

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Kalli Rüis 143062IAPB

**UUDISTEPORTAALI MODAALIDEL
PÕHINEVA KASUTAJALIIDESE
VÄLJATÖÖTAMINE**

Bakalaureusetöö

Juhendaja: Tarvo Treier
MSc

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kalli Rüis

20.05.2017

Annotatsioon

Antud töö eesmärgiks on luua uudisteportaalile modaalidel põhinev kasutajaliides, mis muudaks uudiste lugemise kasutajale mugavamaks, kui see on praegu. Modaal on dialoogiaken, mida kuvatakse ülejäänud veebilehe peal. Seda kasutatakse tihti reklaamide kuvamiseks või vormide täitmiseks, sest nii on võimalik suunata kasutaja tähelepanu tema ees olevale dialoogile nii, et taustal olev tegevusvoog ei katkeks.

Praktilise poole toetuseks uuritakse esmalt kolme populaarsemat eestikeelset uudisteportaalide ning nende rakendusi nutiseadmetele, et hinnata kasutajamugavust ja välimust. Seejärel analüüsitakse kasutuses olevaid modaale eri tüüpi ja eesmärgi täitvatel veebilehtedel, et mõista, milline peaks olema välimuselt ja ülesehituselt hõlpsasti kasutatav modaal, mis ei frustreeriks kasutajat. Teoreetilise osa lõpetuseks tutvutakse erinevate tehnoloogiatega, mida oleks võimalik kasutada praktilise töö realiseerimiseks ning valitakse välja konkreetsed keeled ning raamistikud, millega hakati kasutajaliidest lahendada.

Töö tulemusena luuakse modaalidel põhinev veebileht, mille avalehel kuvatakse kasutajale uudise pilt ning pealkiri, millele klikates näidatakse omakorda modaalise artikli sisu.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 7 peatükki, 20 joonist.

Abstract

The realisation of a news site interface based on modals

The main objective of this thesis is to create a user interface for a news site that is based on modals. A modal is a dialog window that is displayed on top of the rest of the web page. It is often used to display commercials or forms, because this way it is possible to direct the user's attention to the dialog in front of them without breaking the flow in the background.

In support of the practical side of the thesis three most popular Estonian news sites and their smartphone applications are examined to judge their user experience and appearance. Afterwards modals of different types and purposes on three webpages are analysed to understand what a modal should be like to be enjoyable to use. Finally, several technologies are studied to decide the best languages and frameworks to use to create the prototype.

As a result of the thesis a news site interface based on modals is created. On its main page the user is shown the headline and picture of an article. Upon clicking on either the picture or the headline, a modal is opened to display the contents of the article.

The thesis is in Estonian and contains 28 pages of text, 7 chapters, 20 figures.

Lühendite ja mõistete sõnastik

AJAX	<i>Asynchronous JavaScript And XML</i> . Tehnika, mis võimaldab serverilt pärida ja sinna saata andmeid ka pärast veebilehe laadimist [1].
API	<i>Application Programming Interface</i> ehk rakendusliides. Tarkvara rakendamise protokollide, reeglite ja vahendite kogum [2].
CSS	<i>Cascading Style Sheets</i> ehk kaskaadlaadistik. Laadilehed kirjeldavad, kuidas HTML dokumente esitada kuvaril, printeril või häälena [2].
DIY	<i>Do It Yourself</i> ehk tee seda ise.
<i>flexbox</i>	CSS3-le ehitatud kujundustehnika.
<i>html5lib</i>	Pythoni teek HTML-i dokumentide parsimiseks.
JSON	<i>JavaScript Object Notation</i> ehk andmete hoiustamise ja jagamise süntaks [3].
<i>lxml</i>	Pythoni teek HTML-i ja XML-i töötluks.
<i>Onepager</i>	Ühe pika lehena lahendatud veebileht.
<i>Open-source</i>	Avatud lähtekood. Tavaliselt tähendab programmi mille kood on avalik.
<i>RSS-feed</i>	Veebileht, mis annab teistel lehtedel võimaluse kasutada oma veebilehe materjali sisu, loob RSS dokumendi ja registreerib selle RSS kirjastajaga. [2]
UI	<i>User Interface</i> ehk kasutajaliides.
<i>Unicode</i>	16-bitine kood tekstiinfo esituseks [2].
UTF-8	Meetod Unicode märkide teisendamiseks. Märgid on igauks 16-bititi, teisendamine toimub 8-bitiseks märgiks [2].
<i>Web scraping</i>	Tehnika, millega eraldatakse veebilehe HTML-koodist veebilehel kuvatavad andmed, mida on võimalik hiljem kasutada [4].

Sisukord

1 Sissejuhatus	9
1.1 Taust ja probleem	9
1.2 Ülesande püstitus	9
1.3 Metoodika	10
1.4 Ülevaade tööst	10
2 Hetkel kasutuses olevate uudisteportaalide UI analüüs	12
2.1 Postimees	12
2.1.1 Postimehe digileht	12
2.1.2 Postimehe rakendus nutiseadmele	13
2.2 Delfi	15
2.2.1 Delfi veebileht	15
2.2.2 Delfi rakendus nutiseadmele	17
2.3 Õhtuleht	18
2.3.1 Õhtulehe veebileht	18
2.3.2 Õhtulehe rakendus nutiseadmele	20
2.4 Kokkuvõte	21
3 Modaaile kasutatavate veebilehtede analüüs	23
3.1 Pinterest	23
3.2 Basket	25
3.3 Facebook	25
3.4 Kokkuvõte	26
4 Vahekokkuvõte analüüsist	28
5 Liidese realiseerimiseks vajaminevate tehnoloogiate võrdlus	29
5.1 Uudiste ning nendega seonduva pärimine	29
5.1.1 PHP Simple HTML Dom Parser	29
5.1.2 Scrapy	29
5.1.3 BeautifulSoup	30
5.2 Liidese välimus	30

5.2.1 Bootstrap.....	30
5.2.2 W3.CSS	30
5.2.3 Bulma	30
5.2.4 AngularJS	31
5.3 Kokkuvõte	31
6 Kasutajaliidese realiseerimine	32
6.1 Liidese ülesehitus	33
6.2 Uudiste sisu pärimine	33
6.3 Uudiste kuvamine avalehel ja modaalis	34
6.4 Avalehe ja modaali ülesehitus	35
6.5 Reklaamide kuvamine loodud liidese	38
7 Loodud liidese analüüs	40
8 Kokkuvõte	41
Kasutatud kirjandus	42

Jooniste loetelu

Joonis 1. Ekraanitõmmis Postimehe veebilehest.....	13
Joonis 2. Ekraanitõmmis Postimehe nutirakendusest.....	14
Joonis 3. Delfi veebilehe teemalehe Forte ekraanitõmmis	16
Joonis 4. Delfi veebilehe teemalehe Ärileht ekraanitõmmis	16
Joonis 5. Ekraanitõmmis Delfi nutirakendusest.	17
Joonis 6. Õhtulehe veebilehe ekraanitõmmis.	19
Joonis 7. Ekraanitõmmis Õhtulehe nutirakenduse avalehest.	20
Joonis 8. Ekraanitõmmis Bootstrapi veebilehel olevast modaali näitest.....	23
Joonis 9. Ekraanitõmmis veebilehe Pinterest modaalist.....	24
Joonis 10. Ekraanitõmmis veebilehe Basket modaalist.....	25
Joonis 11. Ekraanitõmmis Facebooki lehel kasutusel olevast modaalist.	26
Joonis 12. Kasutajaliidese ülesehitus.....	32
Joonis 13. Ekraanitõmmis Javascripti failis kasutatud get päringust	34
Joonis 14. Ekraanitõmmis töös kasutatud post päringust	35
Joonis 15. Ekraanitõmmis ng-repeat direktiivist index.html failis.....	35
Joonis 16. Ekraanitõmmis loodud kasutajaliidese avalehest.	36
Joonis 17. Modaali loomise ja sisuga täitmise AngularJS-i koodi ekraanitõmmis	37
Joonis 18. Ekraanitõmmis loodud modaalist.....	37
Joonis 19. Üks reklaamide kuvamise võimalusi loodud kasutajaliidises.	38
Joonis 20. Üks modaalis reklaamide kuvamise lahendusi.....	39

1 Sissejuhatus

Kui varasemalt oli uudiste lugemise peamiseks meediumiks paberkujul ajaleht, siis tänapäeval tarbitakse neid enamjaolt interneti teel. Enamikul meist ei käi kodudes enam igapäevaselt ajalehed (näiteks Postimehe kuu keskmine tiraaž aprillis 2017 oli 44 800, mis arvestades, et Eestis elab umbes 1.3 mln inimest on üsnagi vähe), vaid soovi korral liigume vastavate lehtede veebiportaali ning loeme sealt vabalt valitud ajal kõige värskemaid uudiseid [5].

1.1 Taust ja probleem

Peaksime oma infovajaduse saama rahuldatus nii, et sellele ei kuluks palju aega või vaeva. Enamik veebiajalehti on hetkel üles ehitatud sarnaselt: uudisele klikates avaneb see kas uues vahelehes või sellel samal vahelehel. See muudab aga uudise lugemise aeglasemaks, sest uus vaheleht peab end täies mahus ära laadima. Samuti on lugejal raske jälgida, kuhu ta oma uudistevoos jõudnud on.

1.2 Ülesande püstitus

Käesoleva töö eesmärgiks on muuta lugejate jaoks uudiste tarbimine mugavamaks luues selle tarbeks uudisteportaalile brauseripõhine kasutajaliides, mille põhifookuseks on modaalid. Selmet avada uudis vahelehel, avatakse see hoopis modaal, mille sulgedes on kasutaja uudistevoos samas kohas, kus varemgi. Kuna terve veebilehe laadimise asemel lisatakse ühte modaali vaid vajalik info, peaks olema uudise kasutajani jõudmine kiirem. Selles töös keskendutakse just uudiste lugemise osale portaalist ning ei implementeerida kommenteerimise võimalust või muid võimekusi.

Prototüübi realiseerimiseks ei looda ise libauudiseid, vaid päritakse reaalsed uudised ühelt olemasolevalt uudisteportaalilt. Selle tarbeks luuakse parsija, mis kogub portaalist vajamineva info, mida kuvatakse avalehel ja modaal.

Antud töö võib autori arvates huvitada nii erinevaid ajalehtesid kui ka blogipidajaid.

1.3 Metoodika

Eesmärkide saavutamiseks kasutatakse erinevaid meetodeid. Esmalt analüüsitakse hetkel kasutuses olevaid uudisteportaale nii arvutitele kui nutiseadmetele, et leida nende valukohad ja tugevused. Seejärel uuritakse olemasolevaid veebilehtesid, mis kasutavad modaale, et leida nende positiivseid ja negatiivseid külgi. Järgmisena tutvutakse erinevate tehnoloogiatega, mida saab kasutada liidese realiseerimiseks ning valitakse välja need, millega liides valmis tehakse. Lõpuks, kasutades ära kogutud teadmisi, luuakse prototüüp.

1.4 Ülevaade tööst

Järgnevalt kirjeldatakse lühidalt iga peatüki sisu.

Esimeses peatükis hinnatakse hetkel kasutuses olevaid uudisteportaalide liideseid nii arvutitele kui nutiseadmetele ja tuuakse välja nende puudused ning tugevused.

Teises peatükis analüüsitakse erinevaid veebilehti, mis kasutavad sarnast, modaalidel põhinevat, ülesehitust. Hinnatakse nende puudujääke ning positiivseid külgi, et hiljem prototüübi ehituses kogutud informatsiooni ära kasutada.

Kolmandas peatükis on vaatluse alla võetud erinevaid tehnoloogiaid, mida oleks võimalik kasutada prototüübi loomisel. Välja on toodud nende lühitutvustused. Peatüki lõpus toob autor välja oma valiku antud töö realiseerimiseks koos selgitustega.

Neljandas peatükis tehakse lühikokkuvõtte tehtud analüüsist, mille käigus tuuakse välja uudisteportaalide ja modaale kasutavate veebilehtede positiivsed küljed, et prototüübi realiseerimisel oleks selge, mida tasub teha ja mida mitte.

Viiendas peatükis kirjeldatakse autori poolt loodud liidese realiseerimise samme ning tuuakse välja olulisem.

Kuuendas peatükis hinnatakse prototüüpi ning tuuakse välja selle nõrkused ja tugevused.

Seitsmendas ja viimases peatükis võetakse antud töö lühidalt kokku, korrates kõige tähtsamat.

2 Hetkel kasutuses olevate uudisteportaalide UI analüüs

Selles peatükis uuritakse lähemalt kolme populaarsemat eestikeelset uudisteportaaali, et leida nende tugevused ja nõrkused ning et uurida, millega peaks prototüübi realiseerimisel arvestama. Esmalt tuuakse välja portaalide lühitutvustus, seejärel hinnatakse neid lähtudes järgnevatest aspektidest: kasutusmugavus, välimus, uudistevoos navigeerimise lihtsus.

Samuti hinnatakse käesolevas peatükis samade aspektide alusel uudisteportaalide nutiseadmetele mõeldud rakendusi.

2.1 Postimees

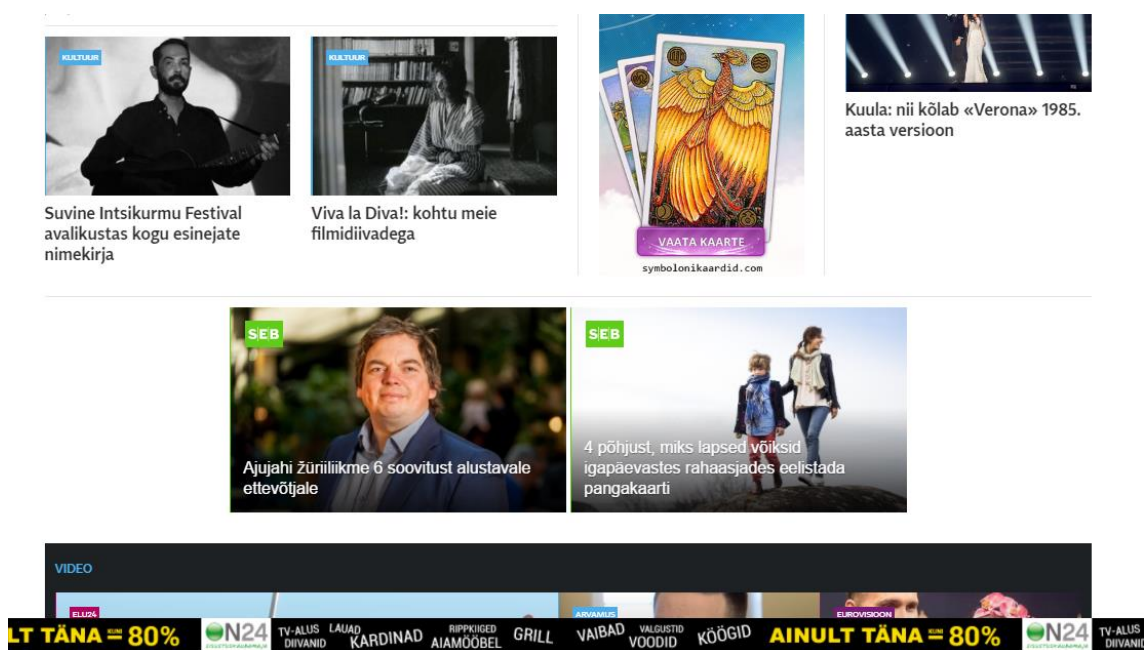
Ajaleht Postimees on Eesti suurim kvaliteetpäevaleht. Äripäeviti on Postimehe tiraaž ligikaudu 45 800 eksemplari, laupäevase Postimehe tiraaž aga 56 200 eksemplari. Argipäeviti ilmub Postimehe vahel Eesti suurim maakonnaleht Tartu Postimees, mis levib Tartu- ja Jõgevamaal ning Otepääl [6].

2.1.1 Postimehe digileht

Märtsis 2017 külastas Postimehe internetiväljaannet ligikaudu 504 361 inimest, mis teeb sellest populaarseima eestikeelse digilehe [7].

Postimehe välimus sõltub sellest, kas kasutaja veebibrauseris on reklaam peidetud või mitte. Kui reklaam on peidetud, näeb veebileht välja puhas ja korralik. Värvitemaatika on lihtne: pealehel on kasutatud helesinist ja valget, teemalehtedel valget ning mõnda muud ühte värvi. Pealeht on *one-pager*, st kogu oluline info on kuvatud ühel peamiselt lehel, mida mööda kasutaja alla liigub, et vanemate uudisteni jõuda, kuid olemas on ka võimalus liikuda eraldi teemalehele, nagu näiteks „Sport“ või „Arvamus“. Samuti on lehe paremas nurgas nupuke, mis kasutaja lehe algusesse tagasi viib.

Kui reklaamid pole aga brauseris blokeeritud, näeb Postimees autori arvates välja kirju. Sellele klikates on oht, et uues avalehes või mõne hüplikaknana avaneb reklaam, mida kasutaja näha ei tahtnud. Tekib tunne, nagu ühte vaatesse oleks lisatud liiga palju informatsiooni. Lisaks on keerulisene aru saada, kas tegemist on reaalse Postimehe artikliga või reklaamse taustaga refereeritud artikliga. Seda illustreerib järgmine joonis.



Joonis 1. Ekraanitõmmis Postimehe veebilehest

Uudiste lugemiseks Postimehe veebilehel tuleb uudise pealkirjale või pildile klikata. Selle tulemusel suunatakse kasutaja samal vahelehel uudise lingile. Selline käitumine on kasutaja vaatepunktist ebamugav, sest terve veebileht peab uuesti ära laadima, et uudist lugeda, mistõttu on lugeja sunnitud ootama. Samuti peab kasutaja uuesti ootama, et uudistevoosse tagasi jõuda, sest ka see peab uuesti laadima. Lisaks ei ole uudistevoosse tagasiliikumist tehtud kuigi mugavaks: valida on kas brauseri „Back“ nupu või peamenüü lingi „Postimees“ vahel. „Back“ nupp viib kasutaja enam-vähem samasse kohta uudistevoos, kus ta varem oli (mõnikord satub lugeja veidi kõrgemale loetud uudisest, mõnikord veidi madalamale), menüünupp suunab kasutaja aga tagasi Postimehe pealehe tippu. Artikli uues vahelehes avamine on aga tüütu – tihti läheb meelest ära, et antud artikkel juba avatud oli ja tehakse seda mitu korda või koguneb brauserisse lihtsalt üleliigselt vahelehti.

2.1.2 Postimehe rakendus nutiseadmele

Postimehe rakenduse välimus Androidi nutiseadmes on minimalistlik. Pealehel on kasutatud samu toone, valget ja helesinist, mida veebilehelgi. Teemalehtedel on igal ühel oma värv, et kasutaja kergemini aru saaks, kuhu ta sattunud on. Peamenüü on tehtud väikseks ja keritavaks, samuti on see avatav väiksest nupust ekraani vasakus nurgas. Oluline on, et peamenüü riba hakkab mööda uudistevoogu, mis on taaskord üles ehitatud

one-pager kujul, alla liikudes kaasa liikuma, mis muudab kasutajale lihtsaks valida soovi korral mõni teemaleht. Samuti vajab märkimist, et uudistevoos alla liikuma hakates on all paremas nurgas alati nupuke, mis suunab kasutaja vastavalt asukohast kas lehe lõppu või algusesse. Ekraani paremal nurgas asub sisselogimisvõimalus.

Uudiste lugemine Postimehe rakenduses Androidi nutiseadmele on mugav. Uudised laadivad ära piisavalt kiiresti, et ei tekiks kärstitust. Kui klikata uudisevoos huvipakkuvale uudisele, suunatakse kasutaja uude vaatesse. Seal on tal võimalik uudis läbi lugeda ning soovi korral, alla liikudes, vaadata ka sarnaseid uudiseid pealkirja all „Loe veel“. Kui aga liikuda tagasi, jätkab uudistevoog sealt samast, kuhu lugeja varem jõudis.

Reklaami tundub olevat nutiseadmele mõeldud rakenduses vähem – teisalt võib see aga olla paremini paigutatud. Iga kindla arvu uudiste vahel on väike reklaam, samuti on see olemas uudise lugemise vaates uudise alguses ning lõpus. Mõnikord avaneb hüpikakna kujul reklaam. Selline lahendus tundub parem, kui veebilehel olevad kaasa liikuvad reklaamid jaluses või uudiste keskel. Antud lahendust illustreerib järgmine joonis:



Joonis 2. Ekraanitõmmis Postimehe nutirakendusest.

2.2 Delfi

Delfi on 1999. aastal Eestis asutatud uudisteportaal, mis kuulub AS Ekspress Meediale. Delfi tegutseb Eestis (delfi.ee), Lätis (delfi.lv) ja Leedus (delfi.lt) [8]

Delfi portaali peamine missioon on anda kvaliteetseid ja operatiivseid uudiseid suurele osale ühiskonnast. Lisaks sellele pakub Delfi laia valikut täiendavaid teenuseid – e-post, fotoportaal, kuulutused, mängud, meelelahutus jpm [8].

Lätis ja Leedus on Delfi suurim uudisteportaal, mille saite külastab Gemius Audience andmetel kuus vastavalt üle 700 000 ja üle 1 100 000 inimese. Lisaks sellele annab Leedu Delfi välja ka mitmeid oma valdkonna juhtivaid ajakirju [8].

Eestis on Delfi populaarsuselt teine uudisteportaal Postimehe järel; seda külastas märtsis 2017 üle 490 000 kasutaja [7].

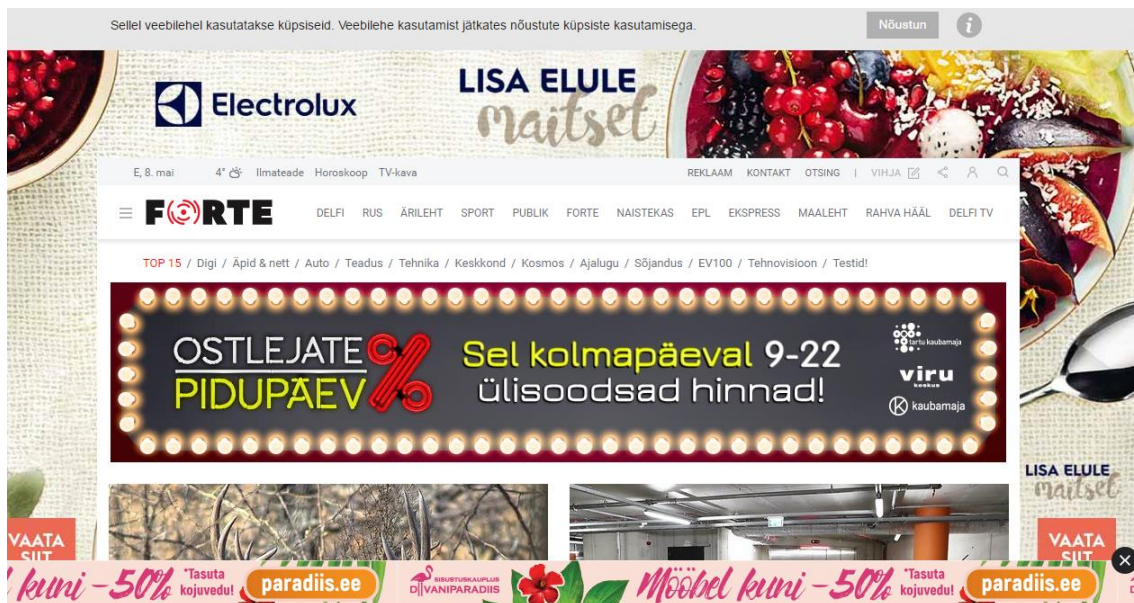
2.2.1 Delfi veebileht

Sarnaselt Postimehele on Delfi välimuses oluline roll reklaamidel. Neid on palju, nad vilguvad ja hüppavad, tekitades autoris oma müraga ebameeldivust. Ka siin tekib see hirm, et veebilehel ringi liikudes ja ekslikult valesse kohta klikates avaneb uuel vahelehel aken, mida lugeja avada ei tahtnudki.

Kui aga brauseris reklaamid keelata, on vaade hoopis teine. Taaskord näeb veebileht välja puhas ja minimalistlik. Välja paistavad ka Delfi põhivärvid: valge, kollane ja sinine. Kuna viimase kahega pole liialdatud, on selline kombinatsioon silmale meeldiv ning ei ärrita. Arvestada tuleb aga faktiga, et reklaamide keelamisel brauseris ei kuvata kasutajale mõningaid portaali artikleid.

Delfi ülesehitus on vägagi sarnane Postimehele: veebilehe alguses on riba, kust on võimalik liikuda erinevatele temalehtedele, kusjuures riba reisib uudistevoos liikudes kaasa.

Teemalehed näevad Delfis väga erinevad välja. Neil pole mingit konkreetset fonti või värvisüsteemi, ka logod on erinevad. See teeb kasutajale veidi keerulisemaks aru saada, kus ta konkreetset asub, kuid kuna ülemine menüüriba liigub uudistevoos kaasa, siis on temalehe logo alati nähtaval. Järgnevalt on jooniste abil välja toodud temalehtede välimuse erinevus.



Joonis 3. Delfi veebilehe teemalehe Forte ekraanitõmmis

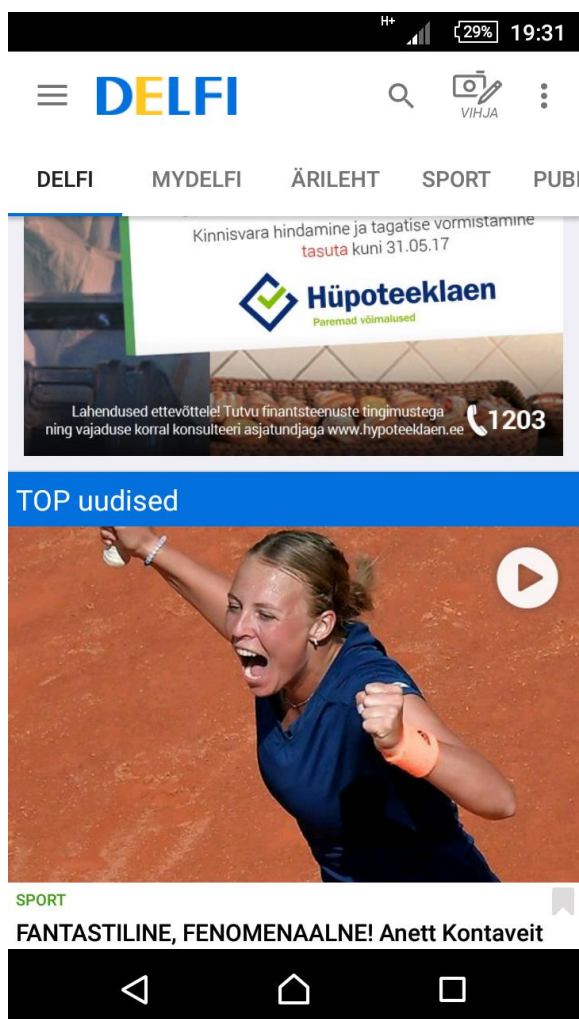


Joonis 4. Delfi veebilehe teemalehe ÄriLeht ekraanitõmmis

Uudise lugemiseks tuleb klikata uudise pealkirjale või pildile. Selle tulemusel suunatakse kasutaja uuele lingile samas vahelehel, kus tal on võimalik pärast lehe laadimist artikli sisuga tutvuda. Seejärel, liikudes tagasi uudistevoogu „Back“ nupuga, avaneb uudistevoog samast kohast, kuhu lugeja varem jõudis. Selline süsteem on veidi kohmakas, sest kasutaja peab kahes etapis ootama lehe laadimist.

2.2.2 Delfi rakendus nutiseadmele

Delfi Androidi rakendus on palju parema välimusega, kui seda on veebiportaal. Reklaame on vähem ning need, mis on, ei hakka nii kergelt silma. Ülemises osas on menüü, kust on võimalik valida kategooriaid. Menüü, sama moodi nagu veebis, on kaasaskäiv, mis on oluline, sest artiklid kuvatakse taas kõik ühel lehel, mis tähendab, et üks vaade on väga pikk ning kasutajal oleks tüütu kerida algusesse tagasi. Sisselogimine asub Androidile omase menüünupukese all. Värvilahendus on sama, mis veebilehel, mida on näha järgmisel joonisel.



Joonis 5. Ekraanitõmmis Delfi nutirakendusest.

Uudisele vajutades avaneb see uues vaates, kus kasutaja saab tutvuda artikli sisuga. Mõnikord kuvatakse enne uudiseni jõudmist selles faasis lugejale reklaami. Eriline on see, et artikli vaates liigub jaluses kaasa menüü, millel asuvad erinevad võimalused avatud artiklit jagada ning kommentaare lugeda ja lisada. Artikli lõpus on märksõnad;

pärast märksõnu sarnase teemaga uudised. Artikli lugemise lõpetades tagasi uudistevoogu liikudes jätkab lugeja sealt, kust enne omale huvipakkuva uudise leidis.

Üleüldiselt tundub Delfi mobiilirakendus mugavam ja hõlpsamini kasutatav, kui seda on veebileht, sest see näeb puhtam välja ning tundub, võib-olla vähesema reklaami tõttu, kiirem. Veidi tüütu on iga avatud artikli juures teade, et veebileht kasutab küpsiseid, ent seegi kaob ära, kui teate ristikese kaudu sulgemise asemel vajutada „Nõustun“ nupule.

2.3 Õhtuleht

Õhtuleht on 6 korda nädalas ilmuv tabloidajaleht, mida iseloomustavad pilkupüüdev kujundus ning rõhk uudistel ja meelelahutusel. Õhtuleht on suurima lugejaskonnaga trükimeediaväljaanne Eestis. Ajaleht ilmub internetis aadressil www.ohtuleht.ee [8].

Gemius Audience hinnangul on see Eestis külastatavuselt kolmas uudisteportaal, kusjuures märtsis 2017 oli külastajate arv ligi 260 000 [7].

2.3.1 Õhtulehe veebileht

Õhtulehe välimuses mängivad olulist rolli reklaamid. Neid on küll vähem ning nad tunduvad statsionaarsemad(st, ei hüppa, virvenda ega liigu uudistevooga kaasa) kui Postimehes või Delfis, ent sellegipoolest nõuavad tähelepanu. Säilib tunne, et lehel valesti klikates avanevad vahelehes mitmed erinevad pakkumised. Mida allapoole liikuda uudistevoos, seda vähemaks jääb nende arv, samuti puudub jaluses reklaam. Lehe välimus muutub autori hinnangul paremaks, kui keelata brauseris reklaamide kuvamine. Järgnevalt on toodud joonis Õhtulehe veebilehest.



Joonis 6. Õhtulehe veebilehe ekraanitõmmis.

Punane menüüriba päises liigub *one-pager* uudistevooga kaasa muutes lihtsaks valida mõnda konkreetset kategooriat. Teemalehele liikudes on näha, et nad erinevad üksteisest värvide poolest: tervise kategooria menüü on näiteks sinist värvi, melu aga kollane. Teemalehtedel menüüriba kaasa ei liigu, mis muudab omakorda kategooria muutmise keerukamaks.

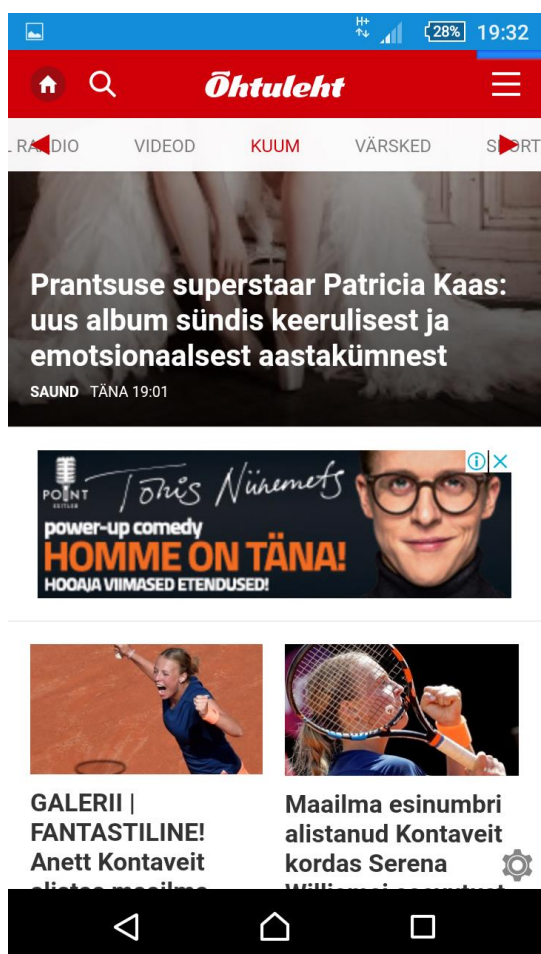
Uudisele liikumiseks peab klikkama uudise pealkirjal või pildil, pärast mida suunatakse kasutaja uuele lehele, kus laaditakse artikkel, selle all sotsiaalmeedias jagamise võimalus, sarnase temaga uudised ning kommentaarid. Uudiselt lahkudes tundub, et lugeja visatakse uudistevoos üsnagi suvalisele kohale, st kas veidi enne või veidi pärast äsja loetud artiklit.

Õhtulehte sirvides jäi veel silma see, et kuniks kõik reklaamid lehel on laaditud, liigub leht väga hakkivald ja katkendlikult. Samuti tundub autoritele tüütu, et reklaamide laadides lehe sisu hüppab: näiteks kategoorias „Galerii“ on viivuks näha esimese artikli pilti ja pealkirja, siis see aga liigutatakse alla poole, et menüü ja artiklite vahele mahuks ära suur reklaam.

2.3.2 Õhtulehe rakendus nutiseadmele

Õhtulehe rakendus Androidi nutiseadmele on välimuselt ilusam kui veebiväljaanne. *One-pager* uudistelehe päises on minimalistlik menüüriba, kus on võimalik teostada otsingut, valida kategooriat. Paremalt nurgas on Androidile omane menüünupp, mille all peitub sisselogimisvõimalus.

Reklaame tundub olevat rakenduses vähem kui veebilehel ja need on paremini asetatud. Ekraani väiksuse tõttu pole külgede lisatud reklaame, samuti on jalus puhas. Järgmisel joonisel on kuvatud rakenduse avaleht:



Joonis 7. Ekraanitõmmis Õhtulehe nutirakenduse avalehest.

Uudisele liikudes kuvatakse päises menüüriba „Kodu“ nupu, pealkirja ja otsinguga. Jaluses on menüüriba, kust saab lugeda kommentaare või jagada uudist sotsiaalmeedias. Viimase nupu asetus on samas veidi ebaõnnestunud: nimelt asub sotsiaalmeedias jagamise nupu juures ja pooleldi peal eraldi sätete nupp, mis keelab või lubab rakenduses

erinevaid toiminguid nagu näiteks heli, värin ja sõnumid. Artikli all asuvad samasse kategooriasse kuuluvad uudised.

Artiklilt tagasi uudistevoogu liikudes ei satu lugeja täpselt samasse kohta, kuhu enne. Voog avatakse mõnikord veidi enne äsja loetud uudist, mõnikord veidi hiljem. See on ebamugav, sest tekitab kasutajas segadust, sest ta ei saa aru, kuhu ta täpsemalt sattunud on, kas on neid uudiseid juba lugenud või mitte.

Üleüldiselt tundub Õhtulehe rakendus Androidile aeglasem, kui veebileht või Postimehe ja Delfi vastavad rakendused. Sellele aitab suuresti kaasa laadimist märkiv edasi-tagasi liikuv jooneke menüüriba üleval, mis on nähtav eelmisel joonisel, samuti hüplemine, mis tekib, kui artiklis olev reklaam ära laaditakse.

2.4 Kokkuvõte

Käesolevas peatükis analüüsiti kolme populaarsemat eestikeelset uudisteportaali. Selle käigus kirjeldati nende välimust ning ülesehitust, hinnati tugevusi ja puudujääke.

Kokkuvõtvalt võib öelda, et lähemalt uuritud portaalid Õhtuleht, Delfi ja Postimees on oma välimuselt ja ülesehituselt vägagi sarnased. Kõigil neist on *one-pager* tüüpi avaleht, mille ülemises osas on menüüriba, milles saab teha kategooriaavalikut, kõik suunasid oma kasutaja uudise lugemiseks uuele vahelehele. Samuti oli sarnane see, et veebilehed olid reklaamist üleüllastunud, mis on mõneti mõistetav, sest reklaamid on nende portaalide peamiseks tuluallikaks, ent lugejale muudavad veebilehe ebamugavaks ja halvasti loetavaks.

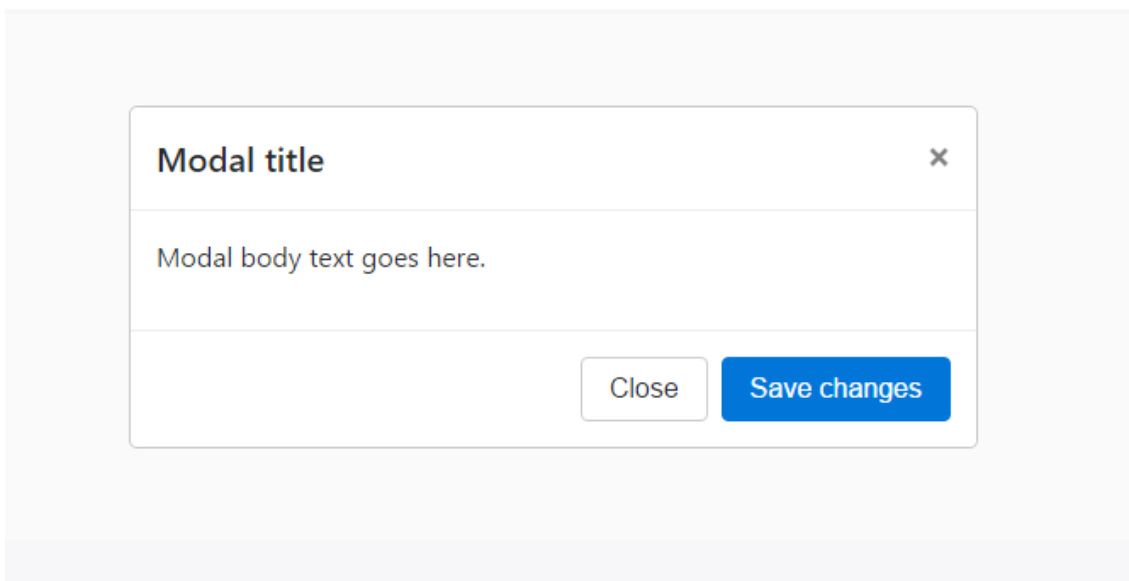
Pärast uudise lugemist suunas kasutaja uudistevoosse kõige paremini tagasi Delfi, kus lugeja jõudis põhimõtteliselt samasse kohta, kust ta uudisele klikates ära liikus. Postimehel ning Õhtulehel jäi sellest funktsionaalsusest vajaka: mõlemad veebilehed suunasid kasutaja mõnikord valitud uudisest ette, mõnikord taha poole.

Androidi rakendused olid välimuselt ja ülesehituselt, kui imeväikseid detaile mitte arvestada, ühesugused. Postimehel oli mugav nupuke, mis suunas kasutaja sõltuvalt tema asukohast kas lehe lõppu või algusesse. Erinevuseks oli veel see, et Õhtulehe rakendus jättis endast tunduvalt aeglasema mulje, sest kasutajale oli näha edasi-tagasi liikuv

laadimist näitav jooneke ning reklaamide ära laadimisel artiklite sisu „hüppas“, st sisu liikus äkiliselt näiteks alla- või ülespoole, et mahutada ära reklaam.

3 Modaale kasutavate veebilehede analüüs

Modaal on teise sõnaga hüplikaken, mida kuvatakse lahti oleva veebilehe peal. Klõkkides modaali taustale ehk tumendatud veebilehe osale või sulgemisnupule see suletakse. Modaale ehitatakse HTML-i, Javascripti ja CSS-ga [9]. Järgnevalt on toodud joonis ühest näidismodaalist.



Joonis 8. Ekraanitõmmis Bootstrapi veebilehel olevast modaali näitest.

Enamjaolt kasutavad veebilehed modaale, et informeerida kasutajat millestki või et paluda kasutajal midagi teha, kas siis näiteks tellida ajalehte, täita vormi või anda teada, et toimunud on mingi uuendus. Modaal on mugav komponent, kui soovitakse, et kasutaja teeks ära mingisuguse tegevuse ilma, et ta hetkel käesolevast tegevusvoost välja toodaks.

Käesolevas peatükis hinnatakse kolme erinevat modaali kolmel veebilehel ja tuuakse välja nende kasutuse, välimuse ja ülesehituse positiivsed ja negatiivsed küljed.

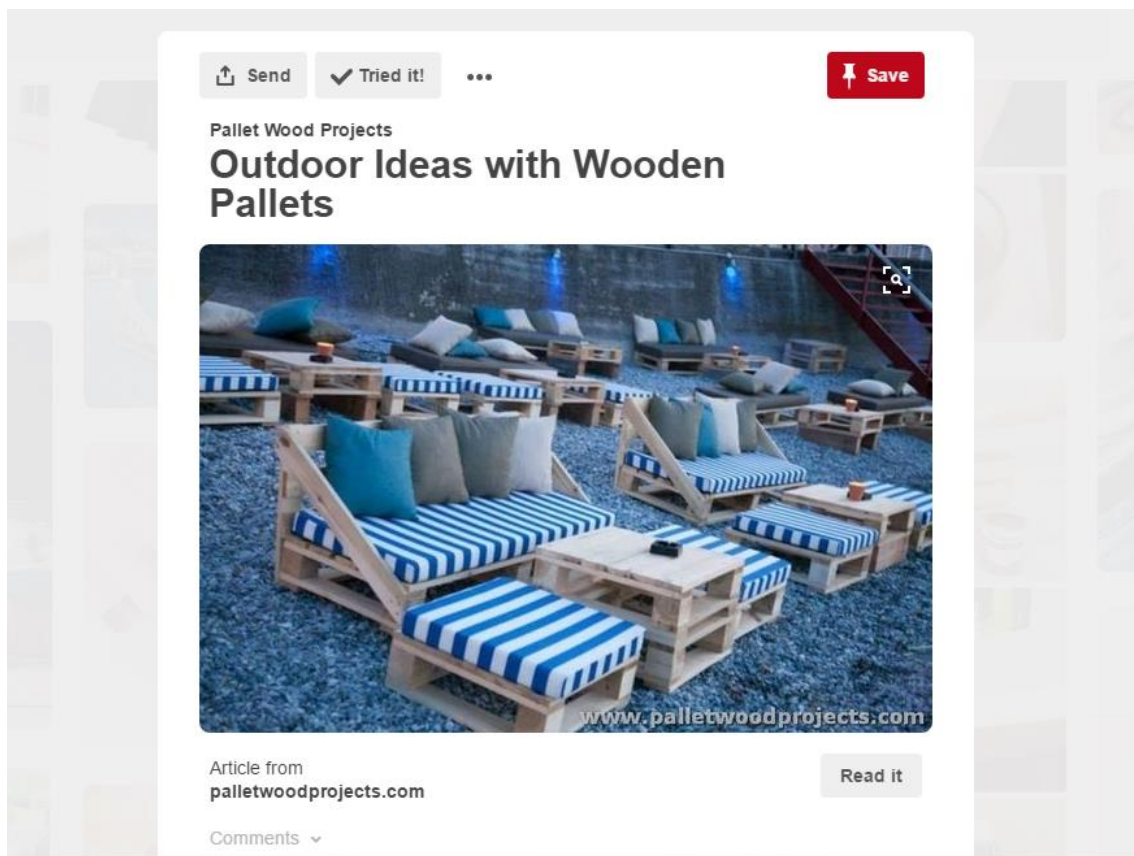
3.1 Pinterest

Pinterest kujutab endast ühte maailma populaarsemat *DIY* ideede jagamise keskkonda. Selle ülesehitus on vägagi lihtne: kasutaja valib teemad, mis teda huvitaksid, näiteks fotograafia või kokandus ning seejärel hakatakse kuvama selle sildiga *pin*'e ehk väiksed *DIY* õpetusi tema seinal. *Pin'ile* klõkkides avatakse see lähivaate jaoks modaalne. Edasise

huvi korral on võimalus modaali peal klikata, mis siis viib originaalse postituseni mõnes blogis [10].

Sellise ülesehituse peamiseks miinuseks on see, et mitte kuskil ei kuvata kasutajale, et modaalile klikates jõutakse algse postituseni. Sellest funktsionaalsusest saavad kasutajad üldiselt teada alles siis, kui nad eksikombel modaalile vajutavad. Samuti tundub modaal üsna suur, alla liikudes pole eraldi sulgemise nuppu, nooltega lehel liikumine ei tööta ja vajutades mujale lehel kaob modaal eest ära.

Positiivse külje pealt aga on mugav, et ei pea iga *pin*'iga tutvumiseks liikuma uuele vahelehele, vaid on võimalik täpsemalt uurida ilma voost lahkumata. Kui *pin* meeldib, saab seda nõ „salvestada“, et seejärel hiljem uuesti vaadata ja kasutada. Leht näeb välja minimaalne ja puhas, lihtne on keskenduda sellele, mida lähemalt uurida sooviti. Samuti pole modaal infoga üleküllastatud. Siin on kujutatud modaali kasutust Pinterest lehel:



Joonis 9. Ekraanitõmmis veebilehe Pinterest modaalist.

3.2 Basket

Eesti veebileht Basket on pühendatud korvpallile. Sellel lehel kuvatakse erinevaid artikleid, liigade ajakavasid ja tulemusi ning paljut muud. Modaal kasutab veebileht tuleva ürituse reklaamimiseks.

Modaal on oma kujunduselt sarnane ülejäänud veebilehele, mis jätab mulje, et see on osa lehest ja ei tekita võõrkeha tunnet. Samuti on see lühike ja annab konkreetselt edasi vajalikku informatsiooni.

Disainiveaks võiks pidada seda, et kuna tegemist on reklaamiga, võiks see mujale veebilehel klikkides eest kaduma, antud modaal aga seda ei tee. Lisaks on modaal sulgemisristike modaalist endast kõrgemal ning halvasti nähtav.

Kasutaja vaatenurgast on tüütu, kui modaale kasutatakse reklaami näitamiseks, sest üldjuhu kuvatakse sellisel juhul modaali ilma kasutajapoolse soovita, näiteks siis, kui leht on laaditud või siis, kui kasutaja kerib lehte. Järgnev joonisel olev modaal sai kasutajale kuvatud pärast lehe laadimist.

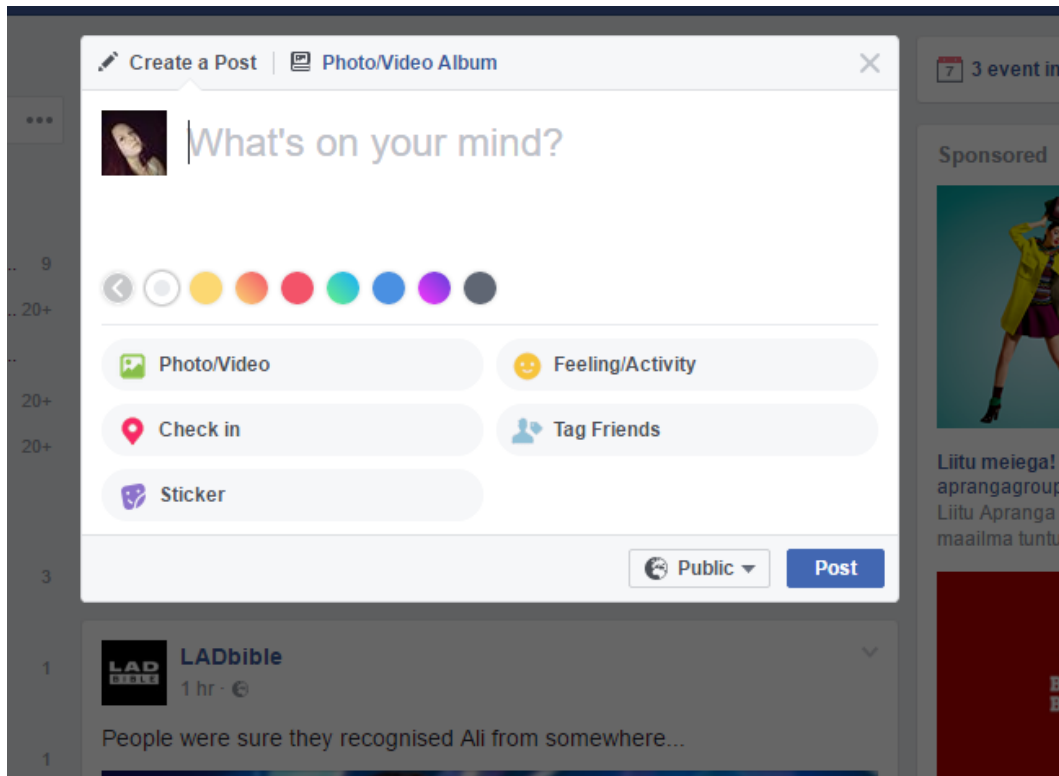


Joonis 10. Ekraanitõmmis veebilehe Basket modaalist.

3.3 Facebook

Facebook on maailma populaarseim suhtlus- ning tänapäeval ka reklaamindusvõrgustik. Seda kasutavad miljonid inimesed igapäevaselt toodete ostmiseks või müümiseks, uudiste lugemiseks, ürituste loomiseks ja paljuks muuks.

Facebook kasutab postituse loomise faasis modaali. See on väike ja konkreetne, selgesti eristatavate osadega: tekstiala, kuhu kirjutada oma postitus, mille puhul saab valida tekstivärvi, võimalik on lisada pilte, videosid, kleepsukesi, tegevust. Samuti saab valida, kas postitust kuvatakse avalikult, sõpradele või iseendale. Aknal on samasugune kujundus, kui ülejäänud Facebook'i lehel, mis on nähtav järgmisel joonisel.



Joonis 11. Ekraanitõmmis Facebooki lehel kasutusel olevast modaalist.

Modaali negatiivseid külgi on raske leida. See on lihtne, pole üleliigseid komponente. Modaal sulgub, kui klikata lehel, selle loomise ja kaotamise animatsioon on lühike.

3.4 Kokkuvõte

Antud peatükis uuriti lähemalt kolme veebilehte, mis kasutavad modaale. Valitud portaalid kasutasid neid üsna erinevateks eesmärkideks, näiteks reklaami kuvamiseks, postituse loomiseks ja kuvamiseks.

Kokkuvõtvalt võib öelda, et modaalide kasutus oli iga lehe puhul õigustatud alternatiiv kuva täielikule vahetusele, sest nii hoiti kasutajat kindlas voos ning ei tekitatud

ebamugavat olukorda, kus valiku või tegevuse tühistamisel oleks kasutaja sattunud kuhugi, kuhu ta liikuda ei soovinud.

Samuti saab öelda, et modaal on kasutajasõbralik ja mugav disainielement sellisel juhul, kui see vastab oma kujunduse poolest ülejäänud veebilehega, on lühike, ei kuva liiga palju informatsiooni, selle sulgemise võimalused on selgelt märgistatud, sellele klikates juhtub kasutaja jaoks oodatud tulemus ning veebilehel mujale klikates see sulgub.

4 Vahekokkuvõtte analüüsis

Selles peatükis võetakse kokku uudisteportaalide ja modaale kasutavate veebilehede tugevad küljed, et oleks selge pilt ees, millega tuleks liidese realisatsioonil arvestada.

Uudisteportaalide ja nutirakenduste positiivse külje pealt võib öelda, et kõik uuritavad digiväljaanded olid loodud ühe, kasutajale üsna mugava, malliga: *one-pager* tüüpi avaleht, mille päises oli menüüriba. See on mugav, sest nii ei pea kasutaja hakkama läbi ketrama erinevaid vahelehti menüüs, et saada üleüldist pilti uudistest. Samuti olid kõigil portaalidel eraldi teemalehed, kus kuvati vastava kategooria uudiseid samamoodi *oner-pager* stiilis. Lisaks olid lehed lihtsad: kasutati paari värvi, enamasti valget ja mingit muud, fondid olid kergesti loetavad.

Modaalide kasutuses oli positiivne see, et kasutajale jäi tema voog alles. See oli väga hästi illustreeritud just Pinteresti näite puhul, kus kasutajale kuvati ühel lehel mitmeid erinevaid õpetusi ning õpetusele klikates tutvustati seda just modaali kujul. Samuti mõjusid modaalid hästi siis, kui nende kujundus sobis kokku ülejäänud lehega, see tähendab, et nad ei jätnud endast võõrkeha tunnet ning siis, kui kasutajale oli selgesti ära märgitud, kust on võimalik modaali sulgeda.

Kokkuvõtvalt võiks öelda, et liidese realisatsioonil tuleks meeles pidada järgmisi aspekte: leht peaks olema minimaalne, kergesti loetav, kasutajale intuitiivne, mitte liiga värviline. Modaalid peaksid sobima kokku ülejäänud veebilehega, olema samuti üsnagi minimaalsed oma välimuselt, kergesti sulgetavad ning avanema vaid siis, kui kasutaja seda soovib ehk siis, kui ta selleks kusagile klikib.

5 Liidese realisatsiooniks vajaminevate tehnoloogiate võrdlus

Käesolevas peatükis analüüsitakse erinevaid tehnoloogiaid, et valida sobilikumad liidese realisatsiooniks.

5.1 Uudiste ning nendega seonduva pärimine

Kasutajaliidese arenduseks on vaja pärida sellele sisu. Paljud, eelkõige välismaised uudisteportaalid omavad API-t, mille kaudu on arendajatel võimalik pärida portaali andmebaasist erinevat infot, näiteks uudiste pealkirju, pilte, autoreid. Selline võimekus on olemas teiste seas NY Times'i veebilehel [11] või CNN-l [12]. Antud töös fookusesse võetud Postimehel aga selline vahend puudub, seetõttu oli vaja leida mõni muu viis, kuidas sama info kätte saada.

Uudisteportaalidel on ka *RSS-feed*, mis on infovahetusstandard, mis lubab välja jagada veebilehelülgedele sisu. Postimehe *RSS-feed* annab välja vaid uudiste pealkirjad, pildid ja lingi sisule. Sellest antud töö realiseerimisel ei piisa.

Sõelale jäi seega tehnika nimega *web scraping*, mis kujutab endast vajaliku info hankimist veebilehe staatilisest HTML-koodist [4]. Selleks on veebis palju erinevaid tööriistu. Järgnevalt on välja toodud 3 populaarsemat.

5.1.1 PHP Simple HTML Dom Parser

Simple HTML Dom Parser on PHP keelele loodud teek, mis võimaldab pärida HTML-koodist vajalikku infot. Sarnaselt jQuery'le on võimalik selles teegis kasutada selektoreid, et leida HTML-i elemente [13]. Nõueteks on, et PHP versioon oleks 5 või kõrgem. Teek on lihtsasti kasutatav, piisavalt dokumenteeritud ning suure populaarsuse tõttu on selle kohta internetis küllaldaselt informatsiooni.

5.1.2 Scrapy

Scrapy näol on tegemist Pythoni keelele loodud *open-source* raamistikuga, mida saab kasutada vägagi erinevates rakendustes, nagu näiteks andmete kaevandamine, info

töötlemine või arhiveerimine [14]. Sellel on olemas põhjalik dokumentatsioon ning suur kasutajaskond.

5.1.3 BeautifulSoup

BeautifulSoup on taaskord Pythoni keelele kirjutatud teek, mille motoks on „Sina ei loonud seda halba veebilehte. Sina üritad sellest vaid andmeid kätte saada. BeautifulSoup on siin, et aidata. Alates 2004. aastast on see hoidnud arendajatele kokku tundide või päevade võrra tööd.“ [15] . BeautifulSoupi eeliseks on see, et ta konverteerib automaatselt sissetulevad andmed *Unicode*'i ja väljaminevad andmed UTF-8 formaati. Samuti kasutab see populaarseid Pythoni teeke *lxml* ja *html5lib*, mis võimaldab proovida läbi erinevaid andmekogumismeetodeid ning leida endale sobivaim.

5.2 Liidese välimus

Kui uudiste sisu on leitud, on vaja seda ka kuvada hea välimusega liideses. Et mitte kombineerida üleliigselt erinevaid raamistikke ning tekitada ebavajalikkku keerulisust, on vaja CSS-i raamistiku valimisel arvestada ka selle võimekusega modaalide loomisel. Järgnevalt on välja toodud populaarsemad CSS ja Javascript raamistikud, mis võimaldavad ka modaalide kasutust.

5.2.1 Bootstrap

Bootstrap on kõige populaarsem HTML-i, Javascripti ja CSS-i raamistik loomaks seadmele reageerivaid nutiseadmed-ennekõike veebilehti [16]. Bootstrapil on väga palju erinevaid komponente, mida saab hõlpsasti oma veebilehel ära kasutada, nende hulgas ka selleks tööks vajalik modaal. Bootstrap on tasuta kasutamiseks.

5.2.2 W3.CSS

W3.CSS on kaasaegne raamistik, mille eelisteks on selle väiksus, lihtne kasutatavus, CSS-i tugi kõigile rohkemkasutatavatele veebibrauseritele ja seadmetele [17]. W3.CSS on samuti tasuta. Ülesehituselt ja sisult on Bootstrap ja W3.CSS üsnagi sarnased, kasutades näiteks ühte peamist konteinerit, kuhu sisse järjest elemente lisatakse.

5.2.3 Bulma

Bulma on neist kolmest kõige vahem tuntud CSS raamistik. See on ehitatud kasutades *flexboxi*, mis on CSS3-s tutvustatud uus kujundusviis. Sarnaselt eelnevale kahele koosneb

ka Bulma erinevatest komponentidest ning on responsiivne, st kohaneb vastavalt seadme ekraani suurusele. Lisaks on Bulma *open-source* ja tasuta [18].

5.2.4 AngularJS

„AngularJS on see, mis oleks HTML, kui see oleks mõeldud veebirakenduste ehitamiseks.“ AngularJS on väga populaarne ja võimas Javascripti *framework* ehk raamistik. AngularJS laiendab HTML-i atribuute direktiividega ja seob andmed HTML-i külge avaldistega. Kuna raamistik on loodud vaid Javascripti kasutades, on sellega loodud kood kergesti testitav, hallatav, refaktoreeritav [19].

AngularJS suureks boonuseks on UI Bootstrap – Bootstrapi komponendid kirjutatuna AngularJS-i raamistikku kasutades, mis muudab nende dünaamilise loomise lihtsaks ja mugavaks: vaja on luua mall ehk komponendi HTML-i põhi ja sellele *controller*, millega mall näiteks nupu vajutuse peale luuakse.

5.3 Kokkuvõte

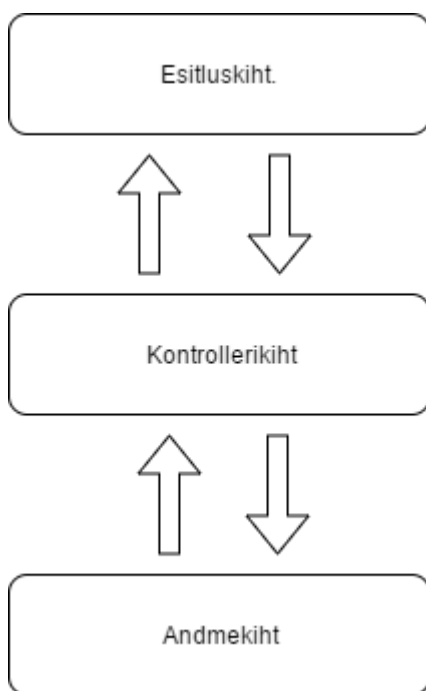
Antud peatükis sai käsitletud erinevaid tehnoloogiaid, mida kasutada prototüübi realiseerimisel. Nimistust said välja valitud järgmised tehnoloogiad: uudiste sisu lugemiseks `simple_html_dom` PHP keeles, AngularJS modaalide dünaamiliseks loomiseks ja sisustamiseks ning Bootstrap kasutajaliidese välimuse korrigeerimiseks.

Tehnoloogiate valikul sai määravaks eelkõige autori varasemad kogemused ja tuttavus erinevate raamistikega.

6 Kasutajaliidese realiseerimine

Käesolevas peatükis kirjeldatakse kasutajaliidese realiseerimise samme. Portaal, millele modaalidel põhinevat vaadet luuakse, on Postimees. See uudisteportaal sai valitud eelkõige seetõttu, et tegemist on populaarseima eestikeelse digiväljaandega, mida külastavad kuus sajad tuhanded inimesed [7].

Veebilehe ülesehitus on jagatud kolmeks suuremaks osaks: esitluskiht, kontrolleriikiht ja andmekiht. Esitluskiht on see osa, mida kuvatakse kasutajale ning millega kasutaja suhtleb. Kasutaja sisendi peale (näiteks nupuvajutuse) pöördub esitluskiht kontrolleriikihi poole, kus tuvastatakse, mida kasutaja soovib, et tehtaks. Selle tulemusena võib kontrolleriikiht saata esitluskihile uusi andmeid, mida kuvada või pöörduda andmekihi poole mingite sisenditega, et pärida uut infot. Andmekihis kogutakse kokku kontrolleriikihi poolt soovitud info ning tagastatakse see kontrolleriikihile, mis omakorda saadab andmed edasi esitluskihti kasutajale kasutamiseks. Üldist ülesehitust illustreerib järgmine joonis.



Joonis 12. Kasutajaliidese ülesehitus

6.1 Liidese ülesehitus

Liidese avalehel kuvatakse uudiste pilt ning pealkiri üsnagi sarnaselt hetkel kasutuses olevatele uudisteportaalide avalehtedele *one-pager* stiilis. Lehe alguses on värskemad uudised ning alla kerides kuvatakse järjest vanemaid uudiseid.

Uudise pealkirjale või pildile klikates avaneb modaal, milles on näha taaskord uudise pealkiri ja foto, nende all aga uudise sisu, milleks võib olla video, pildid või reaalne artikkel.

Antud töös Postimehe teemalehti ei realiseerita, küll aga oleksid need samasuguse ülesehitusega: *one-pager* stiilis leht, millel kuvatavad uudised avaneksid modaalis. Teemalehele saaks kasutaja liikuda Postimehe avalehel asuvast menüüribast, mida samuti hetkel ei realiseerita.

6.2 Uudiste sisu pärimine

Kuna uudistevoost polnud loodava liidese jaoks võimalik ei API ega *RSS-feed*'i kaudu uudiste sisu pärida, pidi selleks kasutama tehnikat nimega *web scraping* ning looma ise parsija, mis oleks võimeline koguma vajalikke andmeid.

Uudiste sisu pärimiseks sai valitud PHP keel ning teek `simple_html_dom`. Et eelmainitud teeki kasutada, on vaja alla laadida `simple_html_dom.php` fail ning see oma PHP failis kaasata.

Avalehe sisu leidmiseks sai loodud PHP fail nimega `parser`. Kuna HTML koodis on loodud mitmed `<div>` elemendid, mis omakorda sisaldavad uuesti `<div>` elemente, siis leiab `parser.php` URL-ilt <https://www.postimees.ee> kõik artiklit sisaldavad HTML elemendid. Seejärel käiakse leitud elemendid tsükliga läbi ning päritakse sellest uudise pealkiri, pilt ning uudise URL. Saadud info salvestatakse sõnastikku, mis omakorda lisatakse listi. Nii tekib on listil valiidne JSON-i kuju, mis muudab selle kasutuse Javascriptis lihtsamaks. Kui list on täidetud, tagastatakse see Javascripti kihile.

Uudise sisu pärimiseks on loodud PHP fail nimega `contentParser`. Selle failile annab Javascripti kiht ette uudise URL-i, millelt `contentParser` seejärel leiab üles uudise pealkirja, pildi, pildi autori, ja artikli sisu, milles võivad omakorda olla pildid, nimekirjad, videod vms. Kui vajalik info kätte saadud, tagastatakse see taaskord Javascripti kihile.

Eraldi loogika oli vaja luua www.ilmajaam.postimees.ee uudistele, sest nende HTML-i ülesehitus erines kardinaalselt ülejäänud Postimehe vahelehtede ülesehitusest.

Algselt oli mõeldud, et töös kuvataks ka galeriid ning kommentaare. Kuna aga need komponendid luuakse Postimehe lehele Javascripti kasutades dünaamiliselt, pole neid võimalik staatilise HTML-i lehe pealt lugeda, mistõttu polnud võimalik koguda ei galerii ega ka kommentaaride sisu.

6.3 Uudiste kuvamine avalehel ja modaalis

Uudiste kuvamiseks veebilehel teeb Javascript AJAX-i *get* päringu `parser.php` failile. See tagastab JSON formaadis Javascriptile vajaliku info. Andmed lisatakse veebilehele kasutades AngularJS-i. See tegevus on kuvatud järgmisel joonisel.

```
$http.get('parser.php').success(function(data) {  
    this.data = data;  
    $scope.articles = data;  
});
```

Joonis 13. Ekraanitõmmis Javascripti failis kasutatud *get* päringust

Uudise sisu kuvamiseks modaalis teeb Javascript aga AJAX-i *post* päringu `contentParser.php` failile. *Post* päringuga, mida on kuvatud ka järgmisel joonisel, annab Javascript PHP-le ette artikli URL-i (näiteks <http://www.postimees.ee/4113071/riik-ehitab-8-miljoni-euroga-riisiperest-turbani-raudteeloigu-mida-utles-tasuvusuuring>) ja pildi URL-i, mida kasutada, kui artiklil endal pilt puudub (näiteks juhul, kui pildi asemel kuvatakse artikli sees hoopiski galeriid) ning PHP fail pärib seejärel sellelt lingilt vajamineva info ning tagastab selle Javascriptile. Andmeid kuvatakse modaalis kasutades AngularJS-i võimekust.

```

$http({
  method: "POST",
  url: "contentParser.php",
  data: {
    data: data[index].url,
    pic: data[index].img
  }
}).success(function(response, $sce) {
  $log.log(response);
  heading = response[0].headline;
  pic = response[0].img;
  lead = response[0].lead;
  text = response[0].text;
  video = response[0].video;
  imgSnippet = response[0].imgSnippet;
  authorPic = response[0].authorPic;
  authorName = response[0].authorName;
});

```

Joonis 14. Ekraanitõmmis töös kasutatud post päringust

6.4 Avalehe ja modaali ülesehitus

Kasutajaliidese avaleht on lihtne HTML-i dokument nimega index.html, kus kasutades AngularJS-i ng-repeat direktiivi luuakse iga uudise kohta <div> element, mille sees on artikli pealkiri ja pilt. Need elemendid on lehe laadimise ajal peidetud kujul, et kasutajale ei paistaks välja uudiste paigutamise virr-varr.

```

<div ng-controller="showNewsCtrl" id="articleList">
  <div class="well" id="well" style="display: none" ng-repeat="article in articles track by $index">
    <div class="item">
      <a href ng-click="showModal($index)" class="article article-text"> {{ article.headline }}</span> </a>
    </div>
  </div>
</div>

```

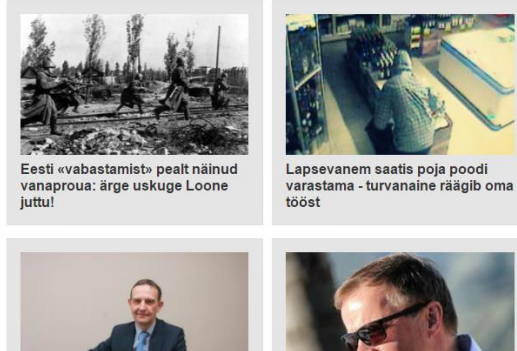
Joonis 15. Ekraanitõmmis ng-repeat direktiivist index.html failis

Lisaks sai avalehele loodud kaks nuppu, „üles“ ja „värskenda“. Esimene neist suunab kasutaja lehe tippu, mida on mugav kasutada siis, kui kasutaja on jõudnud uudiste lugemisega lehe lõppu. Teine värskendab lehte ja viib kasutaja tagasi lehe algusesse, et ta saaks tutvuda värskete uudistega. Avaleht on kujutatud järgmisel joonisel.

Tähelepanu! Tegu on Tallinna Tehnikaülikooli bakalaureuse lõputöö projektiga. See veebileht pole kasutusel ärilistel eesmärkidel.

Postimees

Alati esimene



Joonis 16. Ekraanitõmmis loodud kasutajaliidese avalehest.

Modaali kuvamine on aga keerukam. Et AngularJS kasutades saaks dünaamiliselt luua modaale, on esmalt vaja luua modaali HTML mall. See fail sai nimeks modal.html. Faili sisuks on HTMLi dokument koos AngularJS direktiividega, mis võimaldavad modaali dünaamiliselt täita andmetega.

Modaali loomiseks kasutatakse AngularJS funktsiooni, mis võtab sisendiks erinevad väärtused, nagu näiteks modaali suurus või mall ning loomulikult ka andmed, mida modaalil kuvada. See tegevus on välja toodud järgmisel joonisel.

```

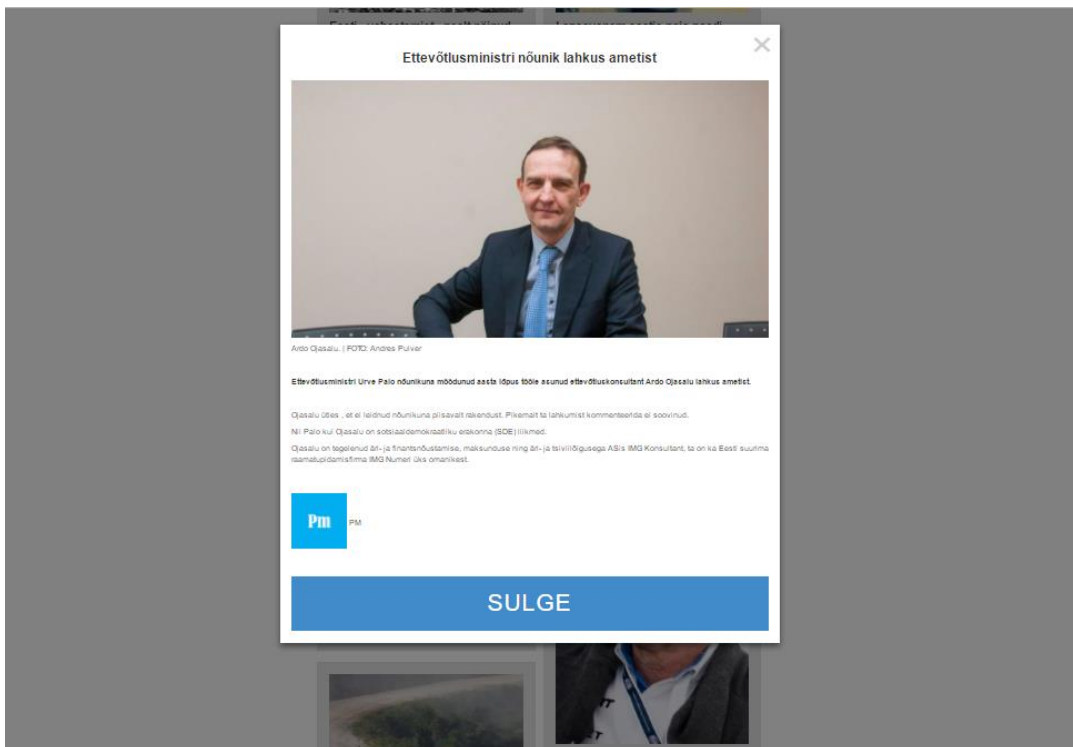
$scope.opts = {
  backdrop: true,
  backdropClick: true,
  dialogFade: false,
  size: 'lg',
  keyboard: true,
  templateUrl : 'modal.html',
  controller : ModalInstanceCtrl,
  resolve: {} // empty storage
};

$scope.opts.resolve.item = function() {
  return angular.copy({
    headline: heading,
    text: text,
    img: pic,
    imgSnippet : imgSnippet,
    video: video,
    authorPic: authorPic,
    authorName: authorName,
    lead: lead });
}

```

Joonis 17. Modaaliloomise ja sisuga täitmise AngularJS-i koodi ekraanitõmmis

Modaalile sai lisatud ka kaks sulgemisnuppu: üks on ristikese kujul pealkirja kõrval ning teine on suure nupuna kujutatud artikli lõpus. Nii ei pea kasutaja kerima artikli algusesse või lõppu, kui ta antud uudist mingil põhjusel lugeda ei soovi. Modaal on kujutatud järgmisel joonisel.



Joonis 18. Ekraanitõmmis loodud modaalist.

6.5 Reklaamide kuvamine loodud liideses

Kuna reklaamid on uudisteportaalidele ning blogidele peamiseks tuluallikaks, on oluline, et nende veebilehtedele loodud kasutajaliides võimaldaks neid lugejatele kuvada. Samas tuleks arvestada ka kasutajatega: reklaamid peaksid olema nähtaval, kuid mitte pealetükkivad ja segavad.

Antud töös loodud kasutajaliideses on reklaamide näitamiseks mitmeid erinevaid võimalusi: neid saab kuvada nii avalehe külgedel, nagu on näidatud järgmises joonises, kui ka jaluses või päises, samuti on reklaame võimalik lisada uudiste vahele.



Joonis 19. Üks reklaamide kuvamise võimalusi loodud kasutajaliideses.

Lisaks on reklaame võimalik kuvada modaalis, kus üheks lahenduseks oleks rakendada sama loogikat, mida kasutatakse nutiseadmetele mõeldud rakendustes, ehk näidata reklaame artikli sees. Seda illustreerib järgmine joonis.



Rallilegend Tommi Mäkinen veab nüüd Toyota võistkonda. | FOTO: @World/Toyota

REKLAAM

Autoralli neljakordse maailmameistri, praeguse Toyota Gazoo Racingi meeskonna juhi Tommi Mäkineni sõnul on tema tiimi pikemaajaline eesmärk tuua WRC-sarja mõni noor Jaapani võidusõitja. Lühemas plaanis on soomlane aga huvitatud lepingust Ott Tänakuga.

REKLAAM

Tommi, su nägu on pidevalt naerul. Oskasid sa seda hooaja eel oodata, et sul on nii palju põhjuseid rõõmustamiseks ja et auto kukkus nõnda hästi välja?

(Lõkerdab naerda.) Muidugi oli hooaja algus lihtsalt oivaline. Viimastel rallidel oleme küll veidi teistest tiimidest maha jäänud, aga saan endale siiski naeratamist lubada, sest tean, mida tegema peame. Me ei süga kukalt, et mis nüüd lahti, ega muretse, kuidas teistele järgi jõuda. Meil on selged plaanid, kuidas autot kiiremaks saada ja seetõttu ma tulevikku ei pelga.

REKLAAM



Viljar Voog Porto

Joonis 20. Üks modaalis reklaamide kuvamise lahendus.

7 Loodud liidese analüüs

Selles peatükis hinnatakse realiseeritud kasutajaliidest ning tuuakse välja selle head küljed ning selle, mis oleks võinud teha paremini.

Positiivse külje pealt saab liidese puhul välja tuua kasutusmugavuse. Modaal laadib kiiresti ning avaneb avalehe peal, mistõttu ei lähe kaotsi järgi uudistevoos ja samas brauseris ei ole avatud mitmeid aknaid. Modaaali sulgemiseks on üks nupp pealkirja juures, teine artikli lõpus, samuti sulgeb modaal, kui klikkida avalehe tumendatud osal. Nii ei pea lugeja kerima modaaali algusesse tagasi, et seda sulgeda. Lisaks on loodud kaks nuppu, millest esimene viib lugeja tagasi uudistevoos tippu ning teine värskendab uudistevoogu. Samuti on jäetud palju ruumi reklaamide kuvamiseks, mis on uudisteportaalidele oluline.

Negatiivse külje pealt aga laeb avaleht umbes kolm sekundit, sest kõik uudised on vaja Postimehe enda lehelt esmalt lugeda ning siis kuvada loodud avalehel. Samuti ei kuvata loodud liideses galeriisid ega kommentaare, sest need lisatakse Postimehe veebilehele Javascriptiga dünaamiliselt, mistõttu ei saa neid staatilisest HTML-st välja lugeda. Lisaks ei ole realiseeritud menüüriba, kust lugeja saaks liikuda end huvitava teemalehe artiklite juurde; see jäi antud töö skoobist välja. Eelmainitud vead on prototüübile omased, mis aga reaalselt probleemiks poleks, kui liides oleks loodud Postimehe enda poolt, sest sisu annaksid nad ise ette ja seetõttu jääks ära nii parsimisele kuuluva ajaga seotud vead kui ka hetkel prototüübist puuduolevad kommentaarid ja galeriid.

8 Kokkuvõte

Antud töö eesmärk oli luua ühele uudisteportaalile modaalist põhinev kasutajaliides. Selle jaoks sai analüüsitud kolme populaarsemat eestikeelselt digiuudisteväljaannet ja nende rakendusi Androidi nutiseadmele ning uuritud nende tugevusi ja nõrkusi. Lisaks tutvuti lähemalt erinevate tehnoloogiatega, mis võimaldaksid kirjeldatud kasutajaliidest luua. Tehnoloogiaid uuriti nii serveri- kui ka kliendipoolseid ning tehti valik.

Uudisteportaalide analüüsist selgus, et kuigi lehtede välimus ja ülesehitus oli ilus, kasutajale mugav ja kergesti arusaadav, siis üleliigne reklaamide kogus muutis lehe kohmakaks. Samuti oli ebamugav see, et uudiste lugemiseks pidi need avama kas uuel vahelehel, risustades sellega oma brauserit, või samal vahelehel ja riskima uudistevoos järje kadumisega. Samuti jäi reklaamide ning uudise uuel vahelehel avamise tõttu tunne, et veebileht on aeglane ja venib.

Käesoleva töö tulemusena on loodud kasutajaliides, mis on nähtav lehel <http://www.loputoo.000webhostapp.com>. Liidese ehituseks kasutati PHP-keelt, AngularJS raamistikku, HTML-i ja CSS-i. Avalehel kuvatakse uudiste pealkiri ja pilt. Nendele klikates avatakse modaal, milles tuuakse kasutajani artikli sisu.

Sellise liidese kasutuselevõtt oleks uudisteportaalidele täiesti reaalne. See on mugavam kasutajale, sest kasutaja järg uudistevoos ei lähe kaduma. Samuti ei ava kasutaja mitmeid ja mitmeid vahelehti brauseris, mida on pärast tüütu sulgeda ning mis muudavad arvuti aeglasemaks.

Uudisteportaaali vaatepunktist ei ole sellist lahendust väga keeruline luua. Lisaks annab ka sellise liidese puhul kuvada kasutajale digiväljaannetele väga olulist tuluallikat ehk reklaami. Seda saaks lisada nii avalehele külgedele, päisesse või jalusesse kui ka modaal näiteks artikli lõikude vahele.

Kasutatud kirjandus

1. *AJAX Introduction*, Refsnes Data. [WWW] https://www.w3schools.com/xml/ajax_intro.asp. (Mai 2017) (veebilehekülg)
2. L. Liikane ja M. Kesa, Arvutisõnastik. [WWW] <http://www.keeleeveeb.ee/dict/speciality/computer/> (Mai 2017) (veebilehekülg)
3. *JSON - Introduction*, Refsnes Data. [WWW] https://www.w3schools.com/js/js_json_intro.asp (Mai 2017) (veebilehekülg)
4. H. Brody, *I Don't Need No Stinking API: Web Scraping For Fun and Profit*, 2017. [WWW] <https://blog.hartleybrody.com/web-scraping/> (Mai 2017) (veebilehekülg)
5. Eesti Ajalehtede Liit, Eesti Ajalehtede Liit, 2017. [WWW] http://www.eall.ee/eall_liikmed.html (Mai 2017) (veebilehekülg)
6. Postimees Grupp, AS Eesti Meedia. [WWW] <http://www.postimeesgrupp.ee/ettevottest/ettevotte-ulevaade> (Mai 2017) (veebilehekülg)
7. gemiusAudience. [WWW] <http://www.audience.gemius.ee/> (Mai 2017) (veebilehekülg)
8. Ekspress Grupp, Ekspress Grupp. [WWW] <http://www.egrupp.ee/> (Mai 2017) (veebilehekülg)
9. Bootstrap. [WWW] <https://v4-alpha.getbootstrap.com/components/modal/> (Mai 2017) (veebilehekülg)
10. Pinterest. [WWW] <https://www.pinterest.com> (Mai 2017) (veebilehekülg)
11. NY Times, The New York Times Company. [WWW] <https://developer.nytimes.com> (Mai 2017) (veebilehekülg)
12. *CNN API Overview*, CNN. [WWW] <https://developer.cnn.com/docs/read/api> (Mai 2017) (veebilehekülg)
13. S. C. Chen, *PHP Simple HTML DOM Parser*. [WWW] <http://simplehtmldom.sourceforge.net/> (Mai 2017) (veebilehekülg)
14. *Scrapy at a glance*, 2016. [WWW] <https://docs.scrapy.org/en/latest/intro/overview.html> (Mai 2017) (veebilehekülg)
15. L. Richardson, *Beautiful Soup*, 2017. [WWW] <https://www.crummy.com/software/BeautifulSoup/> (Mai 2017) (veebilehekülg)
16. Bootstrap. [WWW] <http://getbootstrap.com/> (Mai 2017) (veebilehekülg)
17. *W3.CSS Tutorial*. [WWW] <https://www.w3schools.com/w3css/> (Mai 2017) (veebilehekülg)
18. J. Thomas, Bulma. [WWW] <http://bulma.io/> (Mai 2017) (veebilehekülg)
19. AngularJS. [WWW] <https://angularjs.org/> (Mai 2017) (veebilehekülg)