

Tallinna Ülikool

Digitehnoloogiate instituut

Ülevaade JavaScripti mobiiliraamistikest

Seminaritöö

Autor: Raul Gordejev

Juhendaja: Romil Rõbtšenkov

Autor:””2016

Juhendaja:””2016

Instituudi direktor:””2016

Tallinn 2016

Autorideklaratsioon

Deklareerin, et käesolev seminaritöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

..... (kuupäev) (autor)

Sisukord

Sissejuhatus	4
1 Mobiilirakendused.....	5
1.1 Kohalikud rakendused	5
1.2 Mobiilsed veebirakendused	5
1.3 Hübriid mobiilirakendused	6
2 Mobiiliraamistikud.....	8
2.1 jQuery Mobile	8
2.2 Kendo UI	12
2.3 Ionic	15
2.4 Meteor	17
2.5 Raamistike võrdlus	19
Kokkuvõte.....	22
Kasutatud kirjandus.....	23
LISA 1.....	25

Sissejuhatus

Nutitelefoniid ja tahvelarvutiid on tänapäeval väga populaarseks saanud ja aina rohkem inimesi kasutavad erinevaid nutiseadmeid. Samuti areneb koguaeg edasi ka tehnoloogia ja järjest tuleb juurde uusi võimalusi kuidas ühte asja teha mitmel erineval moel. Võrreldes sellega, mis oli umbes 10 kuni 15 aastat tagasi ja milline oli veeb ning veebiarendus siis, võib öelda, et tehnoloogia on selle koha pealt väga suure sammu edasi läinud. Kui alguses olid veebilehed enamjaolt staatilised ja ehitatud kasutades hüperteksti märginduskeelt (ingl *HyperText Markup Language*) HTML'i, kaskaadlaadistikku (ingl *Cascading Style Sheets*) CSS'i ja lehe dünaamilisuse lisas raster- ja vektorgraafika tegemise programm Adobe Flash¹, siis tänaseks on osa arenduseprotsessist juba arendusmeetodite, raamistike või keelte välja valimine. HTML5 ja CSS3 tulekuga toimus kõige suurem edasimineku. ActionScript'i² asemel on peamine veebi programmeerimiskeel JavaScript, millel on omakorda palju raamistikke ja nende seas nutiseadmetele mõeldud raamistikke, millest siin töös antakse ülevaade. Samuti ei olnud populaarne kohanduv (ingl *responsive*) veebidisain, mida tänapäeval loetakse enesestmõistetavaks ja tähtsaks omaduseks. Tänapäeval on väga vähesed uued lehed mitte-kohanduva disainiga ja populaarseks on muutunud lehtedele ka eraldi mobiilivaate tegemine nii, et mobiilseadmelt lehele minnes suunatakse kasutaja mobiilse-vaatega lehele edasi.

Selle töö eesmärgiks on eelnevalt välja toodud põhjustel luua ülevaade tänapäeval enim kasutatavatest JavaScripti mobiiliraamistikest ja nende erinevustest kui ka sarnasustest. Seminaritöö on mõeldud neile, kellel puudub kokkupuude JavaScript'i mobiilsete raamistikega, ning soovivad populaarsematest raamistikest saada ülevaate. Samuti ka neile, kes soovivad arendada mobiilset rakendust, kuid ei tea mis raamistikku kasutada.

Seminaritöös esimeses peatükis tutvustatakse erinevaid mobiilirakenduste tüüpe. Teises peatükis tuuakse eraldi välja iga valitud raamistiku eripärad ning peatüki lõpus võrreldakse neid omavahel ja antakse lühike kokkuvõte.

Raamistikud, mida siin töös võrreldakse, on välja valitud populaarsuse põhjal välja aastal 2015 (Danny Markov, 2015).

¹ Adobe Flash. Saadaval lehel <http://www.adobe.com/rs/products/flash.html>

² Programmeerimiskeel, mida kasutatakse Adobe Flash programmiga koos. Saadaval lehel <http://www.adobe.com/devnet/actionsript.html>

1 Mobiilirakendused

Mobiilirakendused jagunevad kolme kategooriasse. Nendeks on kohalikud rakendused (ingl *native applications*), mobiilsed veebirakendused (ingl *mobile web applications*) ja hübriid mobiilirakendused (ingl *hybrid applications*). Järgnevates alapeatükkides on täpsemalt välja toodud nende iseärasused.

1.1 Kohalikud rakendused

Kohalikud rakendused ehk mingile kindlale operatsioonisüsteemile eraldi tehtud rakendused on arendatud otse nutiseadmest kasutamiseks. Neid saab alla laadida ja paigaldada rakenduste poest ja olenevalt operatsioonisüsteemist on see erinev. Näiteks Androidil on Google Play³ ja iOS'il App Store⁴. Kohalikud rakendused on mingi kindla operatsioonisüsteemi peale arendatud, mismõttu võib juhtuda, et osasid rakendusi, mis on näiteks saadavad Google Play's, pole App Store's ja vastupidi. See annab väga palju lisafunktsionaalsuse võimalusi rakendusele kui arendada rakendus eraldi igale operatsioonisüsteemile omases programmeerimiskeeles. Näiteks on siis ligipääs seadme kaamerale, GPS'ile, kompassile, kontaktide loendile ja kiirendusmõõturile (ingl *accelerometer*). Kohalikud rakendused saavad ligipääsu kasutatava seadme teadete süsteemile (ingl *notification system*) ja kõige olulisem: rakendusi saab kasutada ka ilma internetita, peale esialgset allalaadimist (Raluca Budi, 2013). Iga operatsioonisüsteemi kohalikud rakendused on arendatud eri programmeerimiskeeltes (vt Tabel 1).

Tabel 1. Kohalikud arenduskeeled eri operatsioonisüsteemidel

Operatsioonisüsteem	Android	iOS	Windows	Linux
Arenduskeel	Java	Objective-C	.NET	C

1.2 Mobiilsed veebirakendused

Mobiilsed veebirakendused ei ole samasugused rakendused, mida saab rakenduste poest alla laadida. Nad on pigem veebilehed, mis on optimeeritud mobiilsest seadmest kasutamiseks ja võivad välimuse poolest väga sarnased kohalike rakendustega olla. Mobiilseid veebirakendusi

³ Google Play store saadaval lehel: <https://play.google.com/store>

⁴ Apple Itunes App Store, saadaval lehel: <http://www.apple.com/itunes/download>

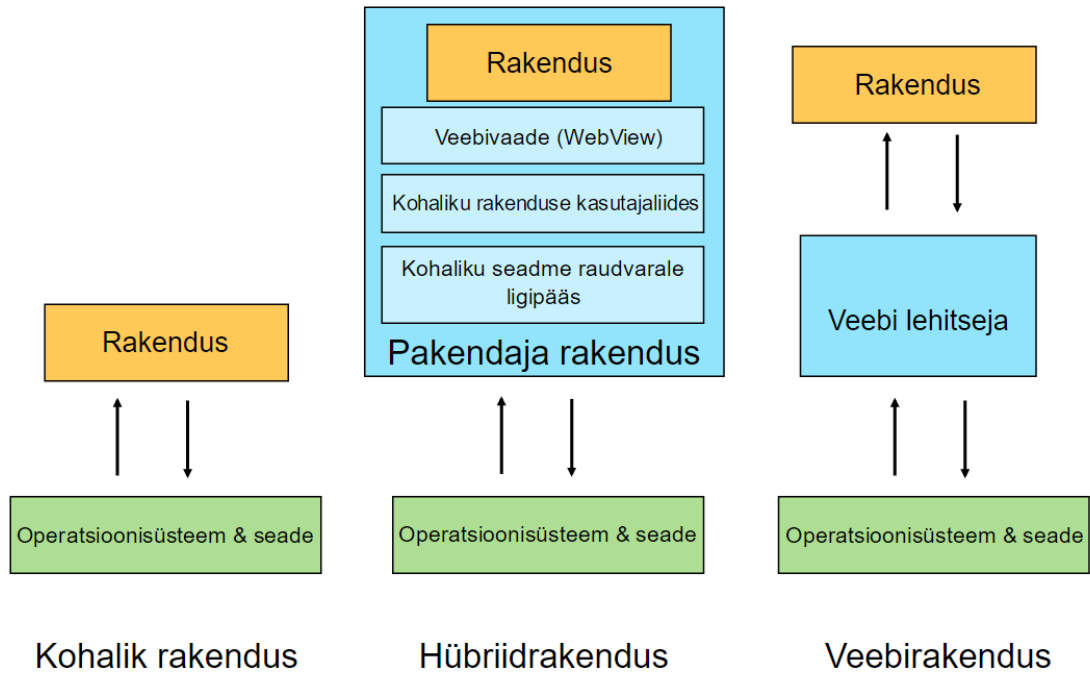
on võimalik vaadata läbi internetilehitseja ja üldiselt on nad kirjutatud JavaScript'is. Sellised rakendused hakkasidki levima HTML5'e tulekuga, kuna sellega tekkis võimalus luua kohalike rakendustega välimuselt sarnaseid mobiilirakendusi ja omada sarnast funktsionaalsust. Nüüdseks, kui enamus veebilehed on HTML5 põhjal kirjutatud, siis pole enam nii suurt vahet märgata osade mobiilsete veebirakenduste ja tavaliste veebilehtede vahel, mis on kohanduva (ingl *responsive*) disainiga ja sobilikud igas seadmes vaatamiseks. Mobiilseid veebirakendusi saab aga peale esimest külastust läbi brauseri „Lisa avakuvale“ (ingl *Add to home screen*) võimaluse salvestada seadme avaekraanile, et edaspidi avada seda lehte nagu päris rakendust (Raluca Budiu, 2013).

1.3 Hübriid mobiilirakendused

Enamus veebiarendajaid valdab hästi selliseid keeli nagu HTML, CSS ja JavaScript. Kuid kohalike mobiilirakenduste loomiseks on vaja teada näiteks Javat, Objective-C'd või .NET'i. Kuid tänapäeval on loodud mitmeid raamistikke, mis sisaldavad erinevaid komponente, millega on võimalik teha välimuselt samasuguseid rakendusi nagu kohalikus keeles saab teha. Väga suur roll on siin erinevatel raamistikel, mis teevad osa tööd arendaja eest ära ja loovad platvormi, kuhu peale arendada edasi oma rakendust. Näiteks on PhoneGap⁵ selline tööriist, mis ühildab veebitehnoloogia ja erinevate operatsioonisüsteemide kohaliku tehnoloogia omavahel. See annab arendajale võimaluse palju rohkem funktsionaalsust lisada oma rakendustesse, tänu sellele, et rakendusel on ligipääs seadmesisestele võimalustele. Hübriid mobiilirakendustel on ligipääs tavalisele veebirakendusele kättesaamatutele funktsioonidele nagu näiteks kiirendusmõõtur, kaamera ja kontaktide loend (vt Joonis 1). Miinuseks hübriid mobiilirakenduste koha pealt loetakse rakenduse kiirust. Kohalikud rakendused on kohati kiiremad kui hübriidrakendused, mistõttu oleks targem otsus ressursinõudlikuma ja turvalisema rakenduse puhul (nt pangarakendus) arendada see ikkagi eraldi igale operatsioonisüsteemile kohalikus keeles. Ka Facebook'i⁶ mobiilirakendus ehitati algselt hübriidrakendusena, kuid Facebooki asutaja Mark Zuckerbergi sõnul oli see suur viga, ning tunnistas, et oleks pidanud kohe tegema kohalikus keeles mobiilirakenduse. Põhjuseks oli rakenduse jõudluse probleem hübriidrakenduses (Dieter Bohn, 2012).

⁵ Rakenduste pakendamise tööriist. Saadaval lehel <http://phonegap.com>

⁶ Facebook. Saadaval lehel <https://www.facebook.com/>



Joonis 1. Mobiilsete rakenduste eritüübid (Globetrotter, 2014)

Üks oluline märksõna hübriid mobiilirakenduste juures on veel veebivaade (ingl *webview*). See tähendab, et mobiili veebilehitseja asemel avatakse hübriidrakendus veebivaate abil kohalikus konteineris, mille vahendusel tekib hübriid rakendusel side nutiseadme võimalustega (John Bristowe, 2015).

2 Mobiiliraamistikud

Hetkel on aktuaalseid ja populaarseid JavaScript'il põhinevaid mobiiliraamistikke väga palju. Antud töös on välja toodud neli populaarsemat raamistikku, milleks on jQuery Mobile⁷, Kendo UI Mobile⁸, Ionic Framework⁹ ja Meteor¹⁰. Järgnevates peatükkides on iseloomustatud neid raamistikke seitsme kriteeriumi põhjal:

- kasutajaliidese mallide hulk (ingl *user interface themes*);
- vidinate hulk (ingl *widgets*);
- töölaua toetus (ingl *desktop support*);
- pakendamise võimalused (ingl *packaging*);
- dokumentatsioon, litsents ja kommuun.

Need kriteeriumid peaksid ainuüksi andma piisavalt hea ülevaate raamistikust, et lugeja suudaks valida endale kõige sobilikuma välja, ning olla teadlik raamistike tugevustest ja nõrkustest.

Selle peatüki lõpus võrreldakse lühidalt neid raamistikke omavahel samadel kriteeriumitel.

2.1 jQuery Mobile

jQuery Mobile on JavaScript'il põhinev raamistik, millega on võimalik luua mitmetel platvormidel töötavaid rakendusi. jQuery Mobile raamistik on loodud 2010 aastal ja hetkel viimane versioon sellest tuli välja 2014 aasta oktoobris. Raamistiku suurus on 351 kb, kokkupakitud versioon 142 kB ning gzip kujul vaid 40 kB. jQuery Mobile vajab töötamiseks ka jQuery't¹¹, mis tuli välja juba aastal 2006.

jQuery Mobile pakub võimalust kujundada erinevaid rakenduse elemente ja testida läbi eri värvilahendusi, mida enda rakenduses kasutada. Selleks on loodud eraldi rakendus

⁷ jQuery Mobile raamistik, saadaval lehel <https://jquerymobile.com>

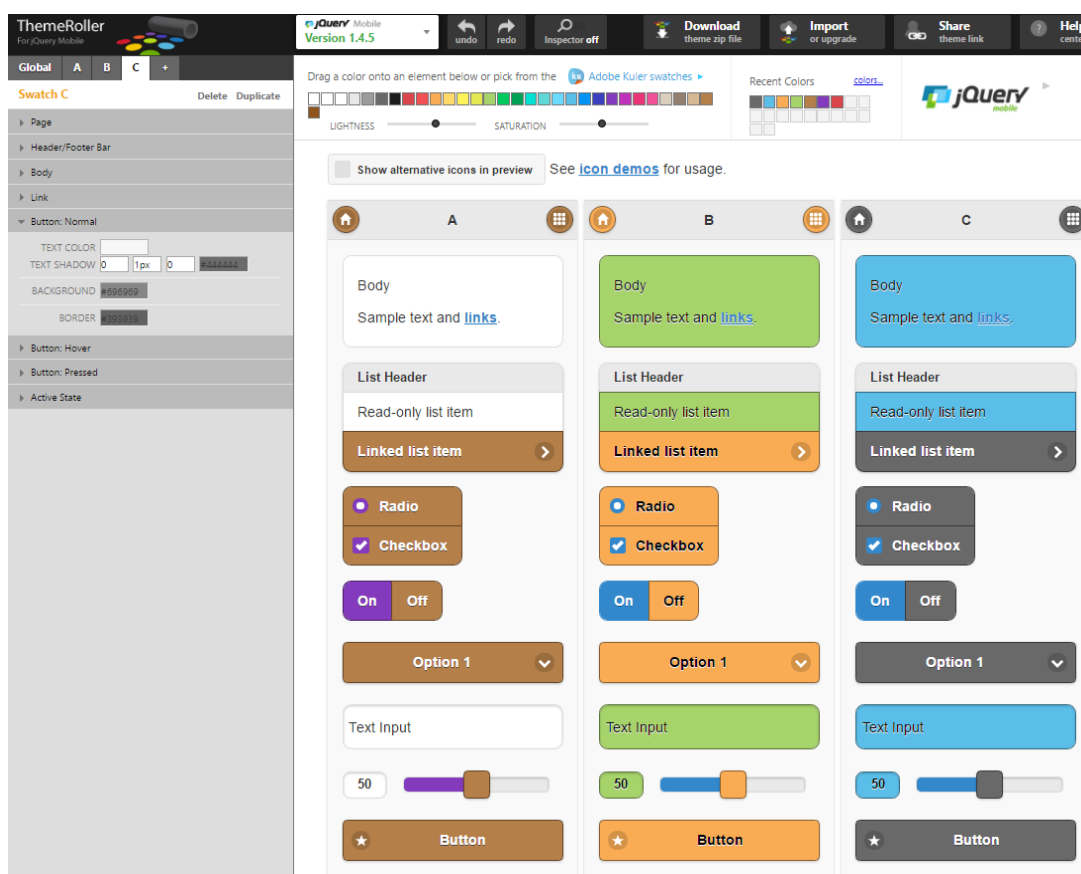
⁸ Kendo UI – JavaScripti mobiilne raamistik. Saadaval lehel <http://www.telerik.com/kendo-ui>

⁹ Ionic on mobiilne JavaScript'i raamistik. Saadaval lehel <http://ionicframework.com>

¹⁰ Meteor on JavaScript'i raamistik, millega on võimalik kiirelt arendada veebirakendusi, mobiilseid rakendusi ning töölaua rakendusi. Meteor on saadaval lehel: <https://www.meteor.com>

¹¹ JavaScript'il põhinev raamistik jQuery. Saadaval lehel <https://jquery.com/>

ThemeRoller¹². Väga mugavaks teeb ThemeRolleri (vt Joonis 2) kasutamine see, et ühe nupuvajutusega on võimalik laadida zip failina kogu kujundatud malli alla, ning kasutada seda enda projektis. On olemas ka importimise ja jagamise nupud. Jagada saab ThemeRolleri loodud malli linki, mida saab lingi saanud kasutaja seda külastades edasi disainida. Jagatud malli hoitakse serveris 30 päeva. Lisaks avaneb võimalus ka rakenduses sees *Help* nupule vajutades leida vastus dokumentatsioonist levinumatele küsimusele, mis võivad tekkida ThemeRollerit kasutades. ThemeRoller'it saab kasutada iga uuema internetilehitsejaga, välja arvatud Internet Explorer kuni versioonini 9 (kaasa arvatud).



Joonis 2. Ekraanitõmmis ThemeRoller'ist

Raamistik sisaldab umbes 300 ikooni, nii SVG kui ka PNG kujul, mis näevad ilusad välja nii ultrakõrglahutusega (ingl *High-Definition*, lühend *HD*) kui tavalahutusega (ingl *Standard-definition* lühend *SD*) ekraanide peal. Platvormid, mis ei toeta SVG ikoone, kasutavad automaatselt PNG kujul ikoone. Veel on raamistikus erinevad CSSi klassid, kus on mitu

¹² jQuery Mobile raamistiku kasutajaliidese disainimise rakendus. Saadaval lehel <http://themeroller.jquerymobile.com>

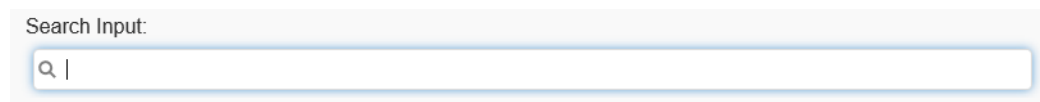
väärtust eelnevalt ära defineeritud. Näiteks *ui-corner-all* klassi kasutamisel mingil HTML elemendil, muutuvad selle kõik nurgad ümaraks. Samuti ka võrgu kavand (ingl *grid layout*) ja kohanduv võrgu kavand (ingl *responsive grid layout*). Lisaks sellele on ka valmis tehtud jQuery Mobile Theme, kus on enamus elemente disainitud ja kust saab enda meeldivuse järgi noppida sobivad välja ja kasutada neid (jQuery Mobile, kuupäev puudub).

Üldiselt jQuery Mobile vidinate kasutamine on väga lihtne. Elemendile vaid id või klassi lisamisel (vt Koodinäide 1) muutub selle välimus ning funktsionaalsus (vt **Error! Reference source not found.**). Klassi *search-mini* lisamisel HTML *input* elemendile tekib suurendusluubi märk elemendi sisse (jQuery Mobile, kuupäev puudub).

```
<label for="search-mini">Search Input:</label>
```

```
<input type="search" name="search-mini" id="search-mini" value="" data-mini="true">
```

Koodinäide 1. Otsingu riba kujundamine jQuery Mobilega



Joonis 3. Otsinguriba peale jQuery Mobile klassi lisamist

Vidinaid on kokku 28 ja igal vidinal on omakorda mitu erinevat versiooni. Näiteks rakenduse päise ja jaluse loomise vidin ning mitmed animeeritud vidinad.

jQuery Mobile on jõuline mobiilirakenduste loomise raamistik, millega on võimalik luua mitmetel platvormidel töötavaid rakendusi luua (ingl *cross-mobile-platform app*). Toetab töölaua internetilehitsejat (nt Google Chrome, Mozilla Firefox, Microsoft Edge ja Internet Explorer), nutiseadmeid ja e-lugereid nagu näiteks Nook või Kindle. jQuery mobile on ennekõike mõeldud puutetundliku ekraaniga seadmete peale arendamiseks (Thoriq Firdaus, kuupäev puudub). Töölaua rakenduste arendamiseks on mõistlikum kasutada jQuery't, sest kui mobiilseid vidinaid töölaua rakenduses kasutada, või vastupidi, ei pruugi tulemus kõige parem olla. Kuna kujundusmallid ja vidinad on mõeldud väiksematele ekraanidele, siis suurema ekraani peal ei ole nad alati proportsioonis.

jQuery Mobile rakendusi saab soovi korral väga lihtsalt pakendada hübriidrakenduseks, kasutades Cordovat¹³. See loob ühenduse kohaliku seadme ja veebirakenduse vahel, mille läbi saab luua rakendusi ühe korraga igale operatsioonisüsteemile (Jorge Ramon, 2015). Cordova on täiesti eraldiseisev platvorm ning ei ole otseselt jQuery ega teiste raamistikega seotud.

jQuery Mobile lehel on olemas väga hea ja põhjalik raamistiku dokumentatsioon. See raamistik põhineb jQuery'l, mistõttu saab ka jQuery dokumentatsioonist õppida seda osaliselt kasutama. jQuery Mobile on koondanud kokku kõik tähtsamad jQuery osad ja on need teisaldanud kokku ühte mobiilile põhinevasse raamistikku. See koosneb erinevatest animatsioonidest, ülemineku efektidest, erinevaid CSS ja HTMLi kujundatud elementides. jQuery Mobile't peetakse lihtsamaks kui teisi JavaScripti mobiiliraamistikke. Sellega on väga lihtne teha ühe lehe rakendusi (ingl kl one-page-application) ehk rakendusi, kus on ainult üks HTML leht, mille sisu on jaotatud eraldi osadeks, mida võib vaadata kui eraldi lehte (vt Koodinäide 2) (Jake Rocheleau, kuupäev teadmata). *Data-role* atribuudiga saab eristada HTML osasi eraldi lehtedeks, millele rakenduse sees saab viidata ja nende lehtede vahel navigeerida, ilma, et peaks tegema mitu erinevat HTML faili.

```
<body>
<!--Esimese lehe algus -->
<div data-role="page" id="foo">

  <div data-role="header">
    <h1>Foo</h1>
  </div> <!--esimese lehe päise blokk -->

  <div data-role="content"> <!--esimese lehe sisu bloki algus -->
    <p>Tekst, mis kuvatakse esimesena, sest see on koodis ees pool.</p>
    <p>Viide teisele lehele <a href="#bar">bar</a></p>
  </div> <!--esimese lehe sisu bloki lõpp -->

  <div data-role="footer">
    <h4>Page Footer</h4>
  </div> <!--esimese lehe jaluse blokk -->
</div> <!--siin lõpeb esimene leht -->

<!--Teise lehe algus -->
<div data-role="page" id="bar">

  <div data-role="header">
    <h1>Bar</h1>
  </div> <!--teise lehe päis -->

  <div data-role="content"> <!--teise lehe sisu bloki algus -->
    <p>Teine leht, mis on koodis all pool kui esimene, seega on alguses peidetud. Tuleb
nähtavale kui esimesel lehel vajutatakse teise lehe viite peale</p>
    <p><a href="#foo">Viide esimesele lehele</a></p>
```

¹³ Apache Cordova – populaarne mobiilsete rakenduste arendamise raamistik, millega on võimalik luua CSS3, HTML5 ja JavaScript'iga mobiilirakendusi. Saadaval lehel <https://cordova.apache.org>

```
</div> <!--teise lehe sisu bloki lõpp -->

<div data-role="footer">
  <h4>Jaluse tekst</h4>
</div> <!--teise lehe jalus -->
</div> <!--teise lehe lõpp -->
</body>
```

Koodinäide 2. jQuery Mobile ühe lehe rakendus¹⁴

jQuery Mobile'1 on MIT¹⁵ litsents. See on tasuta kasutamiseks ning avatud lähtekoodiga (ingl *open-source software*).

jQuery Mobile jaoks on loodud foorum¹⁶. See pole eriti aktiivne, uusi postitusi tuleb hetkeseisuga päevas ainult keskmiselt üks, vahest isegi kahe päeva peale üks ainus. Kokku on seal postitusi üle 13000. Stackoverflow's¹⁷ annab märksõna „jQuery Mobile“ rohkem kui 47000 vastet ja „jQuery“ rohkem kui 784000. Võrreldes teiste raamistikega on seda väga palju.

2.2 Kendo UI

Üks Kendo tugevusi on kohalike rakenduste kasutajaliides mitmetel erinevatel platvormidel: iOS, Android, BlackBerry ja Windows Phone 8 (Grgur Grisogono, 2014). Muuta saab iga kasutajaliidese osa värvi, tekstide kirjatüüpi, lisada erinevaid mustreid elementidele ja ka üleminekuvärve (ingl *gradient*). Kogu kasutajaliidese disainiprotsess on tehtud väga lihtsaks ning mugavaks. Kendo UI Mobile'1 on sarnaselt jQuery Mobile kasutajaliidese disainimise rakendusega ka enda rakendus, nimega ThemeBuilder¹⁸ (vt Joonis 4). CSS'i stiile kasutajaliidese osale saab peale panna hiirega lohistades soovitud kasutajaliidese osale ning selle peal olles hiire rullikuga saab valida täpsemalt, mis osale mingi värv või kirjatüüp läheb. Näiteks roheline värv lohistades *settings* nupu peale, saab hiire rullikuga valida kas see aktiveerub tekstile, taustale või äärele (ingl *border*). Peale sobive kujunduse loomist saab *export* nupule vajutades kätte CSS koodi. Samuti saab importida olemasolevat koodi, ning seejärel disainida. Ohutuks disainimiseks on olemas ka võta tagasi (ingl *undo*) ja üks samm

¹⁴ Saadud lehelt <http://demos.jquerymobile.com/1.3.0/docs/widgets/pages>

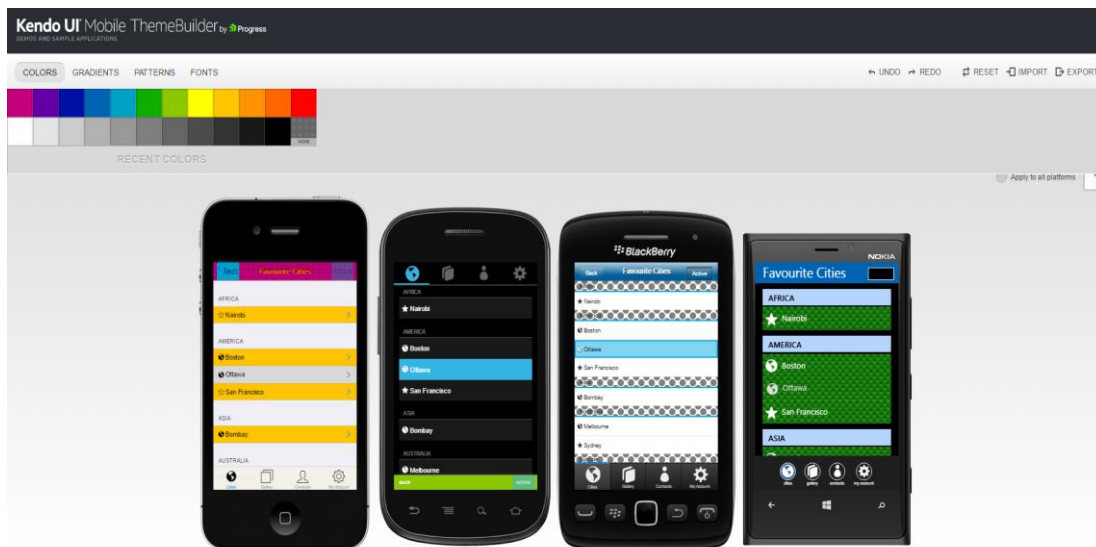
¹⁵ MIT litsents. <https://opensource.org/licenses/MIT>

¹⁶ jQuery Mobile foorum, saadaval lehel <https://forum.jquery.com/jquery-mobile>

¹⁷ Stack Overflow – veebileht, kus kasutajad saavad esitada programmeerimisega seotud küsimusi ja vastata teiste esitatud küsimustele. Saadaval lehel <http://stackoverflow.com>

¹⁸ Kendo UI Mobile ThemeBuilder – rakendus, millega on võimalik disainida iOS, Blackberry, Androidi ja Windows Phone mobiilseid kasutajaliideseid. Saadaval lehel <http://demos.telerik.com/kendo-ui/mobilethemebuilder>

edasi (ingl *redo*) nupud. Rakenduse sees on olemas ka dokumentatsioon ja lisamärkused, mis lihtsustavad malli loomist.



Joonis 4. Ekraanitõmmis Kendo UI Mobile ThemeBuilder rakendusest

Alates 2013. aasta novembrist, on kõik Kendo UI Mobile vidinad kohanduvad igal seadmel. Siiski töölauda rakenduste arendamiseks oleks mõistlikum kasutada Kendo UI Web raamistikku (Grgur Grisogono, 2014).

Kendo UI on jQuery'1 põhinev raamistik, millega on võimalik arendada rakendusi igale platvormile ning seadmele. Ainult nutiseadmetele mõeldud raamistik kannab nime Kendo UI Mobile.

Kendo UI arendajaks on ettevõtte Telerik¹⁹, kes arendab pidevalt oma loodud raamistikke ning uuendab neid kogu aeg, andes aasta jooksul üldiselt kolm suuremat väljalaset.

Kendo UI Mobile kasutab *Model-View-ViewModel* (MVVM) arendusmustrit (ingl *pattern*), mis muudab arendusprotsessi kiiremaks, ning andmete kättesaamine on lihtne nii eeskomponendile kui serveripoolele.

Kendo UI Mobile raamistikul on väga suures koguses erinevaid vidinaid saadaval. Vidinad on jagatud eraldi kategooriatesse ja iga vidina kohta on põhjalik kirjeldus koos koodinäitega ja piltidega. Kuna Kendo UI Mobile on jQuery põhjal ehitatud, siis on nende vidinate kasutamine sarnane jQuery Mobile raamistiku vidinatega. Näiteks *ColorPicker* vidin teeb

¹⁹ Telerik ettevõtte. <http://www.telerik.com/>

kolme rea koodiga (vt Koodinäide 3) tavalisest HTML elemendist täiesti teistsuguse ja palju suurema funktsionaalsusega vidina, mis võimaldab kasutajal värv valida vajutades soovitud värvile ning näha selle värvikoodi. (vt Joonis 5).

```
<input type="color" id="myColorField" /> //tekitab tavalise HTML värvi sisendi elemendi
<script>
  $(document).ready(function(){
    $("#myColorField").kendoColorPicker(); //muudab tavalise HTML elemendi Kendo UI Mobile vidinaks
  });
</script>
```

Koodinäide 3. Kendo UI Mobile ColorPicker vidin



Joonis 5. Kendo UI Mobile ColorPicker vidin²⁰

Kendo UI on tegelikult mõeldud töölaua rakenduste jaoks, Kendo UI Mobile on vaid üks osa suurest raamistikust. Seega saab Kendo UI'ga väga hästi luua ka töölaua rakendusi (Grgur Grisogono, 2014).

Et teha mobiilirakendust, siis Kendo UI raamistik seda ei toeta, ning tuleks kasutada väliseid vahendeid, nagu näiteks Cordovat. Samuti on Kendo UI Mobile raamistiku arendanud firma Telerik arendanud välja ka enda rakenduste pakendamist platvormi AppBuilder²¹, mida saab väga edukalt kasutada Kendo UI Mobile raamistikuga koos (Grgur Grisogono, 2014).

Põhiline dokumentatsioon on näidete kujul ametlikul kodulehel. Enamus seadistamise võimalused ja muu funktsionaalsus on lihtsate koodinäidetena saadaval. Lisaks sellele on ka

²⁰ Colorpicker vidina pilt võetud aadressilt <http://docs.telerik.com/kendo-ui/controls/editors/colorpicker/hsv-dropdown.png>

²¹ AppBuilder – Telerik ettevõtte poolt tehtud rakenduste pakendamise platvorm. Saadaval aadressil <http://www.telerik.com/platform/appbuilder>

mitmeid õpetusi, kaasaarvatud serveri pool (ingl *server-side*) ASP.NET²²i või PHP läbi (Grgur Grisogono, 2014).

Ametlikult kodulehelt on võimalus registreerida lühikesele kursusele (kaks korda neli tundi), mis õpetab kasutama Kendo UI raamistikku ning selle kursuse jooksul tehakse läbi erinevad stsenaariumid, mis võivad tekkida arendades Kendo UI raamistikuga. Hinnaks on sellel ligikaudu 230 €.

Kendo UI Mobile'1 on litsents ärielistel eesmärkidel kasutamiseks, mille hinnad algavad ligikaudu 190'st eurost (Grgur Grisogono, 2014).

Rolliga liikmed saavad abi küsida ametlikest foorumitest, mis on mõeldud esmajärgulistele kasutajatele. StackOverflows on üle 9000 küsimuse Kendo UI kohta, mis on suur hulk, kuid mitte kõik pole suunatud just Kendo UI Mobile'le. Ainuüksi märksõna Kendo UI Mobile peale tuleb üle 500 küsimuse. Telerik üldiselt on aktiivselt osalenud ülemaailmsel üritustel ning on loonud ka mitmeid konverentsisarju erinevatel teemadel (Grgur Grisogono, 2014).

2.3 Ionic

Ionic on loodud 2013 aasta novembris Drifty²³ nimelise ettevõtte poolt ning on ehitatud AngularJS²⁴ peale. AngularJS'i roll on luua rakenduse struktuur ja Ionic loob kasutajaliidese. Üldiselt kasutataksegi neid kahte raamistikke koos. Hetkel kõige viimane versioon Ionic 1.3 tuli välja 21 aprill, 2016 (Ionic, kuupäev puudub). Ionic on npm²⁵ moodul, ning selle kasutamiseks on vajalik veel Node.js²⁶.

Tänu sellele, et Ionic töötab AngularJS peal, mitte jQuery, on võimalik saavutada rakendustes parem jõudlus (Ionic, kuupäev puudub).

Ionic sisaldab palju erinevaid kasutajaliidese elemente, mida on lihtne kasutada ja oma rakendusse lisada. Tuleb vaid lisada HTML elemendile eelnevalt CSS'iga koostatud *class*, mis muudab elemendi kujundust. Üks asi, mis on Ionic'ul ja paljudel teistel raamistikel pole, on *flexbox grid* süsteem. *Flexbox* on CSS3'e atribuut, mida hetkel toetavad vaid kõige

²² ASP.NET – Microsofti poolt arendatud programmeerimiskeel

²³ Drifty ettevõtte ametlik blogi. Saadaval lehel <http://blog.drifty.com/>

²⁴ AngularJS on Google poolt loodud populaarne JavaScript'i raamistik. Saadaval lehel <http://angularjs.org>

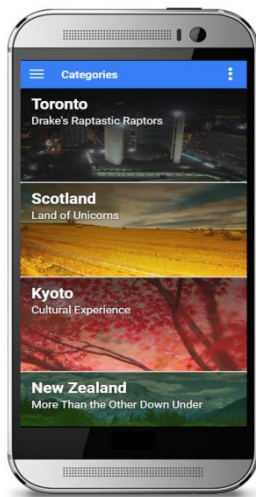
²⁵ NPM on JavaScripti pakettide haldur. Saadaval lehel <https://www.npmjs.com>

²⁶ Node.js on serveripoolne JavaScripti platvorm. Saadaval lehel <https://nodejs.org/en/download/>

uuemad internetibrauserid. *Grid*'i lisamine käib sama moodi nagu teiste Ionicu kasutajaliidese elementide kujundamine. Lisaks on veel üle 500 ikooni, või *Ionicons*'i²⁷, nagu nad ise neid nimetavad (Ionic, kuupäev puudub).

Ionicu erinevus tuleb välja lihtsuses ja Google stiilis kasutajaliidese elementides. Tänu sellele, et kõik elemendid on väga lihtsad ja ei ole lisatud üleliigseid ääristusi ega varje, on Ionicu raamistik kiire. Ionic ei sisalda kohalike rakenduste kasutajaliidest, kuid sisaldab kiiret ja puhast kasutajaliidest.

Ionic'ule on loodud Zach Fitzgerald'i poolt veel eraldi kasutajaliidese laiendus-raamistik (ingl *extension library*) *Ionicmaterial*²⁸, mis pakub valmisolevaid kujundusmalle tasuta kasutamiseks (vt Joonis 6).



Joonis 6. Ekraanitõmmis ionicmaterial'i ühest pakutavast valmiskujundusest

Ionic'u komponendid on lihtsad ja ei sisalda midagi üleliigset. Üldiselt on vidinad HTML elemendid. Võrreldes teiste raamistikega on Ionic'u vidinad pigem lihtsad lehe blokid, mida omavahel kombineerides saab luua korraliku kasutajaliidese. Hetke seisuga pakub Ionic mitmeid vormi elemente (ingl *form elements*), päise tööribasi (ingl *header toolbar*), nuppe, liste, grid'e ja veel palju muud (Grgur Grisogono, 2014). Erinevaid vidinaid on ametlikus dokumentatsioonis 15, kuid igast vidinast on mitmeid erinevaid versioone.

²⁷ Ionicons ikoonide šrift Ionic raamitsikule. Saadaval lehel <http://ionicons.com/>

²⁸ Ionicu'le mõeldud kujundusmallide raamistik. Saadaval lehel <http://ionicmaterial.com>

Ionic on eelkõige mõeldud hübriid mobiilirakenduste loomiseks, ning töölaua rakendusi ei soovitata sellega ehitada. Sisu küll kohandub, kuid tulemus ei jää nii korralik, nagu jääks see mõne teise töölauale mõeldud raamistikuga tehes (Grgur Grisogono, 2014).

Ionic'ul on hetkel kaks versiooni: Ionic 1 ning Ionic 2, millest Ionic 1 on mõeldud pigem hübriidrakenduste loomiseks ja Ionic 2 võimaldab luua nii hübriidrakendusi kui ka töölaua rakendusi Windows 10 operatsioonisüsteemile (Ionic, kuupäev puudub).

Ionic raamistik toetab PhoneGap'i, Cordova ning Trigger.io²⁹ pakendamist. Ionic'ul on ka pilve platvorm (ingl *cloud platform*) Ionic Cloud³⁰, millega on võimalik pakendada rakendust. Ionic Cloud sisaldab veel mitmeid võimalusi, nagu näiteks anda oma arendusmeeskonnale ligipääs arendatavale rakendusele väga lihtsalt.

Ionic'u kodulehel on põhjalik dokumentatsioon³¹. See sisaldab lühikest sissejuhatavat ülevaadet raamistikust ja selle erinevatest versioonidest. Lisaks sellele, on neil kuue osaline raamat, mida saab tasuta ja ilma allalaadimata lugeda nende dokumentatsiooni lehelt. See sisaldab kogu informatsiooni, alates raamistiku tutvustusest lõpetades rakenduse väljastamisega.

Ionic'ul on MIT litsents. Samuti on see avatud lähtekoodiga ja seda võib vabalt projektides kasutada (Ionic, kuupäev puudub).

Ionic'ul on oma ametlik foorum³², mis on keskmiselt aktiivne. Päevas tuleb hetke seisuga sinna ligikaudu 50 uut postitust. StackOverflow's on hetkel üle 29000 küsimuse.

2.4 Meteor

Meteor on täispinu (ingl *full-stack*) JavaScriptil põhinev arendusplatvorm ja raamistik, millega saab arendada kaasaegseid mobiilseid rakendusi ja ka töölaua rakendusi. See ühildub hästi selliste veebitehnoloogiatega nagu AngularJS, mongoDB³³, npm ning Cordova. Hetkel

²⁹ Trigger.io on hübriidrakenduse pakendamise platvorm, mis põhineb JavaScriptil. Sarnaneb PhoneGap'i ja Cordovaga. Saadaval lehel <https://trigger.io>

³⁰ Ionic Cloud platvorm. Saadaval lehel <https://ionic.io/cloud>

³¹ Dokumentatsioon saadaval lehel <http://ionicframework.com/docs/guide/>

³² Ionic raamistiku ametlik foorum. Saadaval lehel <https://forum.ionicframework.com>

³³ MongoDB andmebaas, saadaval lehel <https://www.mongodb.com>

on kõige uuem versioon Meteor 1.4. Seda raamistikku on iseloomustatud oma ametlikul kodulehel järgmise lausega: „Tee rohkem vähema koodiga“ (ingl *Ship more with less code*). Meteoriga saab ehitada nii veebirakendusi kui ka hübriidrakendusi. Samuti saab kasutada muid raamistikke samaaegselt väga lihtsalt koos Meteoriga (Meteor, kuupäev puudub).

Meteoriga tuleb kaasa ainult üks kasutajaliidese mall, milleks on Blaze³⁴. Blaze on Meteor raamistiku front-end kujundamise süsteem, millega on võimalik luua kergesti kasutatavaid kasutajaliideseid. Meteor dokumentatsioon viitab edasi Blaze'i dokumentatsioonile kust leiab erinevaid kujunduspakette ja kasutajaliidese osi. Blaze'i kasutajaliides on kirjutatud Spacebars mallide loomise keeles (ingl *templating language*). Spacebars on esmapilgul sarnane HTML'iga, kuid klasside asemel kasutatakse loogelisi sulgusid, mille vahele pannakse varasemalt JavaScript'i failis deklareeritud klasse, mis annavad sellele elemendile väärtuse. (Blaze, kuupäev puudub).

Meteoril on vidinate jaoks loodud eraldi kataloog Atmosphere³⁵. Sealt leiab üle 11000 vidina, mida kasutada Meteor'i rakenduses. On võimalik ka sorteerida välja kõige populaarsemad või kõige uuemad vidinad.

Kui üldiselt veebiarendajatele on tüütu see, et nad peavad rakenduse erinevaid osi kirjutama erinevates programmeerimiskeeltes, siis Meteor'i puhul saab teha kõike JavaScript'iga. Näiteks Meteor'iga on väga lihtne luua ja suhelda andmebaasiga. Andmebaasi saab panna muutuja külge, ning hiljem sinna asju sisestada (vt Koodinäide 4). Kogumik (ingl *Collection*) on Meteor'is samaväärne SQL tabeliga.

```
BlogPosts = new Meteor.collection('posts');
BlogPosts.insert({
  Title: 'Hello World',
  Content: 'This is the content.',
  Published: true
});
```

Koodinäide 4. Andmebaasi loomine Meteor'i's ning sinna andmete sisestamine

Otsest töölaua rakenduste tuge Meteor ei paku. Kuid hetkel on arendamisel npm'is Meteor Desktop pakett. Selleks on vaja vähemalt Meteor 1.3.3 versiooni ja teadmisi Electron³⁶ platvormist. Electron on idee poolest sarnane PhoneGap'iga ja Cordova'ga, ainult, et mõeldud

³⁴ Blaze saadaval lehel <https://guide.meteor.com/blaze.html>

³⁵ Atmosphere on Meteor raamistiku vidinate kataloog. Saadaval lehel: <https://atmospherejs.com>

³⁶ Electron – töölaua rakenduste loomise platvorm. Saadaval lehel <http://electron.atom.io>

töölaua rakenduste loomiseks. Hetkel pole see veel eriti populaarsust kogunud (Meteor, kuupäev puudub).

Meteori rakendust saab pakendada kasutades Cordova't.

Meteoril on suur kommuun. Nende kodulehel on kirjas järgmine statistika: 11591 paketti (ingl *package*), 184 professionaalset teenusepakkujat, üle 23000 küsimuse Stackoverflows ja 508000 unikaalset Meteor'i installeerimist. Hetke seisuga on tulemas 28 Meteor'i teemalist üritust üle maailma (Meteor, kuupäev puudub). Meteoril on ka oma üsna populaarne foorum³⁷. Awesome-meteor³⁸ repositoorium, mis on loodud Meteor'i kasutajate ja kommuuni poolt välja otsitud abistavatest linkidest ja erinevatest teemadest.

Meteor pakub õpetusi, mis juhendab märkmiku rakenduse ja Whatsapp³⁹ rakenduse nullist järgi tegemist.

Meteor on väga kasutajasõbralik, see tähendab seda, et kõik vajalik nii tutvustav kui ka juba edasijõudnutele vajalik informatsioon on väga hästi leitav. Ainuüksi dokumentatsiooni jaoks on loodud eraldi dokumentatsiooni leht⁴⁰ ning õpetuste lehel⁴¹ on erinevad juhendid ja koodinäited. Sealt leiab üldist informatsiooni Meteor'i kohta, näidisrakendusi, soovitusi kuidas kirjutada Meteor'iga lihtsasti loetavat ja arusaadavat koodi, erinevate pakettide kasutamist, otsingumootori jaoks optimeerimise ehk SEO (ingl *Search Engine Optimisation*) kasutamisest, serveri-poolse õpetusi ja soovitusi – sealhulgas mongoDB kasutamise näiteid ja palju muud. Sealt juba leiab kogu vajaliku info, mida on vaja, et teha ilma varasema Meteor'i teadmiseta valmis mobiilne veebirakendus või hübriidrakendus (Meteor, kuupäev puudub).

Meteor on saadaval avatud lähtekoodiga.

2.5 Raamistike võrdlus

Kuna kõik raamistikud erinevaid üksteisest mingil moel, siis on raske hinnata, et milline nendest on kõige parem. Kõik oleneb sellest, mis sorti rakendust kasutaja soovib arendada ning milline peab rakenduse funktsionaalsus olema. Kui on vaja luua väga ressursinõudlik ja

³⁷ Meteor'i foorum, saadaval lehel <https://forums.meteor.com>

³⁸ Awesome-meteor. Saadaval lehel <https://github.com/Urigo/awesome-meteor>

³⁹ Rakendus WhatsApp. Saadaval lehel <https://www.whatsapp.com>

⁴⁰ Meteor'i dokumentatsiooni leht. Saadaval lehel <https://docs.meteor.com>

⁴¹ Meteor'i õpetuste leht. Saadaval lehel <https://guide.meteor.com>

turvaline rakendus, siis on mõistlikum üldse seadmele omases programmeerimiskeeles luua kindla operatsioonisüsteemi peale eraldi rakendus. Kui aga rakenduse kiirus pole nii oluline, siis on palju lihtsam ja mugavam kasutada mõnda raamistikku, millega luua rakendus, kirjutades see samades programmeerimiskeeltes igale operatsioonisüsteemile ning soovi korral see pakendada ja rakenduste poodi saadavaks teha.

Võrreldud raamistikest koostati ülevaatlik tabel (vt LISA 1), mille abil saab välja tuua, et Stackoverflow vastete põhjal on jQuery Mobile kõige populaarsem. jQuery Mobile põhineb jQuery'l, mis on hetkel kõige populaarsem JavaScript'i raamistik veebiarenduses (üle 780000 küsimuse Stackoverflows). Meteor ja Ionic on Stackoverflow põhjal enam-vähem sama populaarsed, ning Kendo UI Mobile kõige mittepolaarsem. See võib olla tingitud sellest, et Kendo UI Mobile on võrreldavatest ainuke raamistik millel on litsents ärilistel eesmärkides kasutamiseks, ning maksab ligikaudu 190€. jQuery Mobile'l ja Ionic'ul on MIT litsents ning Meteor on avatud lähtekoodiga.

Dokumentatsiooni koha pealt on Meteor kõige kasutajasõbralikum. Meteor'i kodulehelt leiab väga põhjaliku dokumentatsiooni ja ka õpetuste lehe, kus on kõik rakenduse arendamise aspektid välja toodud ja neid kirjeldatud. Samuti on ka Ionic'ul väga põhjalik dokumentatsioon ja tasuta raamat, kus räägitakse kõigest alates raamistiku tutvustusest kuni rakenduse väljastamiseni. Kendo UI Mobile'l ja jQuery Mobile'l on ka hea dokumentatsioon, kuid need põhinevad üldiselt koodinäidetel.

Rakenduse pakendamise võimalust pakuvad Kendo UI Mobile ja Ionic. Teiste raamistike puhul on võimalik pakendada kasutades Phonegap'i või Cordova't.

Otsene töölaua rakenduste arendamise tugi on ainult Kendo UI Mobile'l ja Ionic'ul, kuid Meteor raamistikule on tehtud Electron nimeline tarkvara, millega on ka võimalik töölaua rakendusi luua. jQuery Mobile raamistik on mõeldud ainult mobiilsete seadmete peale, kuid rakendust veebilehitsejas avades see küll töötab, kuid on siiski optimiseeritud mobiilsetele seadetele.

Vidinate koha pealt on Meteor'il kõige suurem valik, kuna on loodud eraldi vidinate kataloog nimega Atmosphere, kust võib leida üle 11000 erineva vidina. jQuery Mobile pakub 28't erinevat vidinat, Kendo UI Mobile 14 ja Ionic 15. Kuid nende raamistike puhul jagunevad vidinad veel omakorda mitmeks erinevaks vidinaks, mis erinevad väikse funktsionaalsuse või disaini poolest üksteisest.

Kasutajaliidese mallide puhul on kõige paindlikum valik jQuery Mobile ja Kendo UI Mobile raamistikel, kuna mõlematel on loodud rakendus, kus on mugav kasutajaliidese osi disainida ning hiljem saada CSS'i kood sealt kätte. Ionic'u kasutajaliidese plussiks on Google stiilis kasutajaliides ning väga lihtsad ja vähem ressursinõudlikumad kasutajaliidese osad. Meteor sisaldab kasutajaliidese mallide asemel raamistikku Blaze, kuid selle kasutamiseks on vaja lisa teadmisi mallide loomise keelest Spacebars.

Kokkuvõte

Seminaritöö on hetkel aktuaalne, kuna kõik võrreldavad rakendused on hetkel populaarsed ja üha enam arendatakse rakendusi pigem kasutades mõnda raamistikku ja igale seadmele korraga, kui kohalikus programmeerimiskeeles eraldi operatsioonisüsteemidele. Tänapäeval on väga paljudest veebirakendustest tehtud ka mobiilsed rakendused. Seminaritöös uuritavate raamistike abil on mugav teha mobiilirakendusi igale operatsioonisüsteemile korraga.

Seminaritöö eesmärgi saavutamiseks tutvustati mobiilirakenduste loomise erinevaid võimalusi ning anti ülevaade mobiiliraamistikust jQuery Mobile, Kendo UI Mobile, Ionic ja Meteor. Seejärel võrreldi neid kindlate kriteeriumite järgi ning koostati ülevaatlik tabel, mis on kättesaadav töö lisas.

Raamistike võrdluse käigus selgus, et pole kindlat parimat raamistikku, vaid neil kõigil on oma eripärad ning enne rakenduse arendamist tuleks mõelda, et milline raamistik sobib kõige paremini rakenduse funktsionaalsusega. Küll aga tuli välja, et kõige populaarsem raamistik on jQuery Mobile ja ainuke raamistik ärielistel eesmärkidel kasutamise litsentsiga on Kendo UI Mobile. Kõige parema dokumentatsiooniga ning suurima vidinate hulgaga on Meteor raamistik. Kõiki võrreldud raamistikke on võimalik pakendada hübriidrakenduseks ning teha neid rakenduste poodides saadavaks.

Seminaritöö raames õppis autor võrreldavate raamistike kohta lisaks ning autoril tekkis huvi raamistikke edasi uurida ning osadega neist ka rakendusi arendada.

Loodud seminaritöö temaga on võimalik edaspidi rohkem süvitsi minna ning teha ka näidisrakendusi ja võrrelda rakenduste arendamist nende raamistike abil. Samuti on võimalik luua ka valitud rakendustest eestikeelseid õppematerjale, mis õpetavad mõnda konkreetset raamistikku arendamisel kasutama.

Kasutatud kirjandus

Danny Markov (2015). *Comparing The Top Frameworks For Building Hybrid Mobile Apps*. Loetud aadressil <http://tutorialzine.com/2015/10/comparing-the-top-frameworks-for-building-hybrid-mobile-apps/>

John Bristowe (2015). *What is a Hybrid Mobile App?* Loetud aadressil <http://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/>

Jake Rocheleau (kuupäev teadmata). *Beginner's Guide to jQuery Mobile*. Loetud aadressil <http://www.hongkiat.com/blog/build-websites-apps-jquery-mobile/>

Thoriq Firdaus (kuupäev teadmata). *10 Frameworks to Build Mobile Application with HTML, CSS & JavaScript*. Loetud aadressil <http://www.hongkiat.com/blog/mobile-frameworks/>

Raluca Budiu (2013). *Mobile: Native Apps, Web Apps, and Hybrid Apps*. Loetud aadressil <https://www.nngroup.com/articles/mobile-native-apps>

Dieter Bohn (2012). *Mark Zuckerberg promises a native Android app, says betting on HTML5 for mobile was a 'mistake'*. Loetud aadressil <http://www.theverge.com/2012/9/11/3317230/mark-zuckerberg-betting-on-html5-for-mobile-was-a-mistake-hints-at>

Jorge Ramon (2015). *Packaging a jQuery Mobile Application with Cordova/Phonegap*. Loetud aadressil <https://dzone.com/articles/packaging-jquery-mobile>

Grgur Grisogono (2014). *5 Best Mobile Web App Frameworks: Kendo UI Mobile*. Loetud aadressil <http://moduscreate.com/5-best-mobile-web-app-frameworks-kendo-ui-part2/>

Grgur Grisogono (2014). *5 Best Mobile Web App Frameworks: Ionic (AngularJS)*. Loetud aadressil <http://moduscreate.com/5-best-mobile-web-app-frameworks-ionic-angularjs/>

Ionic – mobile app framework. (kuupäev puudub). Wikipedia. Loetud 04. november 2016 aadressil [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)#cite_ref-8](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework)#cite_ref-8)

Dragan Gaić (2015). *What do mobile developers think of Ionic framework*. Loetud aadressil <https://www.quora.com/What-do-mobile-developers-think-of-Ionic-framework>

jQuery Mobile. (kuupäev puudub). *Category: Widgets*. Loetud aadressil <https://api.jquerymobile.com/category/widgets/>

jQuery Mobile. (kuupäev puudub). *Textinput Widget*. Loetud aadressil <https://api.jquerymobile.com/textinput/>

Globetrotter (2014). *Hybrid Applications And Android Native Browser*. Loetud aadressil <https://myshadesofgray.wordpress.com/2014/04/15/hybrid-applications-and-android-native-browser/>

Ionic (kuupäev puudub). *Ionic Framework*. Loetud aadressil <http://ionicframework.com/>

Ionic (kuupäev puudub). *Browser support*. Loetud aadressil <http://ionicframework.com/docs/overview/#browser-support>

Ionic (kuupäev puudub). *Ionic documentation overview*. Loetud aadressil <http://ionicframework.com/docs/overview>

Ionic (kuupäev puudub). *Icons*. Loetud aadressil <http://ionicframework.com/docs/components/#icons>

Meteor (kuupäev puudub). *Meteor framework*. Loetud aadressil <https://www.meteor.com>

Blaze (kuupäev puudub). *Spacebars templates*. Loetud aadressil <http://blazejs.org/guide/spacebars.html>

Meteor (kuupäev puudub). *Meteor guide*. Loetud aadressil <https://guide.meteor.com/>

LISA 1

<i>Raamistik</i>	<i>Kasutajaliidese mallid</i>	<i>Vidinate Hulk</i>	<i>Töölaua toetus</i>	<i>Pakendamine</i>	<i>Dokumentatsioon</i>	<i>Litsents</i>	<i>Kommuun</i>
jQuery Mobile	ThemeRoller	28	Olemas, kuid pigem jQueryt kasutada	PhoneGap/Cordova	Põhjalik dokumentatsioon, enamasti koodinäidete kujul	MIT litsents	Ametlik ebaaktiivne foorum. Stackoverflows 47000+ küsimust
Kendo UI Mobile	Themebuilder	14	Optimeeritud mobiilsete seadetele. Töölaua rakenduste jaoks Kendo UI Web	AppBuilder. Toetab ka Cordova't ning PhoneGap'i	Palju näidiskoodi ja koodinäiteid	Litsents ärilistel eesmärkidel kasutamiseks. Maksab ligikaudu 190€	Toetajatele ametlik foorum. Stackoverflows 'Kendo UI' märksõnaga 9000+ vastet, 'Kendo UI Mobile' 500+ vastet
Ionic	Lihtsad ja puhtad kasutajaliidese osad. Google stiil	15	Ionic 2'ga on võimalik Windows 10 peale töölaua rakendusi arendada	Phonegap, Cordova, Trigger.io, Ionic Cloud	Põhjalik dokumentatsioon ning tasuta raamat	MIT litsents	29000+ vastet Stackoverflows. Ametlik foorum, umbes 50 postitust päevas
Meteor	Blaze	Atmosphere (üle 11000)	Otsene tugi puudub. Electron	Cordova	Väga põhjalik	Avatud lähtekoodiga	23700+ vastet Stackoverflows. 508000 unikaalset installeerimist. 28 üritust tulemas. 184 professionaalset teenusepakkujat