selline lähenemine sobib inimestele, kellel pole palju kogemusi veebiserveritega.

Teine võimalus on kloonida Composer'i või GitHub'i¹⁹ repositoorium enda serverisse ja paigaldada Magento endiselt üsna väikse vaevaga. Sellisel juhul tuleb küll ise panna püsti arendusserver ning vajalikud komponendid. See sobib rohkem edasijõudnutele, kes teavad mida nad teevad ja mis probleemid võivad ees oodata.

¹⁷ www.php.net

¹⁸ www.digitalocean.com

¹⁹ Github.com

Selleks, et läbi composeri installeerida Magento, tuleb teha järgmised sammud (Chauhan, 2014):

- installeerida vajalikud komponendid oma serverisse;
- installeerida composer;
- kloonida Magento 2 repositoorium;
- luua andmebaas;
- muuta Magento kaustade õigused selliseks nagu nad olema peavad;
- käivitada Magento 2 installer.

On ka teisi alternatiive, mis sarnanevad viimati mainitud versiooniga. Näiteks võib composeri asemel kloonida Magento GitHubist või laadida alla Magento ametlikult kodulehelt *zip* või *tar* formaadis arhiveeritud (ingl *archived*) fail ning kopeerida see käsitsi läbi FTP (File Transfer Protocol) serverisse ja seejärel installeerida.

3.2 Kujundusmalli loomine ja paigaldamine

Selleks, et luua Magento 2-s uus kujundusmall, mis võtab aluseks kumbagi algsetest mallidest, tuleb luua asukohta *app/design/frontend/<paketi nimi>/<malli nimi>* uus kaust.

Seejärel tuleb lisada fail *theme.xml* ja *registration.php*. Need failid võib võtta Magento algsest teemast, ning muuta seal sees malli nimi ära. Samuti tuleb lisada kaust nimega *web*, kuhu sisse lähevad kõik JavaScript, LESS failid ning pildid ja fondid. Selleks, et sisuhaldusesse tekiks valikusse uus loodud teema, tuleb tekitada ka kaust *media* ja sinna asetada kas jpg või png formaadis eelvaate (ingl *preview*) pilt, mida näidatakse sisuhalduses malli nime kõrval (Magento, 2017).

Sisuhaldusest saab määrata kujundusmalli poe üldiste sätete all, disaini osas.

3.3 Arendustööriistad

Kujundusmallide loomise töö sisaldab endas palju CSS, phtml, JavaScript ja xml failide muutmist. Kuna Magento 2 on võtnud kasutusele CSS asemel LESS-i, siis tähendab see seda, et kuidagi tuleb kompileerida LESS ümber CSS-ks. Algselt on sisse ehitatud Magentosse serveripoolne LESS-i kompileerimine (ingl *server-side LESS compilation*) ja kliendipoolne (ingl *client-side*) kompileerimine. Seda saab vahetada sisuhaldusest, minnes sätete alt arendajate (ingl *developer*) menüüsse ja seal

Front-end development workflow sätteid muutes. Algselt on määratud serveripoolne kompileerimine ning kui on plaan kasutada Grunti²⁰, siis ei pea seal midagi muutma.

Serveripoolne kompileerimine otsib failide taandumis meetodit kasutades üles nõutud LESS failid ja suunab need edasi PHP LESS kompileerijale, mille tulemusena väljastatakse kompileeritud CSS failid järgmisesse asukohta *pub/static/frontend/<paketi nimi>/<teema nimi>/<locale>*.

See meetod on väga aeganõudev, sest peale igat muudatust peab tühjendama Magento staatilised failid, et saaks need asendada uute staatiliste kompileeritud failidega ja salvestada need Magento vahemällu (Magento, 2017).

Kliendipoolne kompileerimine sarnaneb osaliselt serveripoolsele kompileerimisele. Vahe on selles, et kasutatakse PHP kompileerija asemel LESS.js teeki.

Selleks, et serveripoolset kompileerimist kiirendada, kasutatakse Grunt-i, mis on JavaScriptil põhinev ülesannete jooksupäts (ingl *task-runner*). Magento 2-s on juba algselt olemas Grunti konfiguratsiooni failid ja käsud, kuid selle tööle panemiseks tuleb Grunt enne serverisse paigaldada. Selleks on vaja esiteks paigaldada serverisse JavaScripti käitussüsteem (ingl *runtime environment*) node.js²¹.

Selleks, et paigaldada näiteks Ubuntu²² serverisse node.js, tuleb luua serveriga turvaline ssh (ingl *Secure Shell*) ühendus. Paigaldamiseks võib jälgida õpetust „How To Install Node.js on an Ubuntu 14.04 server“²³.

Kui Grunt on installeeritud tuleb veel vaid konfiguratsiooni fail ära seadistada. Selleks on Magentoga algselt kaasas fail *themes.js* (vt Joonis 6), kuhu tuleb lisada enda kujundusmalli andmed samas stiilis nagu seal on algsed Magentoga kaasas olevad mallid. *Locale* alla tuleb panna sisuhaldusest valitud kujundusmalli peamine keel ning *files* alla kõik less failid, mida on soov kompileerida Gruntiga. Antud juhul on lisatud teemale juurde ka Bootstrap²⁴ raamistik ja *_extend.less* fail. Tähele tuleb panna kindlasti, et kirja pannakse vaid faili nimed, ilma laiendita.

²⁰ <https://gruntjs.com/>

²¹ <https://nodejs.org/en/>

²² <https://www.ubuntu.com/>

²³ <https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-an-ubuntu-14-04-server>

²⁴ <http://getbootstrap.com/>

```

21 module.exports = {
22   blank: {
23     area: 'frontend',
24     name: 'Magento/blank',
25     locale: 'en_US',
26     files: [
27       'css/styles-m',
28       'css/styles-l',
29       'css/email',
30       'css/email-inline'
31     ],
32     dsl: 'less'
33   },
34   laheteema: {
35     area: 'frontend',
36     name: 'Raul/laheteema',
37     locale: 'en_US',
38     files: [
39       'css/styles-m',
40       'css/styles-l',
41       'css/source/_extend',
42       'css/email',
43       'css/email-inline',
44       'css/bootstrap'
45     ],
46     dsl: 'less'
47   },

```

Joonis 6. Ekraanitõmmis kohandatud themes.js failist

Nüüdseks peaks olema Grunt paigaldatud ja valmis kasutamiseks. Põhilised käsklused, mis on Magento poolt juba kaasas ja väga abiks kujundusmallide loomisel on välja toodud Koodinäide 1.

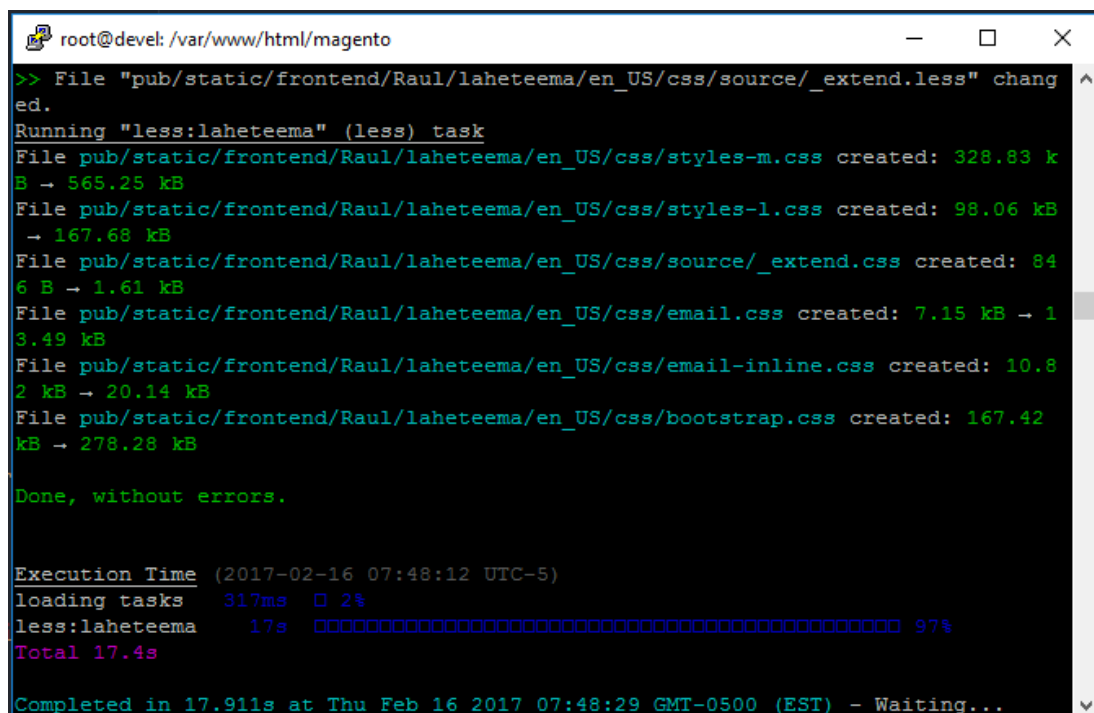
```

#1 failide kustutamine
grunt clean:<malli nimi>
#2 uute seoste loomine ja kompileerimine
grunt exec:<malli nimi>
#3 .css failide kompileerimine
grunt less:<malli nimi>
#4 automaatse kompileerimise käivitus pärast failides muudatuste salvestamist
grunt watch:<malli nimi>

```

Koodinäide 1. Grunt käsud

Esimene käsklus kustutab kõik malliga seotud staatilised failid kaustadest var ja pub/static. Teine käsklus loob uued seosed (ingl *symlink*) ja kompileerib uued staatilised failid. Kolmas käsklus kompileerib .css failid kasutades seoseid, mis tekivad eelmise käskluse jooksul. Koodinäide 1 välja toodud viimane käsklus on ilmselt kõige parem käsklus, mis teeb less failide kompileerimise väga lihtsaks. Grunt käivitab automaatselt kompileerimiseks vajalikud käsklused, kui teha muudatus mõnes less failis (vt Joonis 7). Peale LESS failis muudatuste salvestamist käivitatakse koheselt grunt:less käsklus, mis kompileerib uued css failid. Üldiselt võtab see protsess 15-20 sekundit aega, mis tähendab, et pärast kujunduse stiilifailide muutmist tuleb nii kaua oodata enne kui tulemust brauseris näeb (Magento, 2017).



```
root@devel: /var/www/html/magento
>> File "pub/static/frontend/Raul/laheteema/en_US/css/source/_extend.less" changed.
Running "less:laheteema" (less) task
File pub/static/frontend/Raul/laheteema/en_US/css/styles-m.css created: 328.83 kB → 565.25 kB
File pub/static/frontend/Raul/laheteema/en_US/css/styles-l.css created: 98.06 kB → 167.68 kB
File pub/static/frontend/Raul/laheteema/en_US/css/source/_extend.css created: 846 B → 1.61 kB
File pub/static/frontend/Raul/laheteema/en_US/css/email.css created: 7.15 kB → 13.49 kB
File pub/static/frontend/Raul/laheteema/en_US/css/email-inline.css created: 10.82 kB → 20.14 kB
File pub/static/frontend/Raul/laheteema/en_US/css/bootstrap.css created: 167.42 kB → 278.28 kB

Done, without errors.

Execution Time (2017-02-16 07:48:12 UTC-5)
loading tasks 317ms 2%
less:laheteema 17s 97%
Total 17.49s

Completed in 17.911s at Thu Feb 16 2017 07:48:29 GMT-0500 (EST) - Waiting...
```

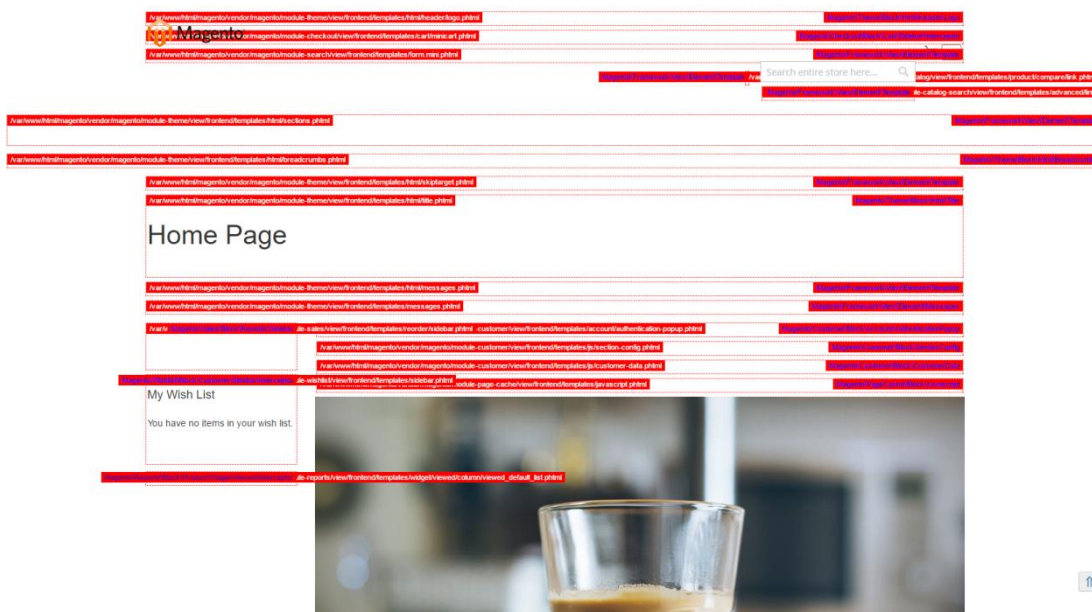
Joonis 7. Ekraanitõmmis grunt:watch käskluse tööst

3.4 Kujundusmalli arendus

Käesoleva töö raames arendab autor kujundusmalli. Põhilised muudatused tehakse põhiliselt avalehele, kuid on ka väiksemaid muudatusi ja parandusi teistel e-poe vaadetel. Enne kujundusmalli arendusprotsessi alustamist on autor lisanud e-poodi näidistooteid ning näidissisu. Kujundusmalli luues on kasutatud

versioonihaldussüsteemi GitHub, ning kõik kood ja viimased muudatused on seal repositooriumis²⁵ üleväl ja saadaval allalaadimiseks.

Kuna Magento koosneb väga paljudest omavahel seotud plokkidest, siis on väga raske hakata otsima õiget plokki kui pole Magentoga eriti kokku puutunud. Õnneks on Magentol võimalus lülitada sisuhaldusest sisse malli tee vihjed (ingl *template path hints*), mis lisab iga bloki juurde selle nime ja asukoha (vt Joonis 8). Neid saab sisse lülitada avades sisuhalduse ja minnes arendajasätete alla ning sealt lülitada sisse *Enabled Template Path Hints for Storefront* ja *Add Block Names to Hints*.



Joonis 8. Malli tee vihjed sisse lülitatuna

Allpool on välja toodud autori suurimad muudatused kujundusmalli arendamisel, mida autor käsitleb edaspidi põhjalikumalt:

- liuguri (ingl *slider*) ja bänneri (ingl *jumbotron*) lisamine avalehele;
- toodete kuvamine avalehel;
- uudiste plokk ja külgribale teadete ploki lisamine;
- lehe ülesossa kerimise nupp (ingl *scroll-to-top*);
- jalusesse pildi lisamine;
- soovikorvi ja toodete võrdluse bloki eemaldamine külgribalt;
- lehekülje päise muutmine;

²⁵ https://github.com/raulicious/magento_laheteema

- toote vaate muudatused.

Liugur ja bänner on lisatud avalehele kasutades Twitter Bootstrap²⁶ raamistiku komponente. Bootstrapi lisamine Magento 2-le on tunduvalt keerulisem kui see oli Magento 1 puhul. Põhjuseks on jällegi LESS-i kasutamine üle CSS-i ja requireJS-ga JavaScripti failide lisamine. Seega tuleb laadida alla vajaminevad Bootstrapi komponendid, mis on eraldi failidena ning seejärel luua üks ühine LESS laiendiga fail (näiteks bootstrap.less) kuhu käsuga `@import` liidestatakse alamfailid (vt Koodinäide 2).

```
@import 'bootstrap/_jumbotron.less';
```

Koodinäide 2. Näide `@import` käsust

See käsklus võtab kaustast nimega bootstrap faili `_jumbotron.less` sisu ja võimaldab seda kasutada. Samuti tuleb fail `bootstrap.less` lisada lehe HTML *head* märgistuste vahele, nagu lisaks CSS-i või JavaScripti tavalisele veebilehelegi. Magento 2 puhul on selleks eraldi xml laiendiga fail `default_head_blocks.xml`, mis tuleb ise luua oma uue kujundusmalli sisse, järgnevasse asukohta: `app/design/frontend/<Paketi nimi>/<Malli nimi>/Magento_Theme/layout/default_head_blocks.xml`.

Sarnaselt tuleb lisada ka Bootstrapi JavaScripti fail `bootstrap.js`. Kuna Magento 2-s käib JavaScripti lisamine läbi requireJS-i ja `bootstrap.js` vajab töötamiseks jQuery-t, siis peab kogu `bootstrap.js` koodi panema funktsiooni sisse mis võtab kasutusele jQuery (vt Koodinäide 3).

```
require(['!jquery'], function($){  
    -bootstrap.js faili sisu läheb siia vahele-  
})
```

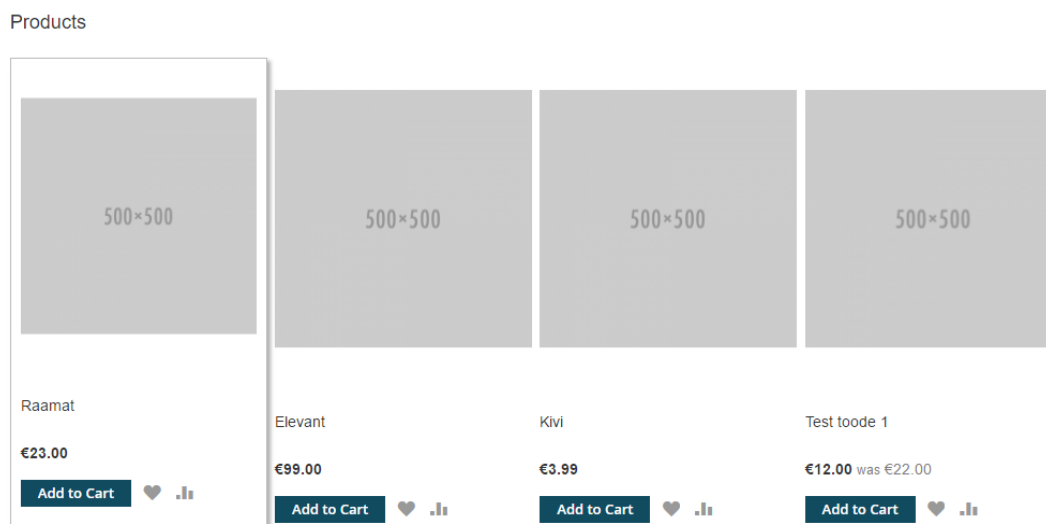
Koodinäide 3. jQuery kasutamine läbi requireJS-i

See funktsioon laeb jQuery enne `bootstrap.js` laadimist, mis on töötamiseks vajalik, sest `bootstrap.js` kood kasutab jQuery komponente.

²⁶ <http://getbootstrap.com/>

Liuguri ja bänneri välja kutsumiseks avalehel saab lisada vastava HTML koodi sisuhaldusest avalehele või läbi phtml laiendiga failide otse kuskile mujale moodulisse.

Toodete kuvamiseks avalehel on Magentol enda vidinad (ingl *widget*). Neid saab lisada läbi sisuhalduse, minnes soovitud lehele. Antud juhul on lisatud see vidin avalehele (vt Joonis 9). Vidinale on lisatud ka peale LESS-iga lohistamise efekt (ingl *hover effect*) failis *web/css/source/_extend.less*. Luues uut kujundusmalli sooviga kasutada kumbagi Magento algsetest mallidest alusena on kõige kergem luua enda kujundusmalli kausta fail *_extend.less* ja teha muudatusi sinna. Teine võimalus on kasutada *_module.less*, kuid sellisel juhul kirjutatakse kõik eelnevad väärtused üle.



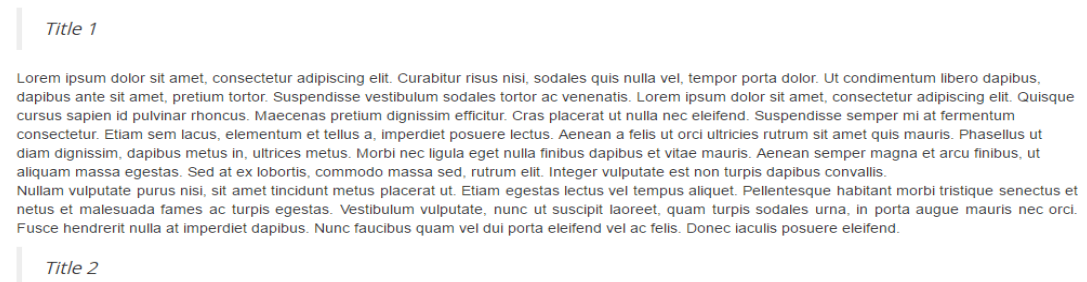
Joonis 9. Uute toodete kuvamise vidin

Lehele on loodud veel uudiste või teadete kuvamiseks mõeldud plokk, milleks on kasutatud Magento sisseehitatud JavaScripti kasutajaliidest (ingl *JavaScript UI*). Neid vidinaid on Magentol kaasas 15 ja antud kujundusmallis on kasutatud kahte erinevat. All olev koodijupp (vt Koodinäide 4) tekitab kokkupandava (ingl *collapsible*) vidina.

```
<div data-mage-init='{ "collapsible":{ "active": false, "animate": 200, "collapsible": true}}'>
<div data-role="title">Pealkiri, millele vajutades avaneb sisu</div>
<div data-role="content"> Sisu tekst läheb siia </div>
</div>
```

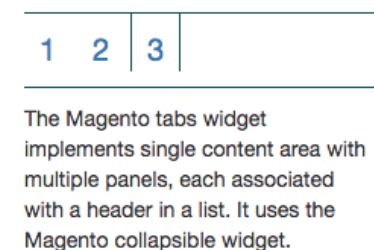
Koodinäide 4. JavaScript UI kokkupandava vidina lähtekood

Koodirea tulemusena tekkiv vidin on nähtav Joonis 10, kus *Title 1* nimeline plokk on avatud, mille tulemusena on selle all olev tekst nähtav. *Title 2* seevastu pole avatud ja tekst on peidetud. Kood on asetatud sisuhaldusest avalehele, sest siis on lihtne selle sisu muuta kasutades Magento sisseehitatud tekstiredaktorit. Võimalik on see vidin välja kutsuda ka otse kuskil malli failis.



Joonis 10. Magento UI kokkupanдав vidin

Teine JavaScripti kasutajaliidese vidin mida on töös kasutatud on *tabs* vidin (vt Joonis 11). See on asetatud soovikorvi moodulisse, sidebar.phtml faili ja see kuvatakse välja igal lehel kus on valitud külgribaga lehe asetuse. See vidin võimaldab näidata väikse ala peal palju informatsiooni. Numbrike vajutades kuvatakse selle all sisu. Sinna saab lisada nii pilte kui ka teisi vidinaid lisaks tekstile.



Joonis 11. Magento UI *tabs* vidin

See vidin on algselt Magento kujundusmallil kasutusel ka toote vaates. Vidina kasutamiseks tuleb võtta kasutusele jQuery UI läbi requireJS-i ning lisada soovitud kohta HTML kood (vt Koodinäide 5).

```
<div id="horizontal_tabs">
  <ul>
    <li><a href="#fragment-1"><span>1</span></a></li>
    <li><a href="#fragment-2"><span>2</span></a></li>
    <li><a href="#fragment-3"><span>3</span></a></li>
  </ul>
```

```

<div id="fragment-1"> <p>1-le vajutades kuvatav tekst</p> </div>
<div id="fragment-2"> <p>2-le vajutades kuvatav tekst </p> </div>
<div id="fragment-3"> <p>2-le vajutades kuvatav tekst </p> </div>
</div>

<script>
require([ 'jquery', 'jquery/ui' ], function($, accordion) {
    $('#vertical_tabs').accordion({ heightStyle: "content" });
}
);
</script>

```

Koodinäide 5. JavaScript UI *tabs* vidin

Kuna autor on näinud paljude valmismallide juures nuppu, mis tekib kui navigeerida lehel alla poole ja millele vajutades lohistatakse vaade lehe päisesse tagasi, siis on ka antud kujundusmallile selline funktsionaalsus loodud. Esiteks on selleks loodud HTML element soovitud kohta ja peale seda antud sellele nupule JavaScriptiga funktsionaalsus. Selleks on loodud enda kujundusmalli web/js kausta uus JavaScripti fail. Antud mallis on pandud selle faili nimeks custom.js (vt Koodinäide 6). Samuti on lisatud fail *head* elemendi sisse, nagu ka varasemalt juba näidatud on ja kuna funktsionaalsuse loomiseks on kasutatud jQuery, siis on pandud kood requireJS funktsiooni sisse mis laeb ära jQuery enne käivitamist.

```

require(['jquery'], function($) {
    $(document).ready(function() {
        $('.arrow').click(function() {
            $('html, body').animate({scrollTop: 0}, 800);
            return false;
        });
    });
    $(window).on("scroll", function() {
        var scrollPos = $(window).scrollTop();
        if (scrollPos <= 0) {
            $(".arrow").fadeOut();
        } else {
            $(".arrow").fadeIn();
        }
    });
});

```

Koodinäide 6. custom.js faili sisu

Esimene *.click* funktsioon lisab nupule funktsionaalsuse, mis peale vajutades kerib 800ms jooksul lehe ülesossa tagasi. Teine funktsioon peidab nupu ära kui ollakse juba lehe üleval osas. Nupule on lisatud ka kujundus, mis asub failis `_extend.less`

Et tuua välja mitmeid erinevaid lähenemisi, siis on antud kujundusmallis eemaldatud külgribalt 2 plokki, kahel erineval viisil, mis on aluseks võetud mallil *Luma* kohe seal olemas. Läbi xml laiendiga failide on väga mugav muuta elementide paigutust või neid lisada ja eemaldada. Üldiselt on igal moodulil enda xml failid, mis mõjutavad ainult seda sama moodulit, kuid *Magento_Theme* kaust hõlmab endas suuremat osa ja sealt kaudu on võimalik muudatusi teha suurema skoobiga. Selleks, et eemaldada toodete võrdlemise plokk võib võtta kas aluseks võetud mallilt, või luua ise *default.xml* fail asukohta: `app/design/frontend/<paketi nimi>/<malli nimi>/Magento_Theme/layout/default.xml`.

Kirjutades *remove* käsklus faili (vt Koodinäide 7), eemaldatakse toodete võrdluse plokk külgribalt igalt e-poe vaatest (kategooria vaade, toote vaate, avaleht jne). Bloki nime saab kätte lülitades sisse malli tee vihjed.

```
<referenceBlock name="catalog.compare.sidebar" remove="true"/>
```

Koodinäide 7. Toodete võrdlemise ploki eemaldamise koodijupp

Et näidata erinevaid tegevusi, siis on lisatud lehe jalusesse uus plokk, mille sees on pilt. Selleks on samuti mitu erinevat võimalust. Üks viis on minnes sisuhaldusest üldsätetesse ja disaini lahtrisse ning sealt saab lisada HTML koodi otse jalusesse. Teised võimalused on `footer.phtml` faili muutmise või sisuhaldusest uue ploki loomine ja selle jalusesse lisamine, nagu on ka antud töös tehtud. Hea tava on luua sellised lehe osad plokkidena ja sisuhaldusest, mida on vaja tulevikus muuta tihedamini. Samuti annab see võimaluse e-poe haldajale kes ei pruugi osata koodi kirjutada teha selliseid lihtsamaid muudatusi nagu piltide vahetamine. Antud kujundusmallis on loodud sisuhaldusest plokk, kuhu on lisatud pilt ja sejärel loodud uus vidin, mis sisaldab pildiga plokki. Vidina loomine on vajalik, kuna vidinaid saab lisada konteinerite (ingl *container*) sisse ja piirata soovi korral selle kuvamist ainult teatud lehe osades.

Väiksemad CSS muudatused hõlmavad endas kategooria menüü kujunduse muutmist ja selle lehe keskele asetamist, checkoutis kujunduse parandamist ning lehe päise ja footeri kujunduse muutmist. Kõik väiksemad muudatused on tehtud `_extend.less` failis ja viimased nendest on kajastatud ka GitHub repositooriumis.

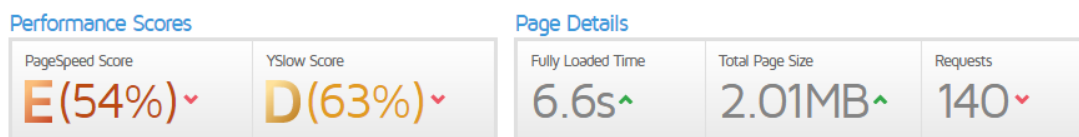
4 Analüüs ja valideerimine

Peale kujundusmalli arenduse lõpetamist on kasulik lülitada sisse vahemällu salvestamine ning ühendada (ingl *merge*) CSS ja JavaScripti failid. Samuti saab väga lihtsalt vähendada failide mahtu minifeerides koodi (ingl *minify*), seda kõike saab teha Magento sisuhaldusest sätete alt. Peale nende tegevuste tegemist peaks lehe kiirus sõltumata internetikiirusest mitmekordselt paranema, sest ei laeta iga kord lehele minnes seda täielikult nullist ning ühendatud CSS ja JavaScripti failid vähendavad lehte laadides päringute arvu.

Järgmistes alapeatükkides on analüüsitud loodud kujundusmalli kiiruse ning mobiilsete vaadete osas. Kuna Magento on mahukas süsteem ja e-pood ei tohiks olla aeglased, siis on kasulik alati testida lehe kiirust ja võimalusel seda parandada. Samuti on tänapäeval väga levinud mobiilsetest seadetest ostlemine ja kui e-pood ei ole mobiilisõbralik võib minna potentsiaalseid kliente kaduma.

4.1 Lehe kiiruse mõõtmine

Lehe kiiruse analüüsimiseks on kasutatud vabavaralist keskkonda gtmatrix. Sisuhaldusest on aktiveeritud kõik vahemälu tüübid ja minifitseeritud JavaScript-i ja CSS-i kood ning vastavalt koondatud kokku ühte faili, kuid gtmatrix keskkond ei tuvasta seda, mistõttu mõõtmise tulemus võib olla moonutatud (vt Joonis 12). Ülejäänud tulemused on kõik kõrgemad kui 95/100. Lehe lõplikuks laadimisajaks pakutakse 6,6 sekundit ning suuruseks 2,01Mb.



Joonis 12. Analüüsi tulemused keskkonnas gtmatrix

Autor tegi veel käsitsi lehe kiiruse testi kasutades Google Chrome veebibrauserit, lülitades arendajate tööriistade alt vahemälu brauseris välja, internetikiirusega 8,13Mb/s keskkonna speedtest²⁷ testi järgi (vt Joonis 13). Sellist viisi testides näitab avalehe laadimise kiiruseks 2,47 sekundit ja lõplikuks ajaks 4,82 sekundit. Kategooria vaade avaneb lõplikult 4,2 sekundiga ning toote vaade 5,32 sekundiga.

²⁷ <http://www.speedtest.net/>

Name	Status	Type	Initiator	Size	Time	Waterfall	4,00 s ▲
logo.svg	200	svg+xml	(index)	4.1 KB	61 ms		
esimene-min.jpg	200	jpeg	(index)	80.8 KB	455 ms		
kolmas-min.png	200	png	(index)	505 KB	1.65 s		
500500.png	200	png	(index)	7.2 KB	390 ms		
500500_1.png	200	png	(index)	7.2 KB	426 ms		
500500_2.png	200	png	(index)	7.2 KB	449 ms		
500500_3.png	200	png	(index)	7.2 KB	471 ms		
TurvalineOstukoht.png	200	png	(index)	36.4 KB	247 ms		
jumbotron.png	200	png	(index)	595 KB	1.45 s		
loader-1.gif	200	gif	Other	17.2 KB	114 ms		

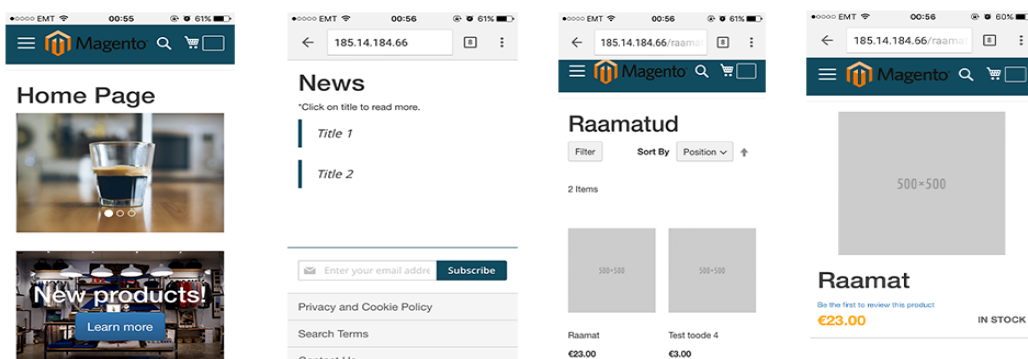
10 / 143 requests | 1.2 MB / 2.0 MB transferred | Finish: 4.82 s | DOMContentLoaded: 889 ms | Load: 2.47 s

Joonis 13. Google Chrome brauseriga lehe kiiruse testimine

Käesoleva töö raames loodud kujundusmall on virtuaalserveri peal millel on 1GB muutmälu ja ühe tuumaline protsessor. Kuna Magento nõuab palju ressursse, siis kindlasti oleks saanud paremad tulemused võimsama serveri peal. Näiteks kuuetuumalise protsessori ja 8gb muutmäluga arvuti peal laeb Magento 2.0, millel on näidistooted sees ja vahemälu sisse lülitatud avalehte umbes sekundiga. Magento e-poe laadimise kiirus on väga suhteline ja oleneb väga paljudest aspektidest, kuid erinevate testide järgi on see kiirem kui Magento 1 (Tataranovih, 2015).

4.2 Mobiilsed vaated

Kuna Magento on juba algselt kohanduva kujundusega ja sobib vaatamiseks igalt nutiseadmelt, siis loodud kujundusmallis ei ole väga suurt tööd mobiilsete vaadetega tehtud. Kõik muudatused on tehtud kasutades Magento kohanduvat kasutajaliidest. Joonis 14 on välja toodud iPhone 5s peal testitud mobiilsed vaated avalehe ülemisest osast, jaluse osast, kategooria vaatest ja toote üksikvaatest.



Joonis 14. Loodud kujundusmalli mobiilsed vaated

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua kujundusmall ja anda läbi selle ülevaade Magento sisuhaldussüsteemist ning põhilistest kohtadest ja tegevustest, mis tulevad ette luues kujundusmalle Magento 2 versiooni jaoks. Töö aktuaalsus väljendub selles, et Magento 2 on hiljuti avaldatud ning kujundusmallide loomise teemal pole veel väga palju informatsiooni. Lisaks oli autoril tööalane vajadus õppida selgeks Magento 2 versioonile kujundusmallide loomine.

Eesmärgi täitmiseks on tutvustatud üldisemalt Magento sisuhaldussüsteemi ja e-kaubanduse arengut ning selle ajalugu. Lisaks on antud ülevaade suurematest erinevustest Magento 1 ja Magento 2 versioonide vahel.

Eraldi on välja toodud bakalaureusetöö käigus loodud kujundusmallile tehtud suuremad muudatused ning tähtsamaid nendest lähemalt kirjeldatud. Töös on tehtud ka toiminguid mis korduvad ja neid ei ole täpsemalt selles peatükis välja toodud, vaid dokumenteeritud versioonihaldussüsteemis GitHub.

Valminud kujundusmalli analüüsi lehe laadimise kiiruse ja mobiilsete vaadete poolest ning jõuti järeldusele, et Magento on väga nõudlik platvorm ning selle efektiivsus oleneb mitmest aspektist. Küll aga sobib see suuremate e-poodide loomiseks väga hästi ja kaasa tulevad mallid juba algselt toetavad mobiilseid vaateid.

Autor leiab, et antud kujundusmalli on võimalik veel edasi arendada ja disainida, sest see ei ole terviklik, vaid on toodud näiteks, mille abil tuua välja suurimad eripärad Magento sisuhaldussüsteemi kohta.

Kasutatud kirjandus

Marroni, L. (2015). *History of Magento and its Rise to Ubiquity*. Loetud aadressil: <https://morrone.com/blog/history-of-magento/>

Francis, S. (20. detsember 2016). *The history of E-commerce*. Loetud aadressil: <https://www.ereviews.co.uk/global-facts-e-commerce/>

Whitby, M. (2015). *Magento version history*. Loetud aadressil: <https://makandracards.com/magento/25033-magento-version-history>

Aheadworks. (25. mai 2015). *Ecommerce Platforms Popularity, May 2015: Two Platforms Take Half*. Loetud aadressil <https://blog.aheadworks.com/ecommerce-platforms-popularity-may-2015-two-platforms-take-half/>

Magento. (kuupäev puudub). *Magento koduleht*. Loetud aadressil www.magento.com

Stanic, A. (2015). *Magento 1 vs Magento 2 – should I stay or should I go?* Loetud aadressil <http://inchoo.net/ecommerce/moving-from-magento-1-to-magento-2/>

Nikunj, K. (2013). *Magento theme fallback work flow*. Loetud aadressil <http://www.technologiespost.com/2013/04/magento-theme-fallback-work-flow.html>

Imre, G & Gordons, G. (2016). *Why is Magento the best platform for E-commerce?* Loetud aadressil: <https://www.quora.com/Why-is-Magento-is-the-best-platform-for-E-Commerce>

Magento. (2017). *Compile LESS with Grunt*. Loetud aadressil: http://devdocs.magento.com/guides/v2.0/frontend-dev-guide/css-topics/css_debug.html

Thompson, R. (16. veebruar 2016). *Magento 2 tutorial: what's new for frontend developers*. Loetud aadressil <https://inviqa.com/blog/magento-2-tutorial-what-is-new-front-end-developers>

Bssadmin. (2015). *What is Magento? The history of Magento*. Loetud aadressil: <https://bsscommerce.com/blog/magento-and-history-of-magento/>

Shah, D. (kuupäev puudub). *6 Reasons Why Ecommerce Business Has Become So Popular?* Loetud aadressil <http://www.selfgrowth.com/articles/6-reasons-why-ecommerce-business-has-become-so-popular>

Vdovin, V. (1. oktoober 2015). *Magento 1 and Magento 2 system requirements comparison*. Loetud aadressil <https://astrio.net/blog/magento-2-system-requirements-comparison/>

Chauhan, A. (5. detsember 2014). *Magento 2.0 – installation guide*. Loetud aadressil: <https://www.ubertheme.com/magento-editorial/magento-2-0-installation-guide/>

Magento. (2017). *Css preprocessing*. Loetud aadressil: <http://devdocs.magento.com/guides/v2.0/frontend-dev-guide/css-topics/css-preprocess.html>

DigitalOcean. (2017). *How to install Node.js on an Ubuntu 14.04 server*. Loetud aadressil <https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-an-ubuntu-14-04-server>

Magento. (2017). *Create a storefront theme*. Loetud aadressil: <http://devdocs.magento.com/guides/v2.0/frontend-dev-guide/themes/theme-create.html>

Tataranovih, A. (16. detsember 2015). *Magento 1 vs Magento 2 performance comparison: speed test results*. Loetud aadressil: <https://blog.amasty.com/magento-1-vs-magento-2-performance-comparison-speed-test-results/>

Summary

Custom Template Development for Magento CMS

Bachelor's Thesis

The goal of this Bachelor's thesis was to develop a custom template for Magento 2 content management system, describe the process and analyse it. The topic is relevant because of the recent release of a new version of one of the biggest and most famous e-commerce platforms. The built template is not a complete template that should be used on a working e-shop, rather than it is made to introduce the differences and similarities between Magento 1 and Magento 2 and to give a fundamental knowledge on building custom templates on Magento 2.

The first chapter of this thesis describes the background and history of Magento and e-commerce in general and compares Magento 1 and Magento 2 versions. In the second chapter, the template development process is described and the most complex parts of theme development on Magento 2 are further described. The third chapter is an analysis of developed themes page load speed and mobile views. The chapter gives an overview of Magento 2 rapidness and on what it depends on in addition to Magentos out-of-box responsive views that can be used to develop templates on.