

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut
Informaatika eriala

Mati Kärner

Avatud lähtekoodiga veebipõhiste kasutajaliideste raamistike võrdlus

Bakalaureusetöö (6 EAP)

Juhendaja: dotsent Helle Hein

Autor: " " mai 2013

Juhendaja: " " mai 2013

Lubada kaitsmisele

Professor: " " mai 2013

Tartu 2013

Sisukord

Sissejuhatus	3
1. Kasutajaliideste raamistikud.....	4
1.1 Eesrakendus	4
1.2 Kasutajaliideste raamistikud veebirakendustes	5
1.3 Võrreldavad raamistikud	6
1.4 Võrdluse kriteeriumid.....	7
1.5 Võrdluse metoodika.....	8
1.6 Testrakendus	11
2. Raamistike võrdlus	13
2.1 Bootstrap.....	13
2.2 Dojo	16
2.3 YUI.....	20
2.4 SproutCore.....	24
2.5 Qooxdoo	27
2.6 Enyo.....	32
3. Kokkuvõtte võrdlustulemustest	36
Kokkuvõtte	39
Summary.....	40
Kasutatud kirjandus	41
Lisad	44
Lisa 1	44
Lisa 2	45
Lisa 3	46

Sissejuhatus

Veebi algusajaks võib pidada aastat 1989, mil Tim Berners-Lee eestvedamisel hakati Euroopa Tuumafüüsika Laboris (CERN) ühendama erinevaid teadusartikleid hüperlinkide abil. Algselt dokumentide jagamiseks mõeldud süsteemist on tänaseks saanud ulatuslikult kasutatav üldotstarbeline keskkond. Internetipangandus, sotsiaalmeedia jpt rakendusvaldkonnad on muutnud veebi meie argipäeva lahutamatuks osaks. Tehnoloogiate, nagu AJAX, HTML5 ja WebGL kasutuselevõtt on veebist teinud atraktiivse tarkvaraplatvormi, mistõttu veebirakendustes võib näha arvestatavat konkurenti töölauarakendustele [1].

Kasutajaliidese disain ja kompositsioon moodustavad olulise osa veebirakenduse planeerimisest ja teostusest [2], kuivõrd sellest sõltub rakenduse aluseks oleva funktsionaalsuse efektiivne kasutamine. Kasutajaliideste raamistikud võimaldavad taaskasutatavate komponentide abil ehitada kiiresti nii lihtsamaid kui ka olemasolevaid komponente kombineerides komplekssemaid kasutajaliideseid. Erinevad raamistikud, lisaks erinevatele tehnoloogiatele ja komponentide valikule, seavad konventsioonide näol raamid arendatavale või planeeritavale rakendusele - seda alates projekti struktuurist kuni ehitusloogika defineerimiseni [3]. Seega ei pruugi olemasolevate toodete hulgast valiku tegemine olla triviaalne tegevus.

Eeltoodut arvesse võttes ongi käesoleva bakalaureusetöö eesmärgiks analüüsida kuut levinumat avatud lähtekoodiga raamistikku. Iga raamistiku puhul antakse lühike ülevaade põhilistest tunnusjoontest ja rakendamisevõimalustest ning võrreldakse raamistikku teistega vastavalt väljatöötatud võrdlusmetoodikale. Võrdlemisel tuginetakse raamistike autorite publitseeritud infol ja dokumentatsioonil. Saamaks ülevaadet raamistiku pakutavatest võimalustest ning puudustest, realiseeritakse iga raamistiku funktsionaalsust kasutades testrakendus, mille jõudluse hindamine on ühtlasi üheks võrdlusmetoodika osaks. Võrdluse tulemusena moodustub raamistikest pingerida.

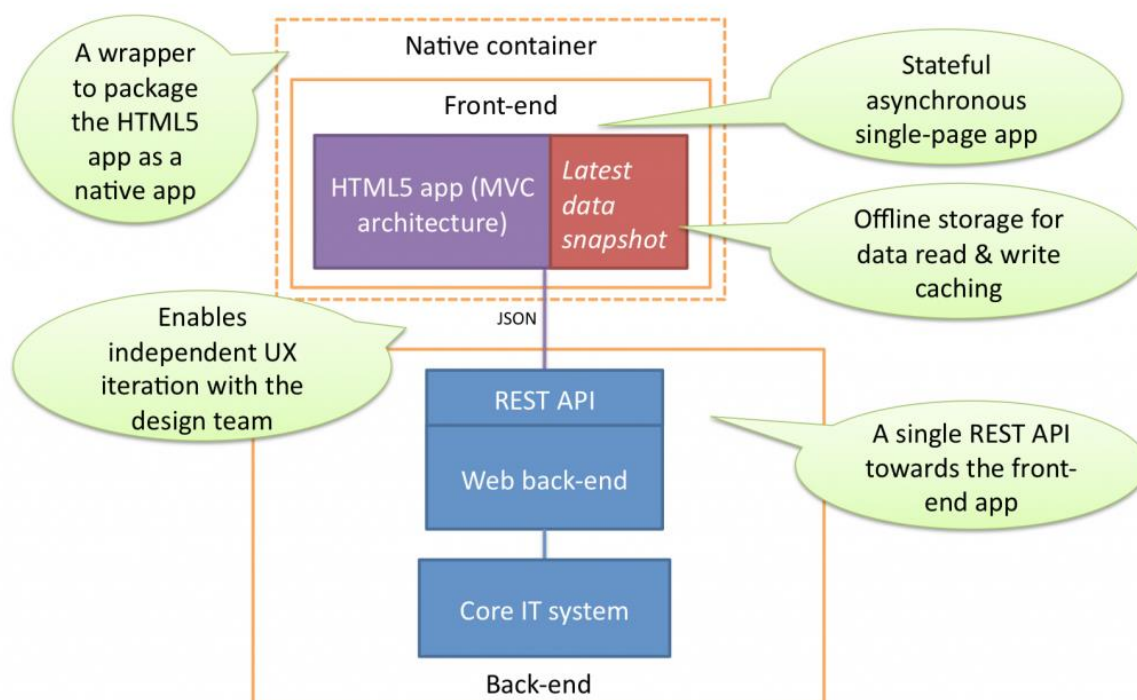
Töö koosneb kolmest peatükist. Esimeses peatükis antakse nimekiri võrreldavatest raamistikest ning tutvustatakse valiku põhimõtteid. Samuti kirjeldatakse võrdlusmetoodikat ning testrakenduse struktuuri. Teises peatükis analüüsitakse raamistike omadusi vastavalt võrdlusmetoodikale. Kolmandas osas tehakse analüüsi tulemustest kokkuvõtte.

1. Kasutajaliideste raamistikud

Viimase aastakümne jooksul on veebist saanud stabiilne tarkvaraplatvorm [1,4], mida iseloomustavad interaktiivsed rakendused ja teenused. Tarkvaraarenduses eelistatakse järjest enam veebirakendusi töölaarakendustele. Viimast kinnitab ka järjest suurenev pilvepõhiste (*Software as a Service*) toodete arv, seda eeskätt asukoha ja ajaliste piirangute puudumise kui ka kasutuskorralt maksmisel (*Pay per use*) põhineva ärimudeli levimise tõttu [4,5].

1.1 Eesrakendus

HTML5 ja AJAX kasutusele võtuga on veebirakenduste arendamisel järjest populaarsust kogunud üheleheline (*single page*) arhitektuur, kus äri loogika on jäetud tagarakenduste hooleks ning suurema skaleeruvuse saavutamiseks tuginetakse kliendipoolsel andmete töötusel - nn iseseisval eesrakendusel. Selline areng omakorda esitab esitluskihile kui ka kasutajaliidese funktsionaalsusele kõrgendatud nõudmised. Samalaadsed nõudmised kohalduvad ka mitmeleheliste rakenduste puhul, sest sarnased tehnoloogiad on ka seal kasutusel.



Joonis 1. Ühelehelise rakenduse struktuur [6].

Erinevate uuringute järgi on keskmiselt kolm neljandikku arendajatest HTML5 kasutajad [7], kuid arvestades, et eelnimetatud tehnoloogia on alles standardiseerimise faasis, siis võib eeldada, et see arv kasvab veelgi.

Joonisel 1 on toodud tüüpilise ühelehelise rakenduse skemaatiline ülesehitus. Rakendus on jagatud ees- ja tagarakenduseks. Viimases realiseeritud programmeerimisliidese abil saab eesrakendus vahetada andmeid ning sooritada muudatusi tagarakenduses asuvas andmekihis. Eesrakendus on olekut säilitav (*stateful*), sageli loodud mõne MVC (*Model-view-controller*) mustrit toetava JavaScriptil põhineva raamistiku abil ning kasutab andmete puhverdamiseks HTML5 Local Storage tehnoloogiat. Andmevahetus ees- ja tagarakenduse vahel on asünkroonne, kusjuures eesrakendus blokeerub üksnes siis, kui kliendi poolt sooritatud päring nõuab kohest vastust [6].

Sõna eesrakendus (*front-end*) on nii erialases- kui ka teaduskirjanduses küllaltki ähmane mõiste, hõlmates endas nii veebirakenduste esteetilise kui ka funktsionaalse poole. Sisuliselt tähendab eesrakendus veebirakendustes kasutajale nähtavat kasutajaliidest, mis võimaldab külastajal suhelda saidi dünaamiliste osadega e. tagarakendustega (*back-end*), nt andmebaasid. Kasutatavateks tehnoloogiateks on HTML, JavaScript, AJAX, CSS, Flash jpt. Mitmesuguseid serveripoolseid raamistikke ja keeli (PHP, ASP, ...) liigitatakse vastavalt nende rollile samuti eesrakendustes levinud tehnoloogiate alla, kuivõrd neid kasutatakse väga sageli rakenduse kuvamiseks vajaliku märgistuskeele genereerimiseks.

1.2 Kasutajaliideste raamistikud veebirakendustes

Tarkvaraarenduses mõistame raamistike all eeldefineeritud koodi või teekide kogumit, millele planeeritavat rakendust rajada. Võrreldes teekidega, mis enamasti pakuvad limiteeritud funktsionaalsust, defineerivad raamistikud rakenduse struktuuri ning võimaldavad koodi taaskasutamist. Raamistike kasutamise peamine eesmärk on rakenduste standardsete osade realiseerimise lihtsustamine jättes viimase raamistike hooleks ning lubades seeläbi arendajatel rakenduse põhilistele nõuetele keskenduda.

Veebirakenduste eesosade realiseerimiseks on kasutusel mitmeid tehnikaid nagu CSS stiililehtedel põhinevad võrestikud, eeldefineeritud HTML mallide kasutamine, teegid, raamistikud jpm. Liigitamine kasutajaliideste raamistikeks ja eesrakenduste raamistikeks on üpriski tinglik, aga arvestades, et praktikas kasutatakse neid sõnu sünonüümidena, siis käesolevas bakalaureusetöös käsitletakse neid ühe ja sama nähtusena.

Kasutajaliideste raamistikud on üheks kiireks kasutajaliideste ehitamise vahendiks. Arendamiskiirus on sageli oluline alustavatele ettevõtetele ja idufirmadele, kus rahalised vahendid on piiratud ning toote turule tulemise ajastusest võib sõltuda ettevõtte edukus. Lõputöö on koostatud just seda silmas pidades, kusjuures keskendub autori kogemusest lähtuvalt ning võrreldavuse huvides kasutajapoolsel skriptimiskeelel JavaScript põhinevatel raamistikel.

Avatud lähtekoodiga kasutajaliideste raamistike kasutamises veebirakenduste arendamisel võib näha järgmisi eeliseid:

- Standarditele nagu nt HTML5 ja CSS3 orienteeritus tagab nii rakenduse parema kättesaadavuse kui ka toe erinevatel platvormidel. Süntaktiliselt korrektne lähtekood muudab rakenduse töökindlaks ning ka otsingumootoritele sõbralikumaks.
- Rakenduste arendamine paljudele veebilehitsejatele on lihtsam, kuna viimaste eripärad on raamistikus hajutatud koherentse programmeerimisliidese abil [8].
- Raamistike kasutamine võimaldab hoida kasutajaliideste arendamise muust koodist lahus, lihtsustades nii kogu projekti koodibaasi haldamist.
- Erinevate seadmete tugi võimaldab hõlpsamini viia rakenduse kohanduva disaini (*responsive web design*) põhimõtetega vastavusse.
- Avatud lähtekood aitab kaasa väheste vigadega rakenduse valmimisele, sest raamistiku funktsionaalsust testitakse ja täiendatakse suure hulga teiste kasutajate poolt. Raamistiku arendamisse kaasatud kogukond on väärtuslikuks toeks arendamise käigus tekkivatele probleemidele.
- Eeldefineeritud stiililehed ja sündmuste töötlemise, sisendi valideerimise jt lisavõimalustega varustatud kasutajaliidese komponendid kiirendavad arendusprotsessi ning tagavad rakenduseülese välimuse.

1.3 Võrreldavad raamistikud

Bakalaureusetöös on võrdlemiseks välja valitud kuus enamlevinud raamistikku: Bootstrap [9] (versioon 2.3.1), Dojo [10] (versioon 1.8.3), YUI [11] (versioon 3.9.1), SproutCore [12] (versioon 1.9.2), Qooxdoo [13] (versioon 2.1.1) ja Enyo [14] (versioon 2.2). Kõik eelloetletud raamistikud on avatud lähtekoodiga.

1.4 Võrdluse kriteeriumid

Võrdluses vaadeldakse iga raamistiku puhul järgmisi kriteeriume:

1. Õppimiskiirus. Õppimiskiirus on rakenduse arendamise seisukohast üks tähtsamaid mõõdikuid, kuivõrd see mõjutab rakenduse valmimiseks kuluvat aega.
2. Sobivus erinevate veebilehitsejatega ja standarditele orienteeritus.
3. Lokaliseerimine. Väga suurt osa tänapäevasest tarkvarast püütakse levitada ülemaailmselt. Lokaliseerimine, s.o kasutajaliidese kohandamine vastavalt kasutaja keelelisele ja geograafilisele piirkonnale, on oluline suurendamiseks rakenduse konkurentsivõimelisust globaalsel turul. Kasutajaliideste raamistike seisukohast on oluline komponentide (tekstiväljad, dialoogaknad, ...) kohandamine vastavalt kasutuskohale.
4. Sisendi valideerimine. Sisendi valideerimine eesrakenduses aitab hoida kokku serveri ressursse, sest väga paljud triviaalsed kontrollid, nt nõutud väljade täidetud jms, saab teha kasutades kasutajapoolset andmetöötlust. Samuti paraneb veebirakenduste kasutusmugavus, sest kasutajat teavitatakse tekkinud vigadest kohe.
5. Integreeritud arenduskeskkonna (IDE) tugi. Integreeritud arenduskeskkond, pakkudes selliseid funktsioone nagu ennetav koodisisestus, projekti struktuuri haldamine, süntaksivigade esiletõstmine, koodi silumine jt, on hea vahend arendaja produktiivsuse tõstmiseks.
6. Kasutajatoe elemendid. Tootjate poolt publitseeritud dokumentatsioon, koodinäited jms abistav materjal on arendaja esmaseks infoallikaks; seepärast on oluline, et info oleks ammendav, loogiliselt üles ehitatud ning ajakohane.
7. Kogukonna aktiivsus. Paljusid avatud lähtekoodiga tarkvaratooteid arendatakse ja täiendatakse üksnes vabatahtlike kogukonna liikmete poolt – seega on kogukonna suurus ja elujõulisus sellelaadsete toodete hulgast valiku tegemiseks üheks peamiseks kriteeriumiks. Aktiivne liikmeskond on tootealaste probleemide lahendamisel väärtuslikuks abiks.
8. Testimisvahendite olemasolu. Testimisvahendite olemasolu on suureks abiks realiseeritud rakenduses puuduste avastamisel.
9. Litsentsi tingimused. Kõik avatud lähtekoodiga tarkvaratooted ei ole tingimata vabavaralised. Võimalus levitada valminud rakendust ka ärilistel eesmärkidel oleks nt alustava ettevõtte puhul heaks kulude kokku hoidmise kohaks.

10. Jõudlus. Jõudlus on üheks peamiseks kasutamismugavust mõjutavaks teguriks, mis omakorda avaldab mõju nii külastatavusele kui ka ettevõtte kasumile [8].

1.5 Võrdluse metoodika

Igas erinevas alampunktis omistatakse võrreldavale raamistikule vastavalt hindamiskriteeriumitele arvuline väärtus, edaspidi punktid, vahemikust 0 – 10. Kui mingis alampunktis pole täidetud ükski tingimus, siis on selle alampunkti eest on tulemus null punkti. Kõigi kümne alampunkti summeerimisel saadakse lõpptulemus skaalal 0 – 100, mille alusel moodustatakse võrreldavatest raamistikest pingerida.

1.5.1 Õppimiskiirus

Õppimiskiiruse hindamise aluseks on testrakenduse realiseerimisele kuluv aeg tundides. Lähtuvalt isiklikust kogemusest hindab autor testrakenduse optimaalseks valmimisajaks viis tundi. Selles alampunktis omistatav punktsumma p leitakse valemist

$$p = \max(0, \min(15 - t, 10)) ,$$

kus t on aeg tundides.

1.5.2 Sobivus erinevate veebilehitsejatega ja standarditele orienteeritus

Toetatavate veebilehitsejate valimisel tuginetakse StatCounteri [15] poolt avaldatud ülemaailmsele statistikale aastal 2012 enimkasutatavate veebilehitsejate osas. Tugi igale järgnevalt loetletud lehitsejale annab ühe punkti:

1. Mozilla Firefox versioonid alates 5.0,
2. Internet Explorer versioonid alates 7.0,
3. Safari versioonid alates 5.0,
4. Google Chrome versioonid alates 14.0,
5. Opera versioonid alates 11.0.

Standardite HTML5 ja CSS3 tugi annab kumbki punktisummasse 2,5 punkti.

1.5.3 Lokaliseerimine

Selles punktis hinnatakse, kas raamistik võimaldab kuvakeele vahetamist ning ka regioonispetsiifilisi formaate (nt numbrid, aja- ja rahaühikud) toetatavate komponentide olemasolu. Kumbki annab punktisummasse viis punkti.

1.5.4 Sisendi valideerimine

Tugi sisendi valideerimist võimaldavatele komponentidele (nt nõutavad väljad, regulaaravaldised jms) annab viis punkti. Juhul, kui on võimalik veateadete kuvamine, siis 10 punkti.

1.5.5 Integreeritud arenduskeskkonna (IDE) tugi

Integreeritud arenduskeskkonna toena käsitletakse spetsiaalselt kogukonna poolt välja arendatud tarkvara või lisamoodulit mõnele kolmanda osapoole poolt arendatavale keskkonnale. Arenduskeskkonna tugi annab 10 punkti.

1.5.6 Kasutajatoe elemendid

Kasutajatoe elementide all hinnatakse järgmisi omadusi:

- Dokumentatsioon on olemas.
- Dokumentatsioon on lihtsasti leitav, st selleni on võimalik jõuda raamistiku veebilehte kasutades.
- Materjalide ülesehitus on loogiline ehk grupeeritud teemade järgi ja jaotatud peatükkideks.
- Dokumentatsioon või muu abistav materjal sisaldab koodinäiteid.
- Kasutaja saab algtekstide muutmise või kommentaaride lisamise abil dokumentatsiooni täiendada.

Iga punkti täidetud annab 2 punkti.

1.5.7 Kogukonna aktiivsus

Kogukonna aktiivsuse all hinnatakse mõne abi küsimiseks mõeldud arutelu keskkonna (foorum, IRC, ...) olemasolu ning viimase 52 nädala keskmist koodihoidlasse (*repository*) tehtud muudatuste arvu. Andmed muudatuste kohta saadakse vastava raamistiku koodihoidlast.

Arutelu keskkonna olemasolu annab viis punkti. Keskmise muudatuste arvu k eest saadavad punktid p leitakse valemist

$$p = \min(k, 5).$$

1.5.8 Testimisvahendite olemasolu

Testimisvahendite olemasolu annab 10 punkti.

1.5.9 Litsentsi tingimused

Litsentsi tingimuste all hinnatakse, kas antud raamistiku abil realiseeritud rakendust tohib levitada ärielistel eesmärkidel, ilma et rakenduse enda aluseks olev lähetekood tuleks avalikustada. Selle tingimuse täidetuse annab 10 punkti.

1.5.10 Jõudlus

Jõudluse hindamisel hinnatakse konkreetse raamistiku põhjal realiseeritud testrakenduse jõudlust YSlow Firefox'i lisa abil. YSlow näol on tegu veebirakenduste optimeerimiseks ja jõudluse hindamiseks mõeldud tarkvaraga, mis mõõdab veebirakenduse vastavust eeldefineeritud kriteeriumitele ning väljastab sisseehitatud hindamisskaala [16] alusel punktisumma vahemikust 0 – 100. Kehtib põhimõte – mida suurem arv, seda parem tulemus.

YSlow's mõõdetakse jõudlust järgmistest kriteeriumitest lähtuvalt:

- Puuduvad Internet Exploreri spetsiifilised CSS avaldised (*CSS expressions*), sest nende liialt sage täitmine on potentsiaalne jõudluse vähendaja. Samuti on need uutes Internet Exploreri versioonides toeta jäetud.
- JavaScript ja CSS koodifailid on vähendatud kujul. Selline tehnika võimaldab vähendada veebirakenduse laadimiseks ja kuvamiseks kuluvat aega.
- DOM (*Document Object Model*) elementide arv. Väga kompleksse ehitusega dokumendid on suure mahuga, mistõttu nende allalaadimisele kuluv aeg suurendab ka lehe kuvamiseks kuluvat aega. Suure hulga elementide läbimine üksiku elemendi poole pöördumiseks on algoritmiliselt keerukam ja seega ka aeganõudvam ülesanne.
- Välditakse tühje väärtusi HTML *src* ja *href* atribuutides. Väga paljud veebilehitsejad käsitlevad tühje internetiaadresse (URL) relatiivsetena ning teevad päringuid kas dokumendi juurkataloogi või veebilehe enda aadressile, raisates nii serveri ressursse.

- Dokument ei tohiks kaasata identseid CSS ja JavaScript faile korduvalt. Leidub lehitsejaid, mis laadivad neid hoolimata kliendipoolsest puhverdamisest korduvalt alla. Sama kehtib ka täitmise kohta.
- Väiksemahulised AJAX päringud tehakse kasutades GET meetodit. GET eelis meetodi POST ees on see, et esimese saatmiseks kulub enamast üksainus TCP pakett, samas kui POST saadab esmalt päringu päised ja seejärel päringu sisu. Meetodi rakendamine aitab saavutada ajalist kokkuhoidu tarbetult võrguliikluselt.

Selle kriteeriumi punktid saadakse YSlow hinde jagamisel kümnega.

1.6 Testrakendus

Saamaks ülevaadet raamistiku pakutavatest võimalustest ning puudustest, realiseeritakse iga raamistiku abil sama ülesehituse ja funktsionaalsusega testrakendus. Testrakenduse välimus ja kasutatavate komponentide valik varieerub sõltuvalt raamistike erisustest ja pakutavatest võimalustest.

Rakenduse sisuks on otsingutulemuste kuvamine vastavalt kasutaja poolt ette antud otsingusõnale kasutades Google'i kohandatud otsingu jaoks mõeldud programmeerimisliidest [17] (Google Custom Search API V1). Testrakendus realiseeritakse ühelehelisena järgides järgmisi põhimõtteid:

- Rakenduse päises antakse kasutatav raamistiku nimetus ja lisaväljana lühike juhend rakenduse kasutamiseks.
- Otsinguvorm ja tulemused on jaotatud kahe vahekaardi abil.
- Otsinguvorm sisaldab pealkirja „Otsing“, tekstivälja otsingusõna sisestamiseks, märkeruutu korduvate tulemuste filtreerimiseks ning otsingu- ja lähtestusnuppu.
- Otsingusõna sisestamiseks kasutatavat tekstivälja täidetust kontrollitakse raamistiku sisendivalideerimise võimalusi kasutades.
- Otsingu sooritamisel lülitatakse otsingutulemuste kaardile ning kuvatakse otsingutulemused.
- Otsingutulemusi kuvatakse lehekülgedel kümne kaupa, kusjuures kuvatakse ka tulemuste koguarv. Iga tulemuse puhul näidatakse lehekülje pealkirja, lühikirjeldust ning veebilehe aadressi. Otsingutulemusel klõpsamine avab vastava lehekülje.
- Lehekülgedel navigeerimine toimub kasutades nuppe „Eelmine“ ja „Järgmine“.

- Pääringute tegemine toimub AJAX GET pääringu abil ning kasutatavaks andmeformaadiks on JSON. Näide pääringust ja vastusest on antud Lisas 1.
- Kasutaja teavitamine pääringutega seonduvate vigade osas toimub raamistiku poolt pakutavaid vahendeid kasutades.
- Otsingu teostamise ja tulemuste laadimise vahepeal kuvatakse kasutajale edenemisinäitur.
- Testrakenduse realiseerimisel järgitakse jõudluse hindamise aluseks toodud kriteeriume.

2. Raamistike võrdlus

2.1 Bootstrap

Bootstrap [9] arendati välja mikrobloginimise lehekülje Twitter arendajate Mark Otto ja Jacob Thorntoni eestvedamisel. Raamistiku kasutuselevõtmise peamiseks ajendiks oli vajaduses ühtlustada Twitteri osadeks olevate eesrakenduste ülesehitust, suurendades nii veebilehe skaleeruvust kui ka aluseks oleva koodibaasi hallatavust [18]. Bootstrap tehti avatud lähtekoodina kättesaadavaks 2011. aasta augustis. Mõned kuud hiljem, 2012. aasta jaanuaris avaldatud versioon 2.0 tõi olulisima uuendusena toe kohanduva disaini põhimõtetele ja muutis raamistiku IT projektide koosarendamise ja majutusteenust pakkuva veebilehe GitHub enim jälgitavaks projektiks [19]. Seisuga aprill 2013 oli Bootstrapil GitHubis üle 48 000 jälgija ja enam kui 14 000 modifitseeritud versiooni.

Bootstrap koosseisu kuuluvad:

- Eeldefineeritud stiililehed, mis annavad välimuse nii raamistiku enda komponentide kui ka teiste enimkasutatavate HTML märgendite jaoks. Siia hulka kuulvad ka stiililehed erinevate veebilehitsejate stiilide vaikeväärtuste lähtestamiseks ning suur valik erinevaid ikoone.
- Võrestiku struktuur ja mõned eeldefineeritud kujundused.
- Kasutajaliidese komponendid nagu vahekaardid, tekstiväljad, mitmesugused menüüd, dialoogaknad jpt. Komponentid defineeritakse deklaratiivselt, st nende kasutamine toimub HTML märgendkeele abil. Enamik komponente on varustatud sündmuse töötlemise võimalustega.

Bootstrap on kergesti kohandatav, tootja lehelt on võimalik alla laadida kõiki põhilisi võimalusi sisaldav komplekt minimeeritud JavaScript ja CSS koodifailidega või teha sama eelnevalt mitmeid parameetreid nagu nt võrestiku laius jt täpsustades. Raamistiku koosseisus ei ole vahendeid MVC ega ka teiste mustrite realiseerimiseks. Seega sobiks see väiksemamahuliste rakenduste nagu nt portfooliode realiseerimiseks või kasutamiseks rakendustes, kus selliste mustrite realiseerimine on võimalik serveripoolsete tehnoloogiatega või teiste eesrakenduste raamistike abil.

2.1.1 Õppimiskiirus

Testrakenduse valmimiseks kulus orienteeruvalt viis tundi, mis langeb kokku käesolevas töös seatud optimaalse tulemusega. Raamistiku kasutajaliidese komponentide kasutamine on deklaratiivse iseloomuga ning on seega igapäevane, kes valdab HTML süntaksit, jõukohane tegevus. Väga suur aeg testrakenduse valmimisest kulus otsingutulemuste kuvamise ja edenemisindikaatori realiseerimiseks, sest selleks otstarbeks raamistikus vahendeid ei leidu. Võrdlusmetoodikast lähtuvalt hinnatakse õppimiskiirust 10 punktiga.

2.1.2 Sobivus erinevate veebilehitsejatega ja standarditele orienteeritus

Toetatud veebilehitsejate osas vastab raamistik kõigile võrdlusmetoodikas väljatoodud nõuetele [20]. Samuti on toetatud CSS3 ja HTML5 kasutamine. Selle punkti koondhindeks on 10 punkti.

2.1.3 Lokaliseerimine

Vaadeldavas Bootstrap versioonis ei ole vahendeid rakenduse lokaliseerimiseks, kuid need on plaanis kaasata tulevastes versioonidesse. Kuivõrd selles punktis ei ole täidetud ükski võrdlusmetoodikas toodud lokaliseerimisega seonduv kriteerium, siis on selle alampunkti eest tulemus null punkti.

2.1.4 Sisendi valideerimine

Testrakenduse realiseerimise käigus ilmnes, et raamistikus sisaldub küll komponendina tekstiväli, mis võimaldab HTML5 atribuudi *required* abil kontrollida välja täidetust, kuid rakenduse omavahendeid kasutades sisendit valideerida, rääkimata regulaaravaldistest ja keerulisematest vormidest, ei ole võimalik. Eelnevast tulenevalt on punktisumma sisendi valideerimise toe eest null punkti.

2.1.5 Integreeritud arenduskeskkonna (IDE) tugi

Kogukonna poolt Bootstrapile integreeritud arenduskeskkonda ega ka liidest mõnele kolmanda osapoole poolt toodetud tarkvarale välja arendatud ei ole. Võrdlusmetoodika põhjal on selles alampunktis omistatav punktisumma null punkti.

2.1.6 Kasutajatoe elemendid

Raamistiku dokumentatsioon on lihtsasti leitav ja grupeeritud teemade järgi. Komponentid ja nende omadused on põhjalikult kirjeldatud, kusjuures iga komponendi puhul on välja toodud koodinäide ja selle tulemus. Hulgased koodnäited olid testrakenduse realiseerimisel väga suureks abiks. Kasutajal endal dokumentatsiooni täiendada ei ole võimalik, seega on selle alampunkti tulemuseks 8 punkti.

2.1.7 Kogukonna aktiivsus

Kogukonna liikmetele on saadaval GitHub¹ keskkond, kus saab raporteerida raamistikus olevatest vigadest ning teha arendusalaseid ettepanekuid. Abi küsimise jaoks ühtki kanalit loodud ei ole, mistõttu on selle kriteeriumi eest tulemuseks null punkti.

GitHub andmetel tehti seisuga 01.04.2013 koodihoidlatesse eelneva 52 nädala jooksul keskmiselt 20,2 muudatust nädalas. Selle kriteeriumi eest omistatav punktisumma on viis punkti.

Selle alampunkti koondindeks on 5 punkti.

2.1.8 Testimisvahendite olemasolu

Raamistikuga ei ole kaasas testimisvahendeid, mistõttu selle alampunkti tulemuseks on null punkti.

2.1.9 Litsentsi tingimused

Bootstrap on tehtud kättesaadavaks Apache litsentsi versioon 2.0 [21] alusel, mis lubab raamistiku abil realiseeritud tarkvara levitada ka ärielistel eesmärkidel, tingimusel, et arendaja säilitab koodifailides autoriõigusega seonduvad materjalid ja kaasab raamistiku litsentsi levitatava tarkvara koosseisu. Rakenduse lähtekoodi ega ka raamistiku modifitseeritud koodi avalikustamise kohustust sellega ei kaasne, seetõttu hindab autor selles alampunktis raamistikku 10 punktiga.

¹ <https://github.com/dojo>

2.1.10 Jõudlus

YSlow väljastab testrakenduse hindeks 100 punkti. Tulemuste objektiivsuse saavutamiseks on autori poolt kirjutatud JavaScript kood minimeeritud. Selle alampunkti hindeks on 10 punkti.

2.1.11 Hindamise tulemused

Eeltoodud alampunktides toodud punktide põhjal omistatakse raamistikule lõpptulemusena 53 punkti. Ülevaade koondpunktisumma kujunemisest on toodud Tabelis 1.

Tabel 1. Raamistiku Bootstrap koondpunktisumma kujunemine.

Õppimiskiirus	Veebilehitsejate tugi ja standarditele orienteeritus	Lokaliseerimine	Sisendi valideerimine	Integreeritud arenduskeskkond
10	10	0	0	0
Kasutajatoe elemendid	Kogukonna aktiivsus	Testimisvahendid	Litsentsi tingimused	Jõudlus
8	5	0	10	10
Kokku				53

2.2 Dojo

Dojo [10] raamistik loodi aastal 2004 Alex Russelli, David Schontzler jt poolt [22] eesmärgiga lihtsustada mitmeplatvormiliste JavaScript ja AJAX tehnoloogiatel põhinevate veebi-rakenduste loomist. Käesolevaks ajaks on raamistiku arendamine koondunud mittetulundusliku organisatsiooni Dojo Foundation alla, mille tegevusse panustavad sellised firmad nagu Google, IBM, Sun, Zend jpt [23].

Raamistik jaguneb neljaks paketiks:

- Dojo – See pakett katab põhilise raamistiku tuumikfunktsionaalse sisaldades mooduleid AJAX päringute tegemiseks ning DOM objektide ja meetoditega ümberkäimiseks, objektihierarhiat, vahendeid andmestruktuuride haldamise ja lokaliseerimise jaoks jpm.
- Dijit – Teek kasutajaliideste komponentide jaoks, mille koosseisu kuuluvad ka paigutuskomponendid ning ARIA (*Accessible Rich Internet Applications*) põhimõtetele vastavad vahendid ligipääsetavuse (*accessibility*) suurendamiseks.

- Dojox – Sisaldab endas eksperimentaalseid mooduleid ja teeke. See pakett eemaldatakse versioonist 2.0.
- Util – Paketis on vahendid raamistiku kohandamiseks (nt minimeeritud pakettide ehitamine) ja testimiseks ning ka rakendusprogramm kontrollimaks arendatavas rakenduses koodi vormindamise konventsioonidest kinnipidamist.

Raamistiku üheks peamiseks tugevuseks on ADM [24] (*The Asynchronous Module Definition*) liidesele vastav modulaarne disain. Kõik raamistiku komponendid ja funktsioonid on jaotatud moodulitesse või teekidesse, mille laadimine toimub asünkroonselt. Selline lähenemisviis võimaldab laadida korraga minimaalse vajaliku moodulite komplekti ning hoida seega kokku tarbetult võrguliikluselt ning vähendada rakenduse laadimiseks kuluvat viivitust. Samuti parendab modulaarsus koodibaasi hallatavust. Ka arendaja poolt kirjutatud kood peaks olema defineeritud moodulite või teekidena.

Raamistiku komponendid baseeruvad objektide hierarhial, millest klasse pärides on võimalik luua kohandatud komponente. Kasutajaliideste ehitamisel on võimalik kasutada objektorienteeritud või deklaratiivset lähenemist. Ka viimasel juhul on võimalik nii komponentidele kui objektidele juurde pääseda. Kasutajaliideste komponentidest on toetatud nii standardsed (tekstiväljad, vahekaardid, märgendruudud jms) kui ka andmeteadlikud (*data aware*) komponendid, mis võimaldavad hõlpsasti andmeid visualiseerida (tabelid, vahendid graafikute kuvamiseks). Raamistikus on olemas mitmeid erinevaid kergesti kohandatavaid stiile.

Raamistiku lai valik mooduleid ning lisavõimalusi teevad sellest ideaalse vahendi üheleheliste HTML5 või RIA (*Rich Internet Application*) tüüpi rakenduste arendamiseks.

2.2.1 Õppimiskiirus

Testrakenduse valmimiseks kulus orienteeruvalt üheksa tundi. Dojo näol on tegu väga suurel hulgal funktsionaalsusust pakuva raamistikuga, mistõttu ligikaudu pool testrakenduse valmimise ajast kulus dokumentatsiooniga tutvumisele. Andmekonteinerite ja andmeteadlike komponentide olemasolu oli otsingutulemuste kuvamisel oluliseks abiks. Raamistiku koosseisus olid kõik testrakenduse realiseerimiseks vajaminevad komponendid. Võrdlusmetoodika põhjal on selles alampunktis omistatav punktisumma kuus punkti.

2.2.2 Sobivus erinevate veebilehitsejatega ja standarditele orienteeritus

Toetatud veebilehitsejate osas vastab raamistik kõigile võrdlusmetoodikas väljatoodud nõuetele. Samuti on toetatud CSS3 ja HTML5 kasutamine.

Selles alampunktis omistatav punktisumma on 10 punkti.

2.2.3 Lokaliseerimine

Dojo võimaldab defineerida erinevate keelte jaoks vastavakeelseid sõnade kogumeid (*resource bundle*), mille komponentideks on paarid, mis koosnevad mingitele komponentidele omastest atribuutidest ja nende tõlgetest. Tõlgitud massiivid paigutatakse eeldefineeritud kausta, kust neid saab vastavate moodulite abil laadida. Raamistikus on olemas ka regioonispetsiifilisi numbri-, valuuta- ja kuupäevaformaate toetavad komponendid. Samuti on olemas ka paremalt-vasakule kirjutatavate keelte tugi.

Dojo puhul on täidetud kõik võrdlusmetoodikas toodud lokaliseerimisega seonduvad kriteeriumid ning seega on alampunkti koondhindeks 10 punkti.

2.2.4 Sisendi valideerimine

Raamistikus on olemas tugi sisendi valideerimist toetavatele komponentidele. Toetatakse mõlemaid – nii raamistiku enda komponentide kui ka tavaliste HTML vormide valideerimist. Paketis Dojox on olemas sisendi kontrollimiseks vastav moodul, samuti saab komponentidele atribuutidena seada mustreid, regulaaravaldisi ning ka veateadete tekste. Veateated on lokaliseeritavad ning nende kuvamise eest kasutajale hoolitseb raamistik enamasti ise.

Kuivõrd raamistikus on olemas vahendid nii sisendi valideerimiseks kui ka sellesisuliste veateadete kuvamiseks, siis on selle alampunkti tulemuseks 10 punkti.

2.2.5 Integreeritud arenduskeskkonna (IDE) tugi

Dojo on toetatud erinevate arenduskeskkondade poolt, kusjuures enamik nende arendajatest panustavad ka ise Dojo projekti. Raamistiku tugi on järgnevates arenduskeskkondades: IBM RAD 8.0.3, Aptana Studio 3 ja WebStorm 2.1. Koodi silumiseks on saadav lisa veebilehitsejale Firefox.

Integreeritud arenduskeskkonna tugi annab koondhindeksse 10 punkti.

2.2.6 Kasutajatoe elemendid

Dokumentatsioon koosneb kasutusjuhendist ja klassideklaratsioonide kirjeldusest. Mõlemad on lihtsasti leitavad, jaotatud vastavalt raamistiku versioonidele, loogiliselt ülesehitatud ning grupeeritud vastavalt pakettidele. Raamistiku kasutamiseks on palju praktilisi õpetusi, mis katavad enamiku rakendusvaldkondadest. Õpetusi saavad kasutajad ka ise koostada. Kasutusjuhend kui ka õpetused on varustatud koodinäidetega, kusjuures välja on toodud ka koodi tulemus.

Dokumentatsiooni osas on täidetud kõik võrdlusmetoodikas seatud kriteeriumid ning seega on selle alampunkti koondhindeks 10 punkti.

2.2.7 Kogukonna aktiivsus

Lisaks GitHub² keskkonnale on nii kogukonna kui ka teistele liikmetele kättesaadav foorum, kus tekkinud probleemidele abi saada. Samuti on olemas ka meililist ning IRC kanal. Võrdlusmetoodikast lähtuvalt annab nende kanalite olemasolu viis punkti.

GitHub andmetel tehti seisuga 01.04.2013 koodihoidlasesse eelneva 52 nädala jooksul keskmiselt 9,5 muudatust nädalas. Selle kriteeriumi eest omistatav punktisumma on viis punkti.

Selle punkti koondhindeks on 10 punkti.

2.2.8 Testimisvahendite olemasolu

Dojo pakettis Util on olemas vahendid moodulite jm funktsionaalsuse testimiseks. Toetatud on ühiktestid kui ka rakenduse jõudluse testimine. Testimine toimub veebilehitsejat või JavaScripti jaoks mõeldud süntaksi-analüsaatorit Mozilla Rhinot kasutades. Viimase lihtsustamiseks on raamistikku kaasatud kestaskript. Testimisvahendite olemasolu annab koondhindeksesse 10 punkti.

2.2.9 Litsentsi tingimused

Raamistikku põhjal realiseeritud tarkvara tohib Academic Free Licence versioon 2.1 [25] kohaselt levitada ka ärielistel eesmärkidel, tingimusel, et arendaja säilitab koodifailides autoriõigusega seonduvad materjalid. Rakenduse lähtekoodi või raamistiku modifitseeritud

² <https://github.com/dojo>

osade aluseks oleva koodi avalikustamise kohustust litsents ette ei näe, seega on selles alampunktis omistatav punktisumma 10 punkti.

2.2.10 Jõudlus

Testrakendus jõudluse mõõtmisel YSlow abil saadi tulemuseks 100st punktist 97 punkti, sest raamistiku komponentide poolt kasutatavas stiililehes sisaldasid Internet Exploreri spetsiifilised avaldised. Punktisumma jõudluse eest on 9,7 punkti.

2.2.11 Hindamise tulemused

Eeltoodud alampunktides toodud punktide põhjal omistatakse raamistikule lõpptulemusena 95,7 punkti. Ülevaade koondpunktisumma kujunemisest on toodud Tabelis 2.

Tabel 2. Raamistiku Dojo koondpunktisumma kujunemine.

Õppimiskiirus	Veebilehitsejate tugi ja standarditele orienteeritus	Lokaliseerimine	Sisendi valideerimine	Integreeritud arenduskeskkond
6	10	10	10	10
Kasutajatoe elemendid	Kogukonna aktiivsus	Testimisvahendid	Litsentsi tingimused	Jõudlus
10	10	10	10	9,7
Kokku				95,7

2.3 YUI

YUI [11] raamistik arendati välja aastal 2005 kasutamiseks mitmetes Yahoo! korporatsioonile kuuluvates veebirakendustes. Aasta hiljem tegi Yahoo!, kelle töötajad on ka praegu YUI põhilised arendajad, raamistiku avatud lähtekoodina kättesaadavaks [26].

Raamistikku iseloomustab modulaarne disain – enamik funktsionaalsusest on jaotatud moodulitesse ja teekidesse. Paljud arendajatele vajalikud ent YUI tuumikust puudu olevad võimalused on realiseeritud kogukonna poolt sadades vabalt kättesaadavates moodulites. Raamistiku võib tinglikult jagada järgmisteks osadeks:

- YUI Core ehk raamistiku tuum sisaldab mooduleid DOM objektidega ümberkäimiseks ning sündmuste töötlemiseks. Raamistik toetab ka uute sündmuste defineerimist, nt kolmekordse hiireklõpsu jaoks. Need moodulid on enamasti kaasatud igasse raamistiku abil realiseeritud rakendusse.

- YUI objektihierarhia moodulite ja kasutajaliideste vidinate (*widget*) loomiseks. Raamistikus on olemas vahendid üheleheliste MVC mustri põhinevate rakenduste loomiseks – nt klassid aadressi põhjal vaadete laadimiseks ning vaadete ja mudelite haldamiseks.
- CSS stiililehed kasutajaliidese komponentide ja vidinate kujundamiseks, veebilehitsejate vaikestiilide lähtestamiseks ning ühtlustamiseks. Samuti toetab raamistik võrestikke.
- Arendajatele mõeldud tööriistad rakenduse analüüsimiseks ning testimiseks. Raamistikus on konsoolilaadne vidin veateadete ja teiste diagnostikasõnumite kuvamiseks. Testimine on brauseripõhine ning seisneb ühiktestide jaoks mõeldud klasside pärimises ja täiendamises. Testitulemusi on võimalik kuvada spetsiaalselt selle jaoks kohandatud konsoolvidina abil, XML või JSON formaadis väljundina veebilehitsejas või edastada see mõnele teisele veebi- või tagarakendusele.
- Vidinad. Raamistikus puuduvad objektitasemel vasted HTML märgendkeeles kasutatavatele komponentidele nagu tekstiväljad, märgendruudud jne, nende omaduste ja sündmuste töötlemine toimub sarnaselt teiste DOM objektidega. YUI vidinate valik hõlmab enimlevinuid raskemini realiseeritavaid komponente nagu näiteks ennetava tekstisisestusega tekstiväljad, kalender, andmetabelid, graafikud, puuteekraanide jaoks mõeldud navigeerimisvidinad jpm.
- Mitmesugused vahendid veebirakenduste realiseerimise lihtsustamiseks nt moodulid AJAX ja JSONP (*JSON with padding*) päringute tegemiseks, animatsioonide ja efektide kuvamiseks, ajaloo ja puhvermälu haldamiseks, andmeallikate (*data store*) ja -skeemide koostamiseks jt.

Raamistiku kasutamiseks piisab rakendusse kaasata minimeeritud alglaaduri (*seed*) funktsiooni täitev skript, ülejäänud vajalikud moodulid laetakse asünkroonselt. YUI on kasutatav ka Google'i CDN (*Content Delivery Network*) abil. Vähesed konventsioonid ja suurt hulka erinevaid funktsioone pakkuvad moodulid teevad raamistikust hea vahendi mistahes veebirakenduste realiseerimiseks.

2.3.1 Õppimiskiirus

Testrakenduse realiseerimiseks kulus kokku kuus tundi. Kõik raskemini realiseeritavad komponendid (vahekaardid, andmeteadlik tabel) olid raamistiku koosseisus olemas. HTML

märgendkeeles kasutatavate komponentide kasutamine aitas õppimise arvelt aega kokku hoida. Võrdlusmetoodikast lähtuvalt hinnatakse õppimiskiirust üheksa punktiga.

2.3.2 Sobivus erinevate veebilehitsejatega ja standarditele orienteeritus

YUI toetab võrdlusmetoodikas välja toodud veebilehitsejatest kõiki peale Opera, kus erinevate moodulite töökindlus ei ole garanteeritud. Brauserite toe eest omistatakse raamistikule neli punkti.

Raamistikus on toetatud CSS3 ja HTML5 kasutamine, mis annab punktisummase 5 punkti. Selle võrdluskriteeriumi koondindeks on üheksa punkti.

2.3.3 Lokaliseerimine

Raamistikus on kuvakeele vahetamine realiseeritav vastavatesse keeltesse tõlgitud sõnekogumite (*resource bundle*) abil, mille laadimine toimub selleks otstarbeks mõeldud moodulit kasutades. Regioonispetsiifilisi numברי-, valuuta- ja kuupäevaformaate toetavaid komponente raamistikus ei ole ning need tuleb arendajal endal programmeerida.

Selle alampunkti koondindeks on viis punkti.

2.3.4 Sisendi valideerimine

YUI vahendite hulgas ei ole sisendivalideerimisega varustatud komponente. Võrdlusmetoodika põhjal on selles alampunktis omistatav punktisumma null punkti.

2.3.5 Integreeritud arenduskeskkonna (IDE) tugi

Kogukonna poolt raamistikule integreeritud arenduskeskkonda ega ka liidest mõnele kolmanda osapoole poolt toodetud tarkvarale välja arendatud ei ole. Seega on selle alampunkti eest tulemus null punkti.

2.3.6 Kasutajatoe elemendid

YUI dokumentatsioon koosneb moodulite kasutusjuhendist, mitmesugustest veebirakendustes vajaminevate ülesannete realiseerimise õpetustest ja nende aluseks olevatest koodinäidistest, klassideklaratsioonide kirjeldustest ning algajatele kasutajatele mõeldud õppematerjalidest. Dokumentatsioon on lihtsasti leitav ning grupeeritud moodulite kaupa. Kasutusjuhend on olemas kõigi raamistiku koosseisu kuuluvate moodulite jaoks ning

enamasti ilmestatud koodinäidetega. Registreerunud kasutajad saavad kasutusjuhendit täiendada, lisada koodinäiteid ning praktilisi õpetusi.

Võrdlusmetoodikast lähtuvalt hinnatakse kasutajatoe elemente 10 punktiga.

2.3.7 Kogukonna aktiivsus

YUI kogukonnale on lisaks GitHub³ keskkonnale kasutada foorum ning IRC kanal. Kogukonna poolt arendatud moodulid, mis ei kuulu raamistiku koosseisu, on galerii lehel ühendatud vastavasisuliste sissekannetega foorumis, et abi leidmine nende kasutamisel tekkivate probleemide korral oleks lihtsam. Arutelu keskkonna olemasolu annab 5 punkti.

GitHub andmetel tehti seisuga 01.04.2013 koodihoidlatesse eelneva 52 nädala jooksul keskmiselt 62,7 muudatust nädalas. Keskmise muudatuste arvu eest omistatav punktisumma on viis punkti.

Selle alampunkti koondhindeks on 10 punkti.

2.3.8 Testimisvahendite olemasolu

Raamistikus on olemas vahendid ühiktestide koostamiseks ning nende käivitamiseks. Testide tulemusi on võimalik kuvada brauseri aknas või konsoolis XML või JSON väljundina. Samuti on võimalik genereeritud väljundit edastada POST päringuga mõnele teisele veebi- või tagarakendusele. Testimisvahendite olemasolu annab koondhindeksesse 10 punkti.

2.3.9 Litsentsi tingimused

YUI on tehtud kättesaadavaks BSD litsentsi alusel [27], mis lubab raamistiku abil realiseeritud tarkvara levitada ka ärilistel eesmärkidel, tingimusel, et autor säilitab koodifailides autoriõigusega seonduvad materjalid ning kaasab raamistikuga kaasasoleva litsentsi levitatava tarkvara koosseisu. Litsents keelab levitatavat tarkvara reklaamida ja kiita kasutades Yahoo! kaubamärki või mõne teise arendaja nime, vastavalt ilma Yahoo! või arendaja eelneva kirjaliku nõusolekuta. Rakenduse lähtekoodi avalikustamise kohustust litsents ette ei näe.

Selle alampunkti tulemuseks on 10 punkti.

³ <https://github.com/yui/yui3>

2.3.10 Jõudlus

YSlow hindab testrakendust 100 punktiga. Võrdlusmetoodikast lähtuvalt on selle alampunkti hindeks 10 punkti.

2.3.11 Hindamise tulemused

Eeltoodud alampunktides toodud punktide põhjal omistatakse raamistikule lõpptulemusena 73 punkti. Ülevaade koondpunktisumma kujunemisest on toodud Tabelis 3.

Tabel 3. Raamistiku YUI koondpunktisumma kujunemine.

Õppimiskiirus	Veebilehitsejate tugi ja standarditele orienteeritus	Lokaliseerimine	Sisendi valideerimine	Integreeritud arenduskeskkond
9	9	5	0	0
Kasutajatoe elemendid	Kogukonna aktiivsus	Testimisvahendid	Litsentsi tingimused	Jõudlus
10	10	10	10	10
Kokku				73

2.4 SproutCore

SproutCore [12] loodi aastal 2006 ettevõtte SproutIt arendatava kasutajatoe tarkvara Mailroom tarbeks ning on mõeldud töölaarakendustele sarnanevate skriptimiskeelel JavaScript põhinevate veebirakenduste loomiseks. Raamistik tehti avalikult kättesaadavaks aastal 2008 [28]. SproutCore on leidnud kasutust tehnoloogiafirma Apple veebirakendustes iCloud ja iWork [29].

Raamistik paigaldatakse skriptimiskeele Ruby moodulina (Ruby Gem), kasutades selleks paketihooldurit RubyGems. Viimase abil toimib ka raamistiku uuendamine. Paigaldatuna on arendajale kasutamiseks käsureatööriist, mille abil saab hõlpsasti genereerida raamistiku projektide ja rakenduste aluseks oleva struktuuri ning arenduse lõppfaasis kombineerida lähtekood terviklikuks rakenduseks. Tervikrakenduse ehitamiseks on nii projekt kui ka rakendused varustatud vastava konfiguratsioonifailiga, milles on võimalik määrata sätteid, nt CSS ja JavaScript failide minimeerimiseks, lisateekide kaasamiseks jms. Lisaks käsureatööriistale on raamistiku koosseisus ka arendusserveri tarkvara ning testimisvahendid.

SproutCore'i eripäraks on olekuskeemidel põhinevad MVC rakendused. Igal rakendusel on teatud hulk olekuid, kusjuures igas olekus on rakenduses kasutatavad ainult teatavad

funktsioonid. Selline lähenemine aitab hoida arendatavas rakenduses vigade arvu minimaalsena, sest erinevad rakenduse osad on üksteisest isoleeritud. Teiseks raamistiku tunnusjooneks on laialdane sidemete (*binding*) ja vaatleja mustri (*Observer Pattern*) kasutamine. Sidemed tähendavad antud raamistiku kontekstis objektide vahelist ühe- või kahepoolset teavituse mehhanismi, mis objekti osaks olevate väärtuste muutumisel toovad olenevalt sideme tüübist kaasa teise objekti väärtuse muutumise või mingi funktsiooni või protseduuri väljakutse jälgival objektil.

Raamistik ja selles programmeerimine põhineb objektorienteeritud tehnikal. Objektihierarhiast klasse pärides on võimalik koostada kohandatud komponente. Kasutajaliideste komponentidest on raamistikus olemas kõik levinuimad nagu nt tekstiväljad, nupud, dialoogaknad jms. Samuti on esindatud andmeteadlikud komponendid, mille sidemete (*binding*) abil mudeliga ühendamine võimaldab vähese tööga andmete kiiret visualiseerimist. Komponentide ja rakenduse välimus on kohandatav SCSS (*Syntactically Awesome Stylesheets*) keelel baseeruvate stiilide või teemade abil.

2.4.1 Õppimiskiirus

Testrakenduse valmimiseks kulus orienteeruvalt 14 tundi. Raamistikus kasutatavad võtted on kohati keerulised õppida ning selgitav materjal nende kohta napp. Enamik testrakenduse realiseerimiseks kuluvast ajast moodustaski otsingumootoritest info otsimine, sest valdav osa raamistiku funktsionaalsusest on ammendava kasutusjuhendita.

Raamistikus olid testrakenduse realiseerimiseks kõik vajaminevad komponendid. Kuivõrd dokumentatsioon on andmeallikate (*data store*) kasutamise seisukohast ebapiisav, tuli otsingutulemuste kuvamine realiseerida andmeteadliku tabeli asemel teiste komponentide abil, nagu ka navigeerimisloogika tulemuste vahel liikumiseks.

Võrdlusmetoodika põhjal omistatakse raamistikule õppimiskiiruse eest üks punkt.

2.4.2 Sobivus erinevate veebilehitsejatega ja standarditele orienteeritus

Raamistik toetab kõiki võrdlusmetoodikas väljatoodud veebilehitsejaid. Samuti on toetatud CSS3 ja HTML5 kasutamine.

Selle punkti koondhindeks on 10 punkti.

2.4.3 Lokaliseerimine

SproutCore toetab kuvakeelte vahetamist eeldefineeritud tõlgete abil. Asukoha spetsiifilisi formaate toetavaid komponente raamistikus ei ole, vastav funktsionaalsus tuleb arendajal endal programmeerida. Võrdlusmetoodikast lähtuvalt on raamistikule omistatav punktisumma lokaliseerimise eest viis punkti.

2.4.4 Sisendi valideerimine

Enamik raamistikus olevaid komponente pärivad objektihierarhiast valideerimisfunktsionaalsuse. Komponente saab valideerida ükshaaval või terve vormi kaupa. Veateadete kuvamine tuleb arendajal endal programmeerida, mistõttu selle alampunkti tulemuseks on viis punkti.

2.4.5 Integreeritud arenduskeskkonna (IDE) tugi

Raamistikule kogukonna poolt integreeritud arenduskeskkonda ega ka liidest mõnele kolmanda osapoole poolt toodetud tarkvarale välja arendatud ei ole. Seega on selle alampunkti eest tulemus null punkti.

2.4.6 Kasutajatoe elemendid

Raamistiku dokumentatsioon on lihtsasti leitav ning koosneb teemade järgi grupeeritud õpetustest ja klassideklaratsioonide kirjeldusest. Õpetused tutvustavad ühe konkreetse näite varal raamistiku põhifunktsionaalsust, kuid jätavad väga paljud raamistikus olevad klassid ja nende kasutamise tähelepanu alt välja. Õpetused sisaldavad koodinäiteid ning neid saab kasutaja ka ise lisada.

Selle alampunkti koondhindeks on 10 punkti.

2.4.7 Kogukonna aktiivsus

Abi raamistiku kasutamisel tekkinud probleemidele on võimalik saada spetsiaalsest Google'i grupist. Selle kanali olemasolu annab koondpunktisummasse viis punkti.

GitHub andmetel tehti seisuga 01.04.2013 koodihoidlatesse eelneva 52 nädala jooksul keskmiselt 8,4 muudatust nädalas. Selle kriteeriumi eest omistatav punktisumma on viis punkti.

Kokku on alampunkti koondhindeks 10 punkti.

2.4.8 Testimisvahendite olemasolu

Raamistikus on realiseeritud funktsionaalsuse testimiseks vahendid ühiktestide koostamiseks, mille käivitamine toimub brauserit ja paigaldatud arendamisserveri tarkvara kasutades. Testimisvahendite olemasolu annab koondpunktisummasse 10 punkti.

2.4.9 Litsentsi tingimused

SproutCore on tehtud kättesaadavaks MIT litsentsi [30] abil, mis lubab raamistiku abil realiseeritud tarkvara levitada ka ärilistel eesmärkidel, tingimusel, et autor säilitab koodifailides autoriõigusega seonduvad materjalid ning kaasab raamistikuga kaasasoleva litsentsi levitatava tarkvara koosseisu.

Selle alampunkti tulemuseks on 10 punkti.

2.4.10 Jõudlus

Testrakenduse jõudluse hindamisel YSlow abil oli tulemuseks 100 punkti, mis võrdlusmetoodikast tulenevalt annab koondpunktisummasse 10 punkti.

2.4.11 Hindamise tulemused

Eeltoodud alampunktides toodud punktide põhjal omistatakse raamistikule lõpptulemusena 71 punkti. Ülevaade koondpunktisumma kujunemisest on toodud Tabelis 4.

Tabel 4. Raamistiku SproutCore koondpunktisumma kujunemine.

Õppimiskiirus	Veebilehitsejate tugi ja standarditele orienteeritus	Lokaliseerimine	Sisendi valideerimine	Integreeritud arenduskeskkond
1	10	5	5	0
Kasutajatoe elemendid	Kogukonna aktiivsus	Testimisvahendid	Litsentsi tingimused	Jõudlus
10	10	10	10	10
Kokku				71

2.5 Qooxdoo

Qooxdoo [13] on mitmeotstarbeline JavaScript raamistik, milles on vahendid üheleheliste RIA tüüpi, mobiilseadmetes töötavate ja DOM funktsionaalsust vajavate serverirakenduste loomiseks. Raamistik loodi 2004. aastal tarkvaraettevõtete 1&1 Internet ja United Internet

poolt [31], kelle töötajad on ka täna raamistiku võtmetähtsusega arendajateks. Raamistik tehti avatud lähekoodina kättesaadavaks 2005. aastal.

Qooxdoo jaguneb viieks osaks:

- Raamistiku aluseks olev objektihierarhia ehk tuum, milles on tugi mitmetele OOP konstruktsioonidele nagu nt liidesed, abstraktsed ja staatilised klassid, automaatselt genereeritavate piilumeetoditega klassimuutjad jt. Qooxdoo objektihierarhiast päritud klassimuutjad toetavad sidemete (*bindings*) kasutamist, mis lihtsustavad oluliselt MVC muustril põhinevate rakenduste realiseerimist. Viimase tarbeks on raamistiku tuumikus saadaval klassid kontrolleri- ning andmeallikate ja -mudelite jaoks. See osa raamistikust on kogu ülejäänud Qooxdoo osade aluseks.
- Veebilehtede realiseerimiseks mõeldud teek, milles on vahendid DOM objektidega ümberkäimiseks ning nendega seotud sündmuste töötlemiseks, CSS atribuutide muutmiseks, mallide haldamiseks, andmetöötlemiseks ning suhtluseks tagarakenduste või veebiteenustega.
- Raamistik töölaarakenduste sarnaste RIA tüüpi üheleheliste rakenduste loomiseks. Raamistiku objektihierarhias on defineeritud klassid mitmesuguste enimkasutatavate kasutajaliidese komponentide jaoks. Kohandatud komponentide loomiseks piisab vaid eelmainitud klasse laiendada. Liideselementide paigutamiseks on raamistikus staatilise või dünaamilise iseloomuga paigutushaldurid. See osa raamistikust on võetud käesolevas töös testrakenduse realiseerimise aluseks.
- Raamistik veebirakenduste loomiseks mitmesugustele pihuseadmetele, milles sisalduvad spetsiaalselt mobiilseadmetele optimeeritud ent raamistiku tavalistele komponentidele sarnase programmeerimisliidese ga vidinad ning vahendid puutekraanidele omaste sündmuste ja ekraaniviibete töötlemiseks.
- Vahendid JavaScriptil põhinevate serverirakenduste realiseerimiseks. Qooxdoo toetab Node.js ja Mozilla Rhino keskkondi.

Raamistikus olevate koheselt kasutusvalmis vidinate valik sisaldab lisaks põhilistele enimkasutatavatele komponentidele (tekstiväljad, märkeruudud jt) ka keerulisemaid ning andmeallikate toega komponente nagu nt WYSIWYG (*What-You-See-Is-What-You-Get*) tekstiredaktor, andmetabel, dialoogaknad, kontekstimenüüd, valideeritavad vormid jpm. Komponentide ühendamine sidemete abil omavahel või andmeallikatega võimaldab

lihtsasti teostada protseduure erinevate valideerimisülesannete (komponentide deaktiveerimine jmt) ning andmeteadlike komponentide värskendamise jaoks.

Qooodoo arenduspaketti on kaasatud mitmesuguseid arendusprotsessi lihtsustavaid tööriistu, mille abil saab genereerida raamistiku projektide ja rakenduste aluseks oleva struktuuri, klassi deklaratsioone, valideerida lähtekoodi, kuvada testitulemusi jmt. Tööriistad jagunevad brauseri- ja käsuraapõhisteks. Käsuraatööriistade kasutamiseks on vajalik mõne Pythoni baasil töötava programmeerimiskeskonna paigaldamine.

2.5.1 Õppimiskiirus

Testrakenduse valmimisele kulus kokku 10 tundi, millest valdav osa kulus dokumentatsiooniga tutvumiseks. Hoolimata raamistiku võimalusterohkusest on selle kasutama õppimine tänu põhjalikule dokumentatsioonile ja seda ilmestavatele koodinäidetele oluliselt lihtsam võrreldes mõne teise töös käsitletud raamistikuga.

Enamus testrakenduse realiseerimiseks vajaminevatest komponentidest oli raamistikus olemas. Edenemisindikaator ja veateadete kuvamiseks mõeldud dialoogaken tuli realiseerida sarnaste komponentide aluseks olevate klasside laiendamise teel.

Võrdlusmetoodikast lähtuvalt hinnatakse õppimiskiirust viie punktiga.

2.5.2 Sobivus erinevate veebilehitsejatega ja standarditele orienteeritus

Raamistik toetab kõiki võrdlusmetoodikas nõutud brausereid, samuti on toetatud HTML5 ja CSS3 kasutamine.

Selle alampunkti koondindeks on 10 punkti.

2.5.3 Lokaliseerimine

Raamistik toetab kuvakeelte vahetamist läbi vastavakeelsete tõlgete laadimise PO (*Portable Object*) failidest. Tõlkefailide koostamiseks on raamistiku koosseisus vastav käsuraaskript. Samuti toetavad põhilised Qooodoo komponendid regioonispetsiifilisi formaate.

Selles alampunktis omistatav punktisumma on 10 punkti.

2.5.4 Sisendi valideerimine

Enamus kasutajaliidese komponente pärivad valideerimisvõimaluse raamistiku objektihierarhiast ning omavad seega ühtset valideerimisliidest sh klassimuutujat veateadete tekstide jaoks. Valideerida saab üksikuid komponente või ka terveid vorme, kusjuures toetatud on ka sidemete (*bindings*) kasutamine. Komponentid on varustatud veateadete kuvamisvõimalustega.

Selle alampunkti koondhindeks on 10 punkti.

2.5.5 Integreeritud arenduskeskkonna (IDE) tugi

Kogukonna poolt on arenduskeskkonnale Eclipse välja arendatud lisamoodul, mille arendustööd toetav funktsionaalsus seisneb mitmesugustes mallides ja ennetavas koodisisestuses.

Selle tingimuse täidetuse annab koondpunktisummale 10 punkti.

2.5.6 Kasutajatoe elemendid

Qooxdoo dokumentatsioon koosneb kasutusjuhendist ja klassideklaratsioonide kirjeldusest, mõlemad on lihtsasti leitavad. Kasutusjuhend on grupeeritud raamistiku funktsionaalsete osade järgi, on äärmiselt põhjalik ning sisaldab hulgaliselt koodinäiteid. RIA rakenduste ehituste näitlikustamiseks on arendajatele õpetus, mis hõlmab tervet arendustsükli kasutajaliidese ehitamisest testimiseni. Dokumentatsioon kui ka õpetused on kasutajate poolt täiendatavad.

Raamistiku veebilehel on interaktiivne galerii (*playground*) kasutajaliideste komponentidest ja nende kasutusvõimalustest, iga näidise aluskood on nähtav ja muudetav.

Võrdlusmetoodikast lähtuvalt hinnatakse kasutajatoe elemente 10 punktiga.

2.5.7 Kogukonna aktiivsus

Tekkinud probleemidele abi saamiseks, ettepanekute tegemiseks ning üldiseks aruteluks on arendajatele kasutamiseks foorum, IRC kanal ning mitmed postiloendid. Nende kanalite olemasolu annab koondpunktisummale viis punkti.

GitHub⁴ andmetel tehti seisuga 01.04.2013 koodihoidlatesse eelneva 52 nädala jooksul keskmiselt 52,3 muudatust nädalas. Selle kriteeriumi eest omistatav punktisumma on 5 punkti.

Kokku on alampunkti koondhindeks 10 punkti.

2.5.8 Testimisvahendite olemasolu

Qooxdoo koosseisus on raamistik ühiktestide koostamiseks, mis koosneb ühiktestide aluseks olevatest klassidest ning brauseripõhisest testimiskeskonnast. Testimaks kasutajaliidese funktsionaalsust saab raamistiku enda klasside abil kirjutada teste Selenium serveri jaoks.

Testimisvahendite osas on täidetud kõik võrdlusmetoodikas toodud kriteeriumid ning seega on alampunkti koondhindeks 10 punkti.

2.5.9 Litsentsi tingimused

Qooxdoo on tehtud kättesaadavaks GNU LGPL (*Lesser General Public License*) litsentsi alusel [32], mis lubab raamistiku abil realiseeritud tarkvara levitada ka ärilistel eesmärkidel.

Kui rakendus üksnes kasutab raamistikku, siis rakenduse enda lähtekoodi avalikustamist litsents ei nõua. Küll aga tuleb kasutajatele tagada võimalus saada rakenduse aluseks oleva raamistiku lähtekood. Samuti näeb litsents ette, et raamistiku kasutamisel tuleb viidata raamistikule ja seda arendavale kogukonnale.

Selle alampunkti tulemuseks on 10 punkti.

2.5.10 Jõudlus

Testrakenduse jõudluse hindamisel YSlow abil oli tulemuseks 100 punkti, mis võrdlusmetoodika põhjal teeb alampunkti tulemuseks 10 punkti.

2.5.11 Hindamise tulemused

Eeltoodud alampunktides toodud punktide põhjal omistatakse raamistikule lõpptulemusena 95 punkti. Ülevaade koondpunktisumma kujunemisest on toodud Tabelis 5.

⁴ <https://github.com/qooxdoo/qooxdoo>

Tabel 5. Raamistiku Qooxdoo koondpunktisumma kujunemine.

Õppimiskiirus	Veebilehitsejate tugi ja standarditele orienteeritus	Lokaliseerimine	Sisendi valideerimine	Integreeritud arenduskeskkond
5	10	10	10	10
Kasutajatoe elemendid	Kogukonna aktiivsus	Testimisvahendid	Litsentsi tingimused	Jõudlus
10	10	10	10	10
Kokku				95

2.6 Enyo

Enyo [14] arendas välja tehnoloogiafirma HP eesmärgiga kasutada seda operatsioonisüsteemis WebOS rakenduste ehitamiseks [33]. Toode on disainitud riistvarast sõltumatuks, kuid sisaldab siiski mõningaid kohandusi HP WebOS ja seda kasutavate seadmete (HP TouchPad) jaoks. Käesolevaks ajaks on Enyo saadaval avatud lähtekoodina ning seda arendatakse kui universaalset veebirakenduste raamistikku, mis töötaks võimalikult paljude laialdaselt kasutusel olevate lehitsejatega.

Enyo on disainitud modulaarsena, st raamistiku keskseks ehitusblokiks on komponendid. Komponentideks võivad olla tavalised HTML märgendid või komponendid, mis ise koosnevad teistest komponentidest. Iga komponent kujutab endast Enyo objektihierarhiast päritud klassi, on unikaalselt identifitseeritav ning kättesaadav ülemkomponendi spetsiaalsest globaalse skoobiga nimeruumi matkivast muutujast. Viimane on eriti mugav seetõttu, et komponentide sisalduvusseos on puustruktuuriga ning komplekssemate komponentide puhul ei ole alamkomponentide poole pöördumiseks tarvis koostada pikki ahelavaldisi. Selline tehnika ei riku OOP kapseldamise põhimõtteid, sest komponentidest on kättesaadavad ainult alamkomponendid ning nende ja ka teiste komponentide klassimuutujate nähtavust saab muuta vastavate annotatsioonidega. Iga komponent hoolitseb, et alamkomponentide nimed oleksid unikaalsed. Nimekonfliktidest teavitatakse arendajat konsooli logitavate veateadete abil.

Suurendamiseks rakendusesisest taaskasutatavust, realiseeritakse keerulisemad komponendid enamasti eraldi, st rakenduse klassist väljaspool. See on võimalik, kuna alamkomponendid ei ole ülemkomponentidest sõltuvuses. Kõik raamistikus olevad komponendid on laiendatavad ning ei piirdu ainult kasutajaliideste vidinatega, vaid sisaldavad klasse nt JSONP päringutega töötamiseks või sündmuste töötlemiseks. Vidinate välimus on lihtsasti muudetav ja baseerub CSS stiililehtedel. Raamistiku koosseis on

põhilised veebirakendustes vajaminevad komponendid ja (osalt raamistiku ajaloo tõttu) spetsiaalselt puutekraanide jaoks kohandatud komponendid. Viimased on käesolevas töös vaadeldud raamistike hulgas kõige arenenumad.

Raamistiku paigaldamiseks piisab alla laadida kokkupakitud arenduspakett, mis sisaldab arendatava rakenduse alusmalli, s.o kataloogstruktuuri koos vastavate failidega, Enyo lähtekoodi ning Node.js platvormil baseeruvat käsureatööriista rakenduses oleva koodi minimeerimiseks ning selle kokku kombineerimiseks terviklikuks rakenduseks.

2.6.1 Õppimiskiirus

Testrakenduse valmimisele kulus orienteeruvalt kuus tundi. Enyo dokumentatsioon on hästi koostatud ning annab lühikese ent hea ülevaate raamistiku aluseks olevatest põhimõtetest. Komponentide kasutamist näitlikustav galerii (*showcase*) ning juhendites leiduvad koodinäited olid testrakenduse realiseerimisel oluliseks abiks. Enyo programmeerimisstiili on lihtne õppida ja mugav kasutada, sest komponentide koostamiseks piisab sageli paari olulisema parameetri täpsustamisest, instantside loomise eest hoolitseb raamistik ise. Kõik testrakenduse ehitamiseks vajaminevad komponendid olid raamistikus olemas.

Võrdlusmetoodika põhjal omistatakse raamistikule õppimiskiiruse eest üheksa punkti.

2.6.2 Sobivus erinevate veebilehitsejatega ja standarditele orienteeritus

Enyo toetab võrdlusmetoodikas välja toodud veebilehitsejatest kõiki peale Internet Explorer 7. versiooni, milles raamistiku abil realiseeritud rakenduste töökindlus ei ole garanteeritud. Brauserite toe eest omistatakse raamistikule neli punkti.

Raamistikus on toetatud CSS3 ja HTML5 kasutamine, mis annab punktisummase viis punkti. Selle võrdluskriteeriumi koondindeks on üheksa punkti.

2.6.3 Lokaliseerimine

Enyo koosseisus on lokaliseerimise otstarbeks eraldi pakett, milles on vahendid kuvakeelde tõlkimiseks ja tõlgete haldamiseks ning ka regioonispetsiifiliste formaatidega ümber käimiseks. Raamistikku on sisseehitatud 87 erinevale riigile (sh Eesti) omaste kuupäeva-, numbri- ja ajaühikute formaatide tugi.

Selle alampunkti koondindeks on 10 punkti.

2.6.4 Sisendi valideerimine

Enyo omavahendite hulgas ei ole sisendivalideerimisega varustatud komponente. Vastav funktsionaalsus tuleb arendajal ise programmeerida.

Võrdlusmetoodika põhjal on selles alampunktis omistatav punktisumma null punkti.

2.6.5 Integreeritud arenduskeskkonna (IDE) tugi

Raamistikule kogukonna poolt integreeritud arenduskeskkonda ega ka liidest mõnele kolmanda osapoolle poolt toodetud tarkvarale välja arendatud ei ole. Seega on selle alampunkti eest tulemus null punkti.

2.6.6 Kasutajatoe elemendid

Enyo dokumentatsioon koosneb õpetustest, juhendist ja klassideklaratsioonide kirjeldusest. Kõik eelloetletud on lihtsasti leitavad, loogiliselt ülesehitatud ning sisaldavad koodinäiteid. Kasutajad ei saa dokumentatsiooni ise muuta ega ka õpetusi lisada. Komponentide kasutamise näitlikustamiseks on arendajatele raamistiku veebilehel galerii (*showcase*).

Võrdlusmetoodikast lähtuvalt hinnatakse kasutajatoe elemente kaheksa punktiga.

2.6.7 Kogukonna aktiivsus

Enyo kasutamisega seonduvatele probleemidele saab abi selle jaoks mõeldud foorumitest, IRC kanalist või postiloendist. Raamistiku veebilehel hoitakse nimekirja firmadest, kes omavad raamistiku kasutamisel laialdaselt kogemusi ja seetõttu võiksid pakkuda tehnilist tuge või soovi korral rakenduse ise valmis ehitada. Võrdlusmetoodikast lähtuvalt annab nende kanalite olemasolu viis punkti.

GitHub⁵ andmetel tehti seisuga 01.04.2013 koodihoidlasesse eelneva 52 nädala jooksul keskmiselt 15 muudatust nädalas. Selle kriteeriumi eest omistatav punktisumma on viis punkti.

Kokku on alampunkti koondhindeks 10 punkti.

⁵ <https://github.com/enyojs/enyo/>

2.6.8 Testimisvahendite olemasolu

Raamistikuga ei ole kaasas testimisvahendeid, mistõttu selle alampunkti tulemuseks on null punkti.

2.6.9 Litsentsi tingimused

Enyo on litsentseeritud Apache litsentsi versioon 2.0 [21] alusel, mis lubab raamistiku abil realiseeritud tarkvara levitada ka ärilistel eesmärkidel, tingimusel, et arendaja säilitab koodifailides autoriõigusega seonduvad materjalid ja kaasab raamistiku litsentsi levitatava tarkvara koosseisu. Rakenduse lähtekoodi ega ka raamistiku modifitseeritud koodi avalikustamise kohustust sellega ei kaasne.

Võrdlusmetoodikast lähtuvalt on selle alampunkti tulemuseks 10 punkti.

2.6.10 Jõudlus

YSlow väljastab testrakenduse hindeks 100 punkti, võrdlusmetoodika alusel annab see koondpunktisummasse 10 punkti.

2.6.11 Hindamise tulemused

Eeltoodud alampunktides toodud punktide põhjal omistatakse raamistikule lõpptulemusena 66 punkti. Ülevaade koondpunktisumma kujunemisest on toodud Tabelis 6.

Tabel 6. Raamistiku Enyo koondpunktisumma kujunemine.

Õppimiskiirus	Veebilehitsejate tugi ja standarditele orienteeritus	Lokaliseerimine	Sisendi valideerimine	Integreeritud arenduskeskkond
9	9	10	0	0
Kasutajatoe elemendid	Kogukonna aktiivsus	Testimisvahendid	Litsentsi tingimused	Jõudlus
8	10	0	10	10
Kokku				66

3. Kokkuvõtte võrdlustulemustest

Eelnevas peatükis analüüsiti kümnest kriteeriumist lähtuvalt kuut erinevat avatud lähtekoodiga raamistikku. Analüüsi tulemused on esitatud koondpunktisumma vähenemise järjekorras Tabelis 7.

Tabel 7. Võrdlustulemused.

	Õppimiskiirus	Veebilehitsejad ja standarditele orienteeritus	Lokaliseerimine	Sisendi valideerimine	Integreeritud arenduskeskkond	Kasutajatoe elemendid	Kogukonna aktiivsus	Testimisvahendid	Litsentsi tingimused	Jõudlus	Kokku
Dojo	6	10	10	10	10	10	10	10	10	9,7	95,7
Qooxdoo	5	10	10	10	10	10	10	10	10	10	95,0
YUI	9	9	5	0	0	10	10	10	10	10	73,0
SproutCore	1	10	5	5	0	10	10	10	10	10	71,0
Enyo	9	9	10	0	0	8	10	0	10	10	66,0
Bootstrap	10	10	0	0	0	8	5	0	10	10	53,0

Võrdluse käigus realiseeriti iga raamistiku funktsionaalsust kasutades testrakendus. Kõigi kuue rakenduse valmimiseks kulus ühtekokku 50 tundi ehk orienteeruvalt 8,3 tundi rakenduse kohta. Raamistike dokumentatsioon oli enamasti hästi koostatud, sisaldas lisaks klassideklaratsioonide kirjeldusele ka koodinäidetega ilmestatud kasutusjuhendit ning õpetusi. SproutCore'i puhul oli dokumentatsioon küll hästi üles ehitatud, kuid paljude raamistiku kossesisis olevate klasside kasutamine oli piisava selgituseta, mistõttu selle raamistiku abil kulus testrakenduse realiseerimisele teistest oluliselt rohkem aega, koguni 14 tundi.

Raamistike võrdluses kogusid teistest enim punkte Dojo ja Qooxdoo, vastavalt 95,7 ja 95 punkti. Kuigi nende raamistike õppimisele ja testrakenduse valmimisele kulus keskmisest rohkem aega, on need võrreldavatest oma lisavõimaluste ning MVC mustri realiseerimiseks mõeldud vahendite osas keerulisemate rakenduste ehitamiseks kõige sobilikumad. Samuti olid need raamistikud ainsad, millele on kogukonna poolt välja arendatud integreeritud arenduskeskkonna tugi ning sisendivalideerimine koos veateadete kuvamisega.

Integreeritud arenduskeskkonna tugi puudus rohkem kui pooltel võrreldavatest raamistikest. Kuid, kuna raamistike aluseks olevad tehnoloogiad (HTML, CSS, JavaScript) on standardsed, saab raamistike abil töötamist lihtsustada ükskõik millise veebirakenduste ehitamist toatava arenduskeskkonna abil (Netbeans IDE, Adobe Dreamweaver jt). Raamistiku spetsiifilised elemendid (klassimallid, ennetav koodisestus klasside ja nimeruumide nimetuste osas jms) jäävad endiselt toeta ning selles osas tuleb arendajal tugineda dokumentatsioonile ning mälule.

Testrakenduse jõudluse mõõtmisel andsid kõik raamistikud väga häid tulemusi, vaid Dojo abil ehitatud testrakenduses esines Internet Explorer'i spetsiifilisi CSS avaldisi. Osades raamistikes oli lõpptoote (*production build*) jaoks olemas minimeeritud versioonid JavaScript ja CSS failidest, kuid enda kirjutatud lähtekood (moodulid, stiililehed) tuli minimeerida kasutades kolmandate osapoolte loodud vahendeid. Enyo, Qooxdoo ja SproutCore omasid selle protsessi automatiseerimiseks käsureapõhist tööriista. Võrdlusmetoodikas loetletud veebilehitsejad olid valdavas osas kõigi raamistike poolt toetatud nagu ka CSS3 ja HTML5 kasutamine.

Peaaegu kõik raamistikud toetasid kuvakeelte vahetamist ning regioonispetsiifiliste formaatidega komponente. Enyo toetas vaikimisi koguni 87 erineva riigi (sh Eesti) numbri-, kuupäeva- ja ajaühikuid. Kuvakeelte vahetamine oli enamasti realiseeritud vastavakeelsete tõlkefailide ja nende laadimise abil. Tõlgitud sõnedele vastavate võtmesõnade üle järje pidamine oli kõigis peale Qooxdoo jäetud arendaja hooleks.

Testimis- ja diagnostikavahendid olid olemas rohkem kui pooltel töös vaadeldud raamistikel ning seisnesid peamiselt logimisvõimalusi pakkuvates konsoolisarnastes komponentides, ühiktestides ja nende käitamiseks mõeldud brauseripõhistes testimiskeskondades. Qooxdoo puhul oli toetatud ka Selenium testide koostamine, mille jaoks olid raamistikus vastavad klassid.

Töös käsitletud raamistike litsentsid on valdavalt liberaalse iseloomuga ning lubavad raamistike põhjal arendatud tooteid levitada ka ärilistel eesmärkidel ilma kohustuseta rakenduse aluseks olevat lähtekoodi avalikustada. Enamik litsentse nõuavad autoriõigusega seonduvate materjalide säilitamist koodifailides, autoritele või kogukonnale viitamist, ning raamistikuga kaasaoleva litsentsi kaasamist levitatava tarkvara koosseisu.

Kõigis raamistikes oli tugi ka pihu- ja nutiseadmetele, seda kas kohanduva disaini põhimõtete järgimise või spetsiaalselt puutetundlike ekraanide jaoks mõeldud komponentide ja sündmuste töötlemise osas. Võrreldavatest võiks nutiseadmetele mõeldud rakenduste ehitamiseks olla kõige sobilikum Enyo, kuivõrd selles on hulganisti erinevaid puutetundlikele seadmetele mõeldud komponente ning programmeerimisliides puutetundlikele ekraanidele omaste sündmuste ja ekraaniviibete töötlemiseks.

Kõige vähem kogus punkte Bootstrap, mille suurimaks miinuseks on arendustöid kiirendavate võimaluste (integreeritud arenduskeskkond, sisendi valideerimine jms) puudumine. Samas on raamistik oma professionaalsete stiilide ja kergekaalulisuse tõttu hea ja kiire vahend mistahes suunitlusega veebirakenduste kasutajaliideste koostamiseks, kuid MVC mustriil baseeruva ning olekut säilitava ühelehelise rakenduse ehitamiseks mitte, sest selleks otstarbeks raamistikus vahendeid ei leidu. Kuid see ei tähenda, et raamistikus olevad komponendid ei võiks olla mõnes sellises, kuid teise raamistiku abil ehitatud, rakenduses kasutusel.

Kokkuvõte

Käesolevas bakalaureusetöös vaadeldi avatud lähtekoodiga veebipõhiseid kasutajaliideste raamistikke kui üht võimalikku vahendit veebirakenduste kasutajaliideste elik eesrakenduste realiseerimiseks ning nende kasutamisest tulenevaid eeliseid. Töös anti lühike ülevaade kuuest erinevast raamistikust (Bootstrap, Dojo, YUI, SproutCore, Qooxdoo, Enyo) ja nende võimalikest kasutusvaldkondadest ning analüüsiti neid vastavalt välja arendatud võrdlusmetoodikale.

Võrdlusmetoodika sisaldas kümmet kriteeriumi, mille alusel omistati igale võrdlusalusele raamistikule hinne skaalal 0 – 100. Võrdluskriteeriumite jõuasetus, pidades silmas alustavaid ettevõtteid ning idufirmasid, oli arendustöö kiirusel ning odavusel. Sellest johtuvalt vaadeldi kõigi kriteeriumite hulgas ka õppimiskiirust, integreeritud arenduskeskkondade olemasolu ning litsentsi tingimusi. Autori kogemusest lähtuvalt keskenduti üksnes kasutajapoolsel skriptimiskeelel JavaScript põhinevatel raamistikel.

Võrdluse tulemusena valmis raamistikest pingerida, milles kõrgeima tulemuse saavutas Dojo – seda tänu suurele hulga lisamoodulitele ning arendustööd kiirendavatele lisavõimalustele. Dojo muudab teiste seast väljapaistvamaks ka arvestatavate suurkorporatsioonide (Google, IBM jt) rahaline ja arendusalane panus raamistiku täiendamisse. Halvima tulemuse saavutas Bootstrap, mille tugevuseks on kergekaalulisus ning komponentide professionaalne välimus, kuid jäi võrdlusaluste raamistike hulgas oma väheste lisavõimaluste tõttu konkurentsist välja. Kõik raamistikud saavutasid hindena rohkem kui poole maksimaalsest tulemusest.

Saamaks ülevaadet raamistike pakutavatest võimalustest ning puudustest, realiseeriti iga raamistiku abil ühelehelise arhitektuuriga testrakendus, mille jõudluse hindamine oli ühtlasi üheks võrdlusmetoodika osaks. Testrakenduse sisuks oli otsingutulemuste kuvamine vastavalt kasutaja poolt ette antud otsingusõnale kasutades Google'i kohandatud otsingu jaoks mõeldud programmeerimisliidest. Kõigi raamistike põhjal realiseeritud testrakendused andsid jõudluse osas väga häid tulemusi.

Bakalaureusetöös läbi viidud analüüsi tulemusi sobib kasutada võrdlusaluste raamistike kohta info kogumisel või nende hulgast sobivaima välja valmimiseks. Välja arendatud võrdlusmetoodika võiks olla rakendatav ka teiste sh tasuliste raamistike kõrvutamiseks ning analüüsimiseks.

A Comparison of Open Source Web UI-Frameworks

Bachelor's Thesis (6 EAP)

Mati Kärner

Summary

Over the last twenty years The World Wide Web has evolved into a stable software application platform. A powerful distribution model and emerging technologies such as WebGL and HTML5, that bridge caps between web and desktop applications, have led to a growing number of developers favoring web as a platform in developing and deploying software. The objective of this thesis was to give a short overview of the benefits of using open-source UI-frameworks in web application development, and to compare six JavaScript based frameworks (Bootstrap, Dojo, YUI, SproutCore, Qooxdoo, Enyo) according to developed methodology with purpose to offer guidance in choosing amongst them.

The comparison methodology was put together of ten criteria (learning speed, browser compatibility and support for HTML5 and CSS3, input validation tools, support for integrated development environments, community viability, testing tools, license terms and performance), each graded on the scale of 1 – 10. The sum of these grades constituted as an overall score (range 0 – 100 points) for each evaluated framework. In order to gain deeper insight into frameworks' advantages and disadvantages a sample application, that also served the purpose of measuring learning speed and performance, was built with every framework.

The comparison concluded with Dojo ranking as first (total of 95.7 points), mainly due to additional development easing tools such as support for integrated development environment and input validation. The second tool of choice would be Qooxdoo (total of 95 points) – a similar to former, but takes longer to acquire. In comparison with Dojo, Qooxdoo had several advantages, such as built-in support for Selenium tests and automated tools for generating translation files. The lowest scoring (total of 53 point) framework was Bootstrap, although lightweight and equipped with professional looking UI-components, lacks essential features such as support for localization and input validation.

Kasutatud kirjandus

- [1] Taivalsaari, A., Mikkonen, T. The Web as an Application Platform: The Saga Continues. - *37th EUROMICRO Conference on Software Engineering and Advanced Applications, Oulu, Finland, August 30 - September 2, 2011*, 170-174. [Online] IEEE Xplore (12.03.2013)
- [2] Howell J. C., Kapfhammer M. G., Roos S. R. An examination of the run-time performance of GUI creation frameworks. - *Proceedings of the 2nd international conference on Principles and practice of programming in Java (PPPJ '03)*. Computer Science Press, Inc., New York, NY, USA, 2003, 171-176. [Online] ACM Digital Library (12.03.2013)
- [3] Björemo M., Trninić P. Evaluation of web application frameworks : Master of Science Thesis in Software Engineering and Technology. Göteborg, Sweden, Chalmers University of Technology, 2010. [Online] Chalmers Publication Library (14.03.2013)
- [4] Pietschmann, S., Voigt, M., Meissner, K. Dynamic Composition of Service-Oriented Web User Interfaces. - *Fourth International Conference on Internet and Web Applications and Services, Venice, Italy, May 24-28, 2009*, 217-222. [Online] IEEE Xplore (16.03.2013)
- [5] Gartner Says Worldwide Software-as-a-Service Revenue to Reach \$14.5 Billion in 2012. (2012) [WWW] <http://www.gartner.com/newsroom/id/1963815> (18.03.2013)
- [6] HTML5 web app architecture explained. [WWW] <http://blog.sc5.fi/2012/01/html5-web-app-architecture-explained/> (20.03.2013)
- [7] 75% of Developers Using HTML5: Survey. [WWW] <http://www.eweek.com/c/a/Application-Development/75-of-Developers-Using-HTML5-Survey-508096/> (20.03.2013)
- [8] Smith P. Professional Website Performance: Optimizing the Front End and the Back End. Indianapolis : John Wiley & Sons, Inc., 2013.
- [9] Bootstrap. [WWW] <http://twitter.github.io/bootstrap/> (20.03.2013)
- [10] Unbeatable JavaScript Tools. [WWW] <http://dojotoolkit.org/> (20.03.2013)
- [11] YUI Library. [WWW] <http://yuilib.com/> (20.03.2013)
- [12] SproutCore. [WWW] <http://sproutcore.com/> (20.03.2013)
- [13] A Universal JavaScript Framework. [WWW] <http://qooxdoo.org/> (20.03.2013)

- [14] Enyo JavaScript Application Framework. [WWW] <http://enyojs.com/> (20.03.2013)
- [15] Top 12 Browser Versions on 2012.
[WWW] http://gs.statcounter.com/#browser_version_partially_combined-ww-yearly-2012-2012-bar (22.03.2013)
- [16] How are the grades computed? [WWW] http://yslow.org/faq/#faq_grading
(22.03.2013)
- [17] Custom Search. [WWW] <https://developers.google.com/custom-search/v1/overview> (01.04.2013)
- [18] Bootstrap from Twitter. [WWW] <https://dev.twitter.com/blog/bootstrap-twitter>
(09.04.2013)
- [19] Cochran D. Twitter Bootstrap Web Development How-To. Birmingham : Packt Publishing Ltd., 2012.
- [20] Browser Compatibility. [WWW]
<https://github.com/twitter/bootstrap/wiki/Browser-Compatibility> (09.04.2013)
- [21] Apache License, Version 2.0. [WWW] <http://www.apache.org/licenses/LICENSE-2.0.html> (09.04.2013)
- [22] A Brief History of Dojo. [WWW]
<http://dojotoolkit.org/referenceguide/1.8/quickstart/introduction/history.html>
(09.04.2013)
- [23] The Dojo Foundation. [WWW] <http://dojofoundation.org/> (09.04.2013)
- [24] AMD. [WWW] <https://github.com/amdjs/amdjs-api/wiki/AMD> (09.04.2013)
- [25] Academic Free License v2.1. [WWW] <http://spdx.org/licenses/AFL-2.1>
(09.04.2013)
- [26] Miraglia E. The Yahoo! User Interface Library. [WWW]
<http://yuiblog.com/blog/2006/02/13/the-yahoo-user-interface-library/> (15.04.2013)
- [27] Software Licence Agreement. (BSD Licence)
[WWW] <http://yuilib.com/license/> (15.04.2013)
- [28] Ocean E. SproutCore 1.0 History. [WWW]
<http://wiki.sproutcore.com/w/page/12412998/SproutCore10-History> (16.04.2013)
- [29] McLean P. Apple's open secret: SproutCore is Cocoa for the Web. [WWW]
http://appleinsider.com/articles/08/06/16/apples_open_secret_sproutcore_is_cocoa_for_the_web (04.04.2013)
- [30] The MIT Licence (MIT). [WWW] <http://opensource.org/licenses/MIT> (04.04.2013)

- [31] Developers. [WWW] <http://qooxdoo.org/project/developers> (22.04.2013)
- [32] GNU Lesser General Public License v2.1. [WWW]
<http://www.gnu.org/licenses/lgpl-2.1.html> (22.04.2013)
- [33] About Enyo. [WWW] <http://www.enyojs.com/about/> (22.04.2013)

Lisad

Lisa 1

Näide päringust kasutades Google'i kohandatud otsingu jaoks mõeldud programmeerimisliidest.

Päring:

GET

```
https://www.googleapis.com/customsearch/v1?key=APIKEY&cx=002647590579739994937:ntvk68za0vo&q=otsing&num=10&start=1
```

Vastus:

```
{
  "kind": "customsearch#search",
  "url": {
    "type": "application/json",
    "template":
      "https://www.googleapis.com/customsearch/v1?q={searchTerms}&num={count?}&start={startIndex?}&lr={language?}&safe={safe?}&cx={cx?}&cref={cref?}&sort={sort?}&filter={filter?}&gl={gl?}&cr={cr?}&googlehost={googleHost?}&c2c=off={disableCnTwTranslation?}&hq={hq?}&hl={hl?}&siteSearch={siteSearch?}&siteSearchFilter={siteSearchFilter?}&exactTerms={exactTerms?}&excludeTerms={excludeTerms?}&linkSite={linkSite?}&orTerms={orTerms?}&relatedSite={relatedSite?}&dateRestrict={dateRestrict?}&lowRange={lowRange?}&highRange={highRange?}&searchType={searchType}&fileType={fileType?}&rights={rights?}&imgSize={imgSize?}&imgType={imgType?}&imgColorType={imgColorType?}&imgDominantColor={imgDominantColor?}&alt=json"
  },
  "queries": {
    "nextPage": [
      {
        "title": "Google Custom Search - otsing",
        "totalResults": "702000",
        "searchTerms": "otsing",
        "count": 10,
        "startIndex": 11,
        "inputEncoding": "utf8",
        "outputEncoding": "utf8",
        "safe": "off",
        "cx": "002647590579739994937:ntvk68za0vo"
      }
    ]
  },
  "request": [
    {
      "title": "Google Custom Search - otsing",
      "totalResults": "702000",
      "searchTerms": "otsing",
      "count": 10,
      "startIndex": 1,
      "inputEncoding": "utf8",
      "outputEncoding": "utf8",
      "safe": "off",
    }
  ]
}
```

```

    "cx": "002647590579739994937:ntvk68za0vo"
  }
]
},
"context": {
  "title": "Google"
},
"searchInformation": {
  "searchTime": 0.119947,
  "formattedSearchTime": "0.12",
  "totalResults": "702000",
  "formattedTotalResults": "702,000"
},
"items": [
  {
    "kind": "customsearch#result",
    "title": "Delfi Search",
    "htmlTitle": "Delfi Search",
    "link": "http://otsing.delfi.ee/",
    "displayLink": "otsing.delfi.ee",
    "snippet": "Find relevant information on Estonian web pages. Also has an interactive map of the country. [Estonian, English, Russian]",
    "htmlSnippet": "Find relevant information on Estonian web pages. Also has an interactive map of \u003cbr\u003e the country. [Estonian, English, Russian]",
    "formattedUrl": "otsing.delfi.ee/",
    "htmlFormattedUrl": "\u003cb\u003eotsing\u003c/b\u003e.delfi.ee/"
  },
  ...
]
}

```

Lisa 2

Kasutatud lühendid ja nende vasted.

ADM	<i>The Asynchronous Module Definition</i>
AJAX	<i>Asynchronous JavaScript And XML</i>
API	<i>Application Programming Interface</i>
ARIA	<i>Accessible Rich Internet Applications</i>
CDN	<i>Content Delivery Network</i>
CERN	European Council for Nuclear Research
DOM	<i>Document Object Model</i>
GNU LGPL	GNU Lesser General Public License

IDE	<i>Integrated Development Environment</i>
IRC	<i>Internet Relay Chat</i>
JSON	<i>JavaScript Object Notation</i>
JSONP	<i>JSON with Padding</i>
MIT	Massachusetts Institute of Technology
MVC	<i>Model–View–Controller</i>
REST	<i>REpresentational State Transfer</i>
RIA	<i>Rich Intrenet Application</i>
SCSS	Sassy CSS
TCP	<i>Transmission Control Protocol</i>
URL	<i>Uniform Resource Locator</i>
UX	<i>User eXperience</i>
WYSIWYG	<i>What-You-See-Is-What-You-Get</i>

Lisa 3

Raamistike abil realiseeritud testrakenduste lähtekood on salvestatud tööga kaasasolevale DVD-le.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Mati Kärner (sünnikuupäev: 18.07.1989)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Avatud lähtekoodiga veebipõhiste kasutajaliideste raamistike võrdlus“, mille juhendaja on Helle Hein,

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus **13.05.2013**