

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Marten Truu 142772IAPB

**TÄISÜHENDUSPARTNERVÕRGUGA
VEEBIRAKENDUS
CARDS AGAINST HUMANITY
LAUAMÄNGU BAASIL**

Bakalaureusetöö

Juhendaja: Roger Kerse

MSc
Lektor

Tallinn 2017

Autorideklaratsioon

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

Autor: Marten Truu

22.05.17

Annotatsioon

Lõputöö eesmärgiks on analüüsida olemasolevaid digitaalsed Cards Against Humanity lauamängul baseeruvaid lauamänge nende kasutajate ja ruumide halduse ning mänguprotsessi põhjal. Analüüsi alusel luuakse nõuded Cards Against Humanity-l baseeruva rakenduse loomiseks, mida kasutatakse neile nõuetele vastava võrgurakenduse arenduses, kus võimalikult suur osa kommunikatsioonist tehakse kasutades partnerühendusi.

Töö käigus loodi Cards Against Humanity-l põhineva mängu veebirakendus. Kogu mängusisene kasutajate vaheline suhtlus rakenduses tehakse kasutades loodud täisühenduspartnervõrku. Partnervõrgu kasutamine võimaldab otsesuhtlust rakenduse kasutajate vahel ning vabastab serveriressursid, mida muidu kasutataks klientide vaheliste sõnumite edastamiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 5 peatükki, 11 joonist.

Abstract

A FULL MESH PEER NETWORK IN A WEB APPLICATION BASED ON CARDS AGAINST HUMANITY

Many modern web browsers support WebRTC, which is a technology that allows the browsers to communicate with each other in real time and directly. This removal of third parties from communication allows for more effective messaging between clients and to free up resources for servers that would otherwise be used to accommodate client to client messaging.

The aim of this thesis is to analyse existing digital Cards Against Humanity party games based on how these implementations handle user authentication, game room management and the game process. Based on this analysis, requirements for a digital Cards Against Humanity clone are created, which will be used in the development of a web application that conforms to the requirements and where as much communication as possible is done peer to peer.

As result of this thesis a web application was created, which implements a Cards Against Humanity style game on a peer-to-peer full mesh network used for all in-game data communication between clients.

The thesis provides an overview of the technologies used to create the application as well as how the technologies were used in the making of the application. The thesis also explains how the implemented peer-to-peer communication rooms are initialized and how data is exchanged between connected clients.

Compared to WebSockets, a much lower percentage of browsers support WebRTC. Due to this, another data channel should be provided in applications using the technology. WebRTC based peer network solutions are much more complex and time consuming to create than solutions with a centralized server. Despite the browser support and complexity issues, using a peer network allows for direct real time communication between clients without taxing the application server.

The thesis is in Estonian and contains 28 pages of text, 5 chapters, 11 figures.

Lühendite ja mõistete sõnastik

WebRTC	Kommunikatsiooni protokollid mis võimaldavad reaaliajakommunikatsiooni veebis [1].
Partnerühendus	<i>Peer-to-peer connection.</i> Partnerilt partnerile ühendus.
Täisühendusvõrk	<i>Full mesh network.</i> Võrk kus igal seadmel on ühendus iga teise võrgus oleva seadmega.
WebKit	Veebibrauseri mootor, mida kasutatakse Safari, App Store ja paljude muude macOS-i, iOS-i ja Linuxi rakenduste poolt [2].
HTTPS	Turvaline versioon hüpertextiedastusprotokollist, mille puhul kogu brauseri ja veebilehe vaheline suhtlus on krüpteeritud [3].
API	<i>Application programming interface.</i> Rakendusliides, protokollid ja meetodid mille põhjal üks rakendus kasutab teise rakenduse teenuseid.
WebSocket	Tehnoloogia, mis võimaldab interaktiivset brauseri ja serveri vahelist kommunikatsioonisessiooni [4].
JavaScript	Dünaamiline, erinevaid programmeerimise stiile toetav interpreteeritav või sünkroonkpileeritav programmeerimiskeel [5].
ICE	<i>Interactive Connectivity Establishment.</i> Interaktiivse ühenduse loomine. Kasutatakse partnervõrgunduses, et leida võimalikult otsene viis kahel arvutil omavahel suhelda.
<i>Frontend</i>	Otseselt kasutajale nähtav ja tema poolt mõjutatav osa rakendusest.
<i>Backend</i>	Serveripoolne osa rakendusest, millega kasutajapoolne osa rakendusest suhtleb.
URL	<i>Uniform Resource Locator.</i> Viide võrguressursile, mis kirjeldab selle asukohta arvutivõrgus.
NAT	<i>Network Address Translation.</i> Võrguaadresside transleerimine. Võrguaadresside kaardistamine ühest aadressiruumist teise. Kasutatakse ühe aadressiga terve arvutite grupi kirjeldamiseks [6].

<i>Reverse proxy server</i>	Vahendav server, mis suunab klientide päringuid soovitud serverisse.
PID	<i>Process identifier.</i> Unikaalseks protsesside identifitseerimiseks kasutatav number.
<i>hangup</i>	Protsessile terminali sulgemisel saadetav signaal.

Sisukord

1 Sissejuhatus.....	11
1.1 Ülesande püstitus	12
2 Analüüs	13
2.1 Olemasolevad lahendused.....	13
2.1.1 Pretend You're Xyzzy	14
2.1.2 Cards for Humanity	15
2.1.3 World's Filthiest CAH Online (azala.info).....	15
2.2 Rakenduse nõuded	16
2.2.1 Funktsionaalsed nõuded kasutajate haldusele.....	16
2.2.2 Funktsionaalsed nõuded ruumidele	16
2.2.3 Funktsionaalsed nõuded mängule.....	17
2.2.4 Mittefunktsionaalsed nõuded kogu rakendusele.....	17
3 Rakenduse disain	18
3.1 Süsteemi arhitektuur	18
3.1.1 Serverirakenduse tööshoidmine.....	19
3.2 Kasutajapoolses rakenduses kasutatavad tehnoloogiad ja teegid	20
3.2.1 React	21
3.2.2 Redux.....	21
3.3 Serveripoolses rakenduses kasutatavad tehnoloogiad ja teegid.....	22
3.4 Kommunikatsiooni-tehnoloogiad	22
3.4.1 WebRTC	22
3.5 Võrgutopoloogia	24
4 Veebirakendus.....	27
4.1 Kasutajad.....	29
4.2 Ruumid.....	30
4.2.1 Ruumiga liitumine	32
4.2.2 Ruumist lahkumine	33
4.3 Mäng	34
4.3.1 Vajalik kaartide arv mängus	37

5 Kokkuvõte.....	39
------------------	----

Jooniste loetelu

Joonis 1. Süsteemi arhitektuur	19
Joonis 2. RTC Partnerühenduse loomine	24
Joonis 3. Võrgutopoloogia	25
Joonis 4. Võrgutopoloogia WebRTC toeta brauseri puhul	26
Joonis 5. Rakenduse mobiilivaated	28
Joonis 6. Sisselogimise vaade	30
Joonis 7. Ruumide vaade	32
Joonis 8. Ruumiga liitumise tegevusdiagramm	33
Joonis 9. Tavamängija vaade mängus	35
Joonis 10. Hindajast mängija vaade mängus	36
Joonis 11. Vaade peale võitja selgumist mängus	37

1 Sissejuhatus

Cards Against Humanity [7] on lauamäng, kus mängijad täidavad etteantud sõnade ja lausetega lüngad lausetes või vastavad nende abil küsimustele. Mängijatele esitatavad lünkadega laused ja küsimused asuvad mustadel kaartidel ja vastusevariandid valgetel kaartidel. Laused kaartidel julgustavad mängijaid piinlike ja tabu teemade üle nalja viskama [8]. Antud lauamäng on kättesaadaval Creative Commons BY-NC-SA 2.0 [9] litsentsi alusel. See tähendab, et Cards Against Humanity sisu võib kasutada milleks iganes, kuid seda kasutades tuleb viidata mängu loojatele, sisu kasutamise läbi ei tohi teenida kasumit ja mängu põhjal loodud asju peab jagama samal viisil mil mängu looja seda teeb [7]. Tulu teenimise piirangu tõttu on sellel lauamängul põhinevatel olemasolevatel veebirakendustel piirangud kasutajate arvule ja probleeme serverite ülalhoidmise kuludega.

WebRTC [10] partnerühenduste loomist toetavad 76.94% Eestis kasutatavatest brauseritest ja 65.19% globaalselt kasutuses olevatest brauseritest [11]. See tehnoloogia võimaldab kasutajate brauseritel üksteisega ilma vahendajateta reaalaja andmesidet pidada. Kolmandate osapoolte eemaldamine kasutajate vahelisest sidepidamisest teeb kommunikatsiooni efektiivsemaks ja hoiab ära serveripoolsete ressursside kulutamise, mida muidu kasutataks sõnumite vahendamiseks ja ühenduste hoidmiseks suhtluse osapooltega.

Töö tulemusena valmib Cards Against Humanity lauamängul baseeruv veebirakendus, mis kasutades WebRTC partnerühendusi võimaldab luua suvalisel ajal liitumist ja lahkumist lubava täisühendusvõrgu (*full mesh network*) üksteisega ühenduses olla soovivate kasutajate vahel.

Teema valik tuleneb autori huvist Cards Against Humanity lauamängule digitaalne versioon luua, antud mängus vajatavate andmesidesõnumite sobivusest partnervõrgus edastamiseks ja võimalusest luua täisühendus partnervõrk veebis.

1.1 Ülesande püstitus

Bakalaureusetöö eesmärk on analüüsida olemasolevaid digitaalseid Cards Against Humanity baasil loodud mängu kasutajate ja ruumide halduse ning mänguprotsessi põhjal. Töö käigus loodavad andmesidevõrgu, kasutajatehaldus-süsteemi ja mängu protsessi lahendused peavad vastama analüüsi käigus leitud nõuetele.

Töö teine eesmärk on luua võimalikult palju andmesidet üle partnerühenduste saatev Cards Against Humanity-l baseeruv veebirakendus, kirjeldada ülesande lahendamiseks valitud tehnoloogiaid, rakenduse keskkonna ülesehitust ja rakendust ennast.

2 Analüüs

Analüüsi käigus leitab autor, mis on Cards Against Humanity-l baseeruva võrgurakenduse suhtluskanali ja implementatsiooni juures oluline. Olemasolevate lahenduste kasutajate haldamise, mänguruumide loomise ja kasutajatele pakutavate mängu kohandamisvõimaluste implementatsioonide uurimise põhjal kirjeldab autor funktsionaalsed ja mittefunktsionaalsed nõuded. Funktsionaalsed nõuded esitatakse nõuetena kasutaja haldusele, ruumidele, mängule ja rakendusele. Kirjeldatud mittefunktsionaalsed nõuded kehtivad kogu rakendusele.

2.1 Olemasolevad lahendused

Lõputöö käigus valmivale rakendusele sarnaseid lahendusi otsiti Google [12], iOS App Store [13], Google Play Store [14] ja Cards Against Humanity reddit'i kommuuni leheküljelt [15]. Analüüsiks valis autor brauseris töötavad rakendused kuna võrgurakendustel on potentsiaal töötada kõigil platvormidel, mis toetavad modernseid veebibrauserite tehnoloogiaid, võrgurakendused on kasutajatele kiiresti kättesaadavad ning ei nõua rakenduse alla laadimist ja installeerimist kasutaja poolt. Võrgurakendustel on ka eelis arendajale, sest ei ole vaja luua erinevate platvormide jaoks erinevaid rakendusi.

Cards Against Humanity-l baseeruvatel võrgurakendustel on probleeme serverite ülalhoiukuludega. Cards Against Originality [16] mängu mängiti 11 kuu jooksul kolme miljoni inimese poolt kokku 47 aastat, kuid selle rakenduse autor ei suutnud enam rakenduse serverimise kuludega toime tulla [16]. Kahel uuritud rakendusel kolmest on praegu või minevikus olnud probleeme serverite ülalpidamise kuludega. Kõik rakendused võimaldavad kasutajatel luua kasutajakonto eesmärgiga kasutajate loodavat informatsiooni kasutajate endi ja teiste kasutajate poolt hiljem rakenduses kasutada. Kasutajakontode loomine on uuritud rakendustes tehtav külaliskonto puhul ühe või kahe nupuvajutusega. Kahes rakenduses oli võimalik näha ruumide nimekirja, spetsiifiliselt ruume luua, nendega liituda ja neid parooliga kaitsta. Rakenduses, kus ruumide nimekirja ei olnud, oli ruumiga liitumine automaatne või veebilehe aadressi alusel. Kõikide rakenduste ruumidega oli võimalik liituda kasutades ruumispetsiifilist veebilehe aadressi. Kahes rakenduses oli ruumidega lubatud liituda kuni mängu

alustamiseni, World's Filthiest CAH Online lubas ka liituda mängu kestel. Mängu kuvamisega erinevatel ekraanisuurustel sai hästi hakkama üks rakendus.

Alljärgnevalt toob autor välja erinevad olemasolevate lahenduste kirjeldused, keskendudes erinevatele sisselogimise ja ruumide halduse lahendustele ning minimaalselt ka mängusisestele protsessidele rakendustes.

2.1.1 Pretend You're Xyzzzy

Pretend You're Xyzzzy on arendusjärgus olev Cards Against Humanity veebikloon [17]. Veebileht on väga minimaalselt stiliseeritud ning sisaldab väga palju informatiivset teksti ja üksteise peal asetsevaid elemente.

Mängu kasutajaks registreerimine ja sellega kaasnev sisselogimine on lihtsad, sisestada tuleb vaid soovitud hüüdnimi, ning seejärel on võimalik autoriseeritult jätkata ruumide vaates.

Ruumide vaatesse jõudmiseks peab kasutaja läbima kaks lehte, kus on kirjas muudatuste logi, teadaolevad vead mängus, mängusisesed võimalused, piirangud ja tulevikuplaanid. Ruumide vaates on võimalik mängu luua, otsida, vaadelda ja nendega liituda. Lisaks näidatakse kas ruumiga seonduv mäng on alustatud või mitte. Ruume on võimalik kõikide ruumiga seonduvate sõnade põhjal filtreerida, et leida sobiv ruum vastavalt otsingusõnale. Iga ruumiga seonduv kaks nuppu. Üks mänguruumiga liitumiseks ja teine vaatlemiseks. Kui mänguruumiga liitumise nuppu vajutada mängus, mis juba on alustatud seisundis ignoreeritakse nupuvajutust ning puudub tagasiside ignoreerimise põhjuste kohta.

Mängu vaates on võimalik sätestada maksimaalse punktiseisu, mängijate ja vaatlejate arvude limiite ning maksimaalset ooteaega mängijate tegevusele. Valitud seadeid näevad kõik ruumiga liitunud mängijad. Mänguruumile saab määrata ka salasõna, mis tähendab et ruumide loojad saavad lubada vaid salasõnale juurdepääsu omavatel kasutajatel mänguruumidega liituda. Kui ruumist lahkub ruumi looja saab vastavad õigused endale üks mänguruumis olevatest kasutajatest. Mängu ei ole võimalik alustada ilma minimaalselt kolme mängijata.

Käimasoleva mängu puhul näidatakse kasutaja poolt hindamiseks edastatud kaardid koheselt välja. Kaartide edastamine peaks olema kuni hindaja poolt võitva

kombinatsiooni valimiseni anonüümne, kuid kohese kaartide väljanäitamise tõttu on samas ruumis mängivate inimeste korral näha kes valimise lõpetas ja kaardid edastas.

Lehele on olnud vaja teha mitmeid muudatusi, et Amazon Web Services kasutamise makseid kontrolli alla saada [17]. Tundub olevat siiani suhteliselt ressursimahukas, kuna rakendusel on kasutusel 3 serverit piirangutega 2000 kasutajat ja 400 mängu serveri kohta.

2.1.2 Cards for Humanity

Cards for Humanity [18] on heategevuslik Cards Against Humanity veebiversioon, mis võimaldab annetada lastele, samas lubades kasutajatel jääda põlastusväärseks ja piinlikult imelikuks (*awkward*), nagu nad seda oma naturaalses olekus on [18]. Esilehel näidatakse kasutajale elementaarseid mängureegleid ja pakutakse võimalust end registreerida, külastajana mängida või sisse logida. Veebileht saab hakkama ekraanisuuruse muutustele ning on kasutatav väiksema ekraaniga nutiseadmetes.

Mänguruumid on lahendatud eelmisest rakendusest väga erinevalt. Ruumide valiku vaade puudub täielikult. Külalisena sisse logides on võimalik mängida vaid suvaliste kasutajatega. Külalisena sisselogimise nupule vajutamine tähendab, et automaatselt luuakse uus ruum või liitutakse juba olemasoleva ruumiga kus on teised suvalised kasutajad. Registreerunud kasutajad saavad luua ise uusi ruume, millega saab liituda minnes brauseris rakenduse poolt genereeritud aadressile või mängida koos külaliskasutajatega. Kui saab kokku üle kolme mängida sooviva kasutaja on mänguruumiga esimesena liitunud või ruumi loomise algatanud kasutajal õigus mängu alustada. Maksimaalne inimeste arv ühes mängus on kuus ning selle arvuni jõudmisel algab mäng automaatselt.

Kaartide valimiseks ja hindamiseks mängu kestel on aega 20 sekundit, ning seda muuta võimalik ei ole. Kaarte valides näidatakse need sarnaselt eelmise rakendusega koheselt välja. Mängust on lubatud suvalisel hetkel lahkuda kuid mängu kestel liituda ei ole võimalik.

2.1.3 World's Filthiest CAH Online (azala.info)

World's Filthiest CAH Online [19] on veebileht, mis võimaldab internetis kasutajatel Cards Against Humanity lauamängu digitaalsel kujul mängida [19]. Mängule lisati

hiljuti (1.05.2017) ka mobiilivaate nupp, mis võimaldab muuta mängu ilma muid veebilehe elemente paigast ära ajamata kasutatavaks väikestel ekraanidel [20].

Azala.info server suudab toetada maksimaalselt 30 paralleelselt eksisteerivat mänguruumi. Mängus on võimalik luua salasõnaga kaitstud mänguruumi ning mänguruumidega on võimalik liituda kasutades genereeritud aadressi. Antud rakendus lubab kasutajatel mängudega liituda ka mängu ajal, lülitada ajalimiidid välja või neid vastavalt soovile muuta.

2.2 Rakenduse nõuded

Nõuded loodavale rakendusele tulenevad uuritud rakenduste kasutajate- ja ruumihalduse ning mängumehaanika uurimise käigus leitud positiivsetest ja negatiivsetest omadustest. Võrgupartneritel põhinev andmesidekanal peab olema võimeline pakkuma sama funktsionaalsust mida olemasolevad tsentraalse serveriga lahendused pakuvad.

Funktsionaalsed nõuded kirjeldavad mida rakendus tegema peaks, ehk kirjeldavad süsteemi käitumist sõltuvalt sellest kuidas nad süsteemi funktsionaalsusega seotud on. Mitte-funktsionaalsed nõuded seavad piiranguid sellele, kuidas rakendus teeb seda mida ta tegema peaks. Mitte-funktsionaalne nõue on kriteerium, mille alusel on võimalik süsteemi opereerimist hinnata.

2.2.1 Funktsionaalsed nõuded kasutajate haldusele

1. Kasutajal on võimalik luua kasutajakonto.
2. Kasutajal on võimalik luua külalis-kasutajakonto kiireks esmaseks sisselogimiseks.
3. Kasutajakonto on võimalik meelde jätta.
4. Meelde jäetud kasutajakonto korral on sisselogimise protsess automaatne.

2.2.2 Funktsionaalsed nõuded ruumidele

1. Ruumid peavad võimaldama kasutajatel nendega liituda suvalisel hetkel kui mäng ei ole ruumis alustatud.
2. Ruumidega peab olema võimalik uuesti liituda kui mängija mängu käigus mingil põhjusel ruumist lahkub.
3. Ruume peab olema võimalik teha privaatseks.

2.2.3 Funktsionaalsed nõuded mängule

1. Mängu alustamisel valitakse suvaliselt esimene hindaja (kaardi tsaar).
2. Mänguringi alustamisel esitatakse mängijatele üks must kaart.
3. Mänguringi alustamisel on igal mängijal 10 valget kaarti.
4. Mängijate poolt saadetud vastused näidatakse välja alles peale seda kui viimane mängija on valitud kaardid edastanud.
5. Hindajal on võimalik valida üks kaardikombinatsioon mängijate vastuste seast.
6. Peale hindamist näidatakse välja võitnud kaardikombinatsioon.
7. Peale hindamist alustatakse uut mänguringi.
8. Uue mänguringi alustamisel valitakse ettemääratud ringlusviisi alusel järgmine hindaja.

2.2.4 Mittefunktsionaalsed nõuded kogu rakendusele

1. Rakenduse kasutajaliides peab olema inglise keeles.
2. Rakenduse kasutajaliides peab olema kasutatav põhiliselt portree režiimis kasutatavatel mobiilsete seadmete ekraanidel.
3. Protsessid ja tegevused rakenduses peavad olema tagasisidestatud.
4. Rakenduse kasutajaliides ei tohi vajada eraldi pikka tekstilist seletamist.
5. Rakendus ei tohi välja näidata elemente mida antud rakenduse olekus kasutajal kasutada võimalik ei ole.
6. Rakendus tuleb üles ehitada nii, et ei oleks väga keeruline lisada kasutajaliidesesse uusi keeli (eesti keel).
7. Võimalikult palju kasutajate vahelist suhtlust peab toimuma rakenduse kasutajate vahel ilma rakenduse veebiserverit kasutamata.
8. Server peab olema võimeline kliente teavitama reaalajas kui neile oluline informatsioon teistelt klientidelt serverisse jõuab.
9. Süsteemi kasutamisel ei tohi samas ruumis viibivatel kasutajatel olla märgatavaid viivitusi informatsiooni kuvamisel.
10. Juhul kui ühel kasutajal mingil põhjusel ühendus teiste kasutajatega katkeb, ei tohi see segada teistel kasutajatel rakenduse kasutamist.
11. Kasutajal peab olema võimalus mingil põhjusel lahktud mänguga uuesti liituda.

3 Rakenduse disain

Selles peatükis tutvustab autor süsteemi arhitektuuri, rakenduse poolt kasutatavaid tehnoloogiaid, teenuseid ja raamistikke ning kirjeldab kuidas rakenduses neid kasutatakse.

3.1 Süsteemi arhitektuur

Süsteem on üles ehitatud sellisel moel, et võimalikult palju informatsiooni vahetaksid süsteemi kasutajad omavahel. Rakendus haldab kasutajakontosid ning pakub kasutajatele võimalust omavahel tekstipõhiselt suhelda, seetõttu suhtleb kliendi brauser serveriga HTTPS päringute ja turvaliste WebSocket ühenduste abil ning kliendid omavahel partnerühenduste kaudu, milles on kommunikatsioon krüpteeritud.

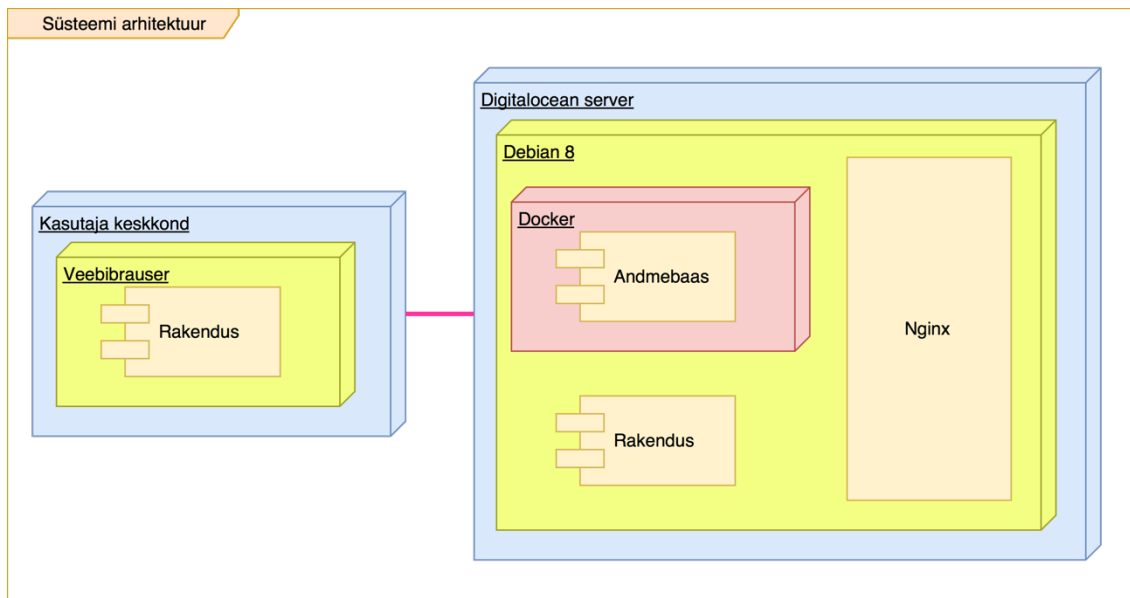
Rakenduse server asub DigitalOceanis [21]. DigitalOcean on teenus, mis pakub virtuaalmasinaid, millele on võimalik peale panna üks kuuest valikus olevast operatsioonisüsteemist või suurem valik juba mõne installeeritud tehnoloogiaga Ubuntu [22] versioon. Kuna autor on tuttav Debian-il [23] baseeruvate Linuxi distributsioonidega valis autor serveri operatsioonisüsteemiks Debian 8 [24]. Tänu täielikule kontrollile operatsioonisüsteemi üle saab serveris luua täpselt vajadustele ja soovidele vastava keskkonna rakenduse jaoks.

Virtuaalmasina operatsioonisüsteemil jookseb omakorda Docker-i [25] konteiner kuhu on installeeritud Postgres [26] andmebaas, millega rakenduse *backend* suhtleb. Andmebaas on paigutatud Docker konteinerisse kuna see garanteerib, et andmebaas töötab erinevatel operatsioonisüsteemidel ja keskkondades ühtmoodi. Lisaks on andmebaasi ülesseadmine ja konfigureerimine Docker-i konfiguratsioonifailidega oluliselt lihtsustatud.

Veebilehe serverimiseks kasutatakse NGINX-i [27], mis on tasuta, vabavaraline, kõrge jõudlusega HTTP server ja *reverse proxy* [27]. Autor valis NGINX-i kuna seda on lihtne üles seada ja suudab serverida mitut erinevat veebilehte samalt pordilt, mis on tähtis kuna DigitalOcean-i virtuaalmasin on kasutuses ka teiste veebilehtede serverimiseks. Rakendus kasutab ka NGINX-i *reverse proxy* võimalust, mida on vaja erinevatele allikatele tehtavate päringute tegemise lubamiseks, kuna rakenduse *backend* ja *frontend*

serveritakse erinevatelt portidelt. Lisaks hakkab NGINX automaatselt tööle peale serveri taaskäivitamist.

Süsteemi arhitektuuri kujutab järgnev joonis (Joonis 1).



Joonis 1. Süsteemi arhitektuur.

3.1.1 Serverirakenduse tööhoidmine

Backend rakendus kasutab Node.js-i [28], ning selle töös hoidmiseks peab kasutama mingit muud programmi, mis serverirakenduse taustale tööle jätab kui serveri käsurealiidestest lahkuda ning taaskäivitaks rakenduse, kui mingisugusel põhjusel protsess lõpetatakse.

Serverirakenduse töös hoidmiseks on mitmeid erinevaid võimalusi, mis langevad kahte kategooriasse: Linuxi tööriistad ja Node.js teigid. Kuna Linuxi tööriistad on standardiseeritud, ning neid on võimalik rakendada rohkematel otstarvetel, kui erinevaid Node.js programme (nodemon [29], PM2 [30], Forever [31]), siis otsustas autor valida systemd, nohupi ja Upstardi vahel.

systemd [32] on komplekt baas ehitusplokke Linux süsteemile. systemd annab süsteemi- ja teenusehalduri, mis jookseb PID (*process id*) 1-1 ja paneb tööle ülejäänud süsteemi [32].

nohup [33] käsk jooksub käsku, mille defineerib käsu parameeter ja kõiki sellega seonduvaid argument parameetreid, ignoreerides kõiki *hangup* (SIGHUP) signaale või modifitseerides spetsifitseeritud protsessi valikuga ignoreerida kõiki *hangup* (SIGHUP) signaale [33]. Kuna nohup käsuga jooksumatud programmid ignoreerivad *hangup* signaale, saab seda kasutada serveri taustal tööle panemiseks nii, et peale väljalogimist jääb server tööle.

Upstart [34] on sündmustepõhine asendus */sbin/init* taustaprotsessile, mis tegeleb ülesannete ja teenuste lähtestamisega süsteemi käivitamisel, nende peatamisega süsteemi seisatamisel ja nende jälgimisega süsteemi jooksmise ajal [34].

nohup üksi ei paku käsu uuesti jooksumist peale käsu ettearvamatutel põhjustel sulgumist, ning käsu tööle panekut peale süsteemi taas-käivitamist. Dockeri dokumentatsiooni sõnul kasutavad enamused uuemaid Linuxi distributsioone *systemd*-d, et hallata missugused teenused pannakse tööle süsteemi käivitamisel [35]. Upstart ja *systemd* pakuvad suhteliselt sama funktsionaalsust, ning see on ka installeeritud algselt virtuaalmasina operatsioonisüsteemile. Seetõttu kasutab süsteem lõputöö raames loodud rakenduse serveri jooksumiseks *systemd*-d, mida kasutades on *journalctl* [36] tööriistaga väga lihtsalt võimalik programmi logisid jälgida.

3.2 Kasutajapoolses rakenduses kasutatavad tehnoloogiad ja teegid

Kasutajapoolne (*frontend*) osa rakendusest on kirjutatud JavaScript-is [5], mis kasutab ECMAScript 2015 [37] standardit. *Frontend* kasutab ka JSX-i [38] mis on XML [39] laadne süntaks JavaScriptile, mida kasutatakse vaatekomponentide kirjeldamisel. Et ilma nende programmeerimiskeele lisade toeta brauserites rakendust kasutada konverteeritakse Babel-i [40] abil programmikood vanemale JavaScripti standardile.

Erinevate ekraanisuuruste toetamiseks ja ühtse stiili hoidmiseks rakenduses kasutatakse Bootstrap-i. Bootstrap on HTML-i, CSS-i ja JavaScripti raamistik ekraanisuuruste muutustele reageerivate, eesotsas mobiilidele mõeldud veebirakenduste ehitamiseks [41]. Bootstrapi poolt pakutavast kasutab rakendus vaid kohandatud CSS klasse.

3.2.1 React

React [42] on kasutajaliideste ehitamiseks mõeldud JavaScripti teek. React võimaldab disainida lihtsalt deklaratiivseid vaateid iga rakenduse oleku jaoks ning uuendab ja formuleerib täpselt õiged komponendid vastavalt muudatustele andmetes [42].

React kannab hoolt põhiliselt rakenduse vaadete eest ning seetõttu on arendajal võimalus valida teiste ülesannete jaoks rakenduses kasutatavad tehnoloogiad ise. Rakenduses kasutatakse React-i kasutajapoolses rakenduses vaadete esitamiseks.

3.2.2 Redux

Redux [43] on ennustatav olekukonteiner JavaScripti rakendustele. Redux aitab kirjutada järjekindlat ühtmoodi käituvaid rakendusi. Seda teeki kasutades asub kogu rakenduse olekupuud ühes kindlas hoidlas (*store*). Ainus viis olekupuud muuta on saata välja tegevus (*action*), mis kirjeldab juhtunut. Väljendamaks kuidas tegevus olekupuud muudab kirjutatakse reduktoreid (*reducer*), mis on puhtad funktsioonid (*pure function*). Puhas funktsioon tagastab alati sama sisendi korral sama väljundi. Reduktor ei tohi muundada olekupuud vaid peab tagastama alati uue olekupuud.

Lisana Reduxile kasutatakse rakenduses Redux Thunk [44] vahevara. Redux Thunk vahevara võimaldab kirjutada tegevuste loojaid (*action creators*), mis tagastavad funktsioone, mitte tegevusi. Seda tegevuste loojat saab kasutada tegevuse täitmise edasilükkamiseks või teha tegevus ainult juhul kui mingisugune tingimus on täidetud [44]. Redux Thunki kasutatakse rakenduses asünkroonsete JavaScripti päringutega saadud vastuste korral tegevuste täitmiseks ja ruumi liige olles mängija tegevustest teiste mängijate informeerimiseks samaaegselt.

Rakenduses kasutatavad reaalaaja-suhtluskanalid (WebSocket, WebRTC) on implementeeritud Reduxi vahevarana. Reduxi vahevara on esimene mida teavitatakse käivitatud tegevusest, vahevaral on juurdepääs rakenduse olekupuule, võime tegevustele reageerida ning kui käivitatud tegevus ei olnud mõeldud vahevarale, edastada tegevus järgmisele vahevarale või olekupuud reduktorile.

3.3 Serveripoolses rakenduses kasutatavad tehnoloogiad ja teegid

Serveripoolne osa rakendusest (*backend*) on kirjutatud Node.js-i [28] kasutades, mis on JavaScripti käituskeskkond, mis on ehitatud Chrome V8 [45] JavaScripti mootoril [28]. *Backend* on arendatud kasutades ECMAScript 2015 [37] standardit, mistõttu ka *backend*-i programmikood konverteeritakse toodangkeskkonnas (*production environment*) kasutamisel vanemale JavaScripti standardile. Arenduskeskkonnas kasutatakse toodangkeskkonnas mitte kasutamiseks mõeldud babel-node-i [46].

Rakenduse Postgres [26] andmebaasiga suhtlemiseks ja andmebaasimigratsioonide haldamiseks kasutatakse Knex.js-i [47]. Knex.js on SQL päringute konstrueerija erinevatele relatsioonilistele andmebaasisüsteemidele, mille hulka kuulub ka kasutatav Postgres [47].

3.4 Kommunikatsiooni-tehnoloogiad

Rakenduse tööks vajalikke, kuid mitte kiirelt muutuvaid andmeid küsitakse serverilt HTTP päringutega. Kiirelt muutuvate andmete (mänguruumid, ruumidega liitumise soovid) sünkroniseerimiseks rakendust kasutatavate klientidega kasutatakse WebSocket-eid [4]. Kogu ruumisine suhtlus toimub klientide vahel kasutades WebRTC [10] partnerühendusi.

WebSocket [4] on tehnoloogia, mis teeb võimalikuks avada interaktiivne kommunikatsioonisessioon brauseri ja serveri vahel. Kasutades WebSocket tehnoloogiat saab saata sõnumeid serverile ja saada välistest sündmustest tulenevaid vastuseid ilma, et peaks serverit pidevalt teatud ajavahemiku möödudes uuenenud informatsiooni küsima [4]. WebSocket tugi on globaalselt 94.16% veebibrauseritest [48]. Sarnaselt WebRTC kommunikatsioonile on ka WebSocket sõnumite saatmine ja vastuvõtmine toimingutest tulenev (*event based*). Laialdase toe ja reaalaaja suhtluse tõttu on see tehnoloogia hea alternatiiv kasutamiseks klientide puhul kelle brauserid ei toeta WebRTC-d.

3.4.1 WebRTC

WebRTC [10] on avatud lähtekoodiga veebiraamistik, mis võimaldab reaalaajasuhtlust brauserites. Raamistik sisaldab fundamentaalseid komponente kõrge-kvaliteediliseks

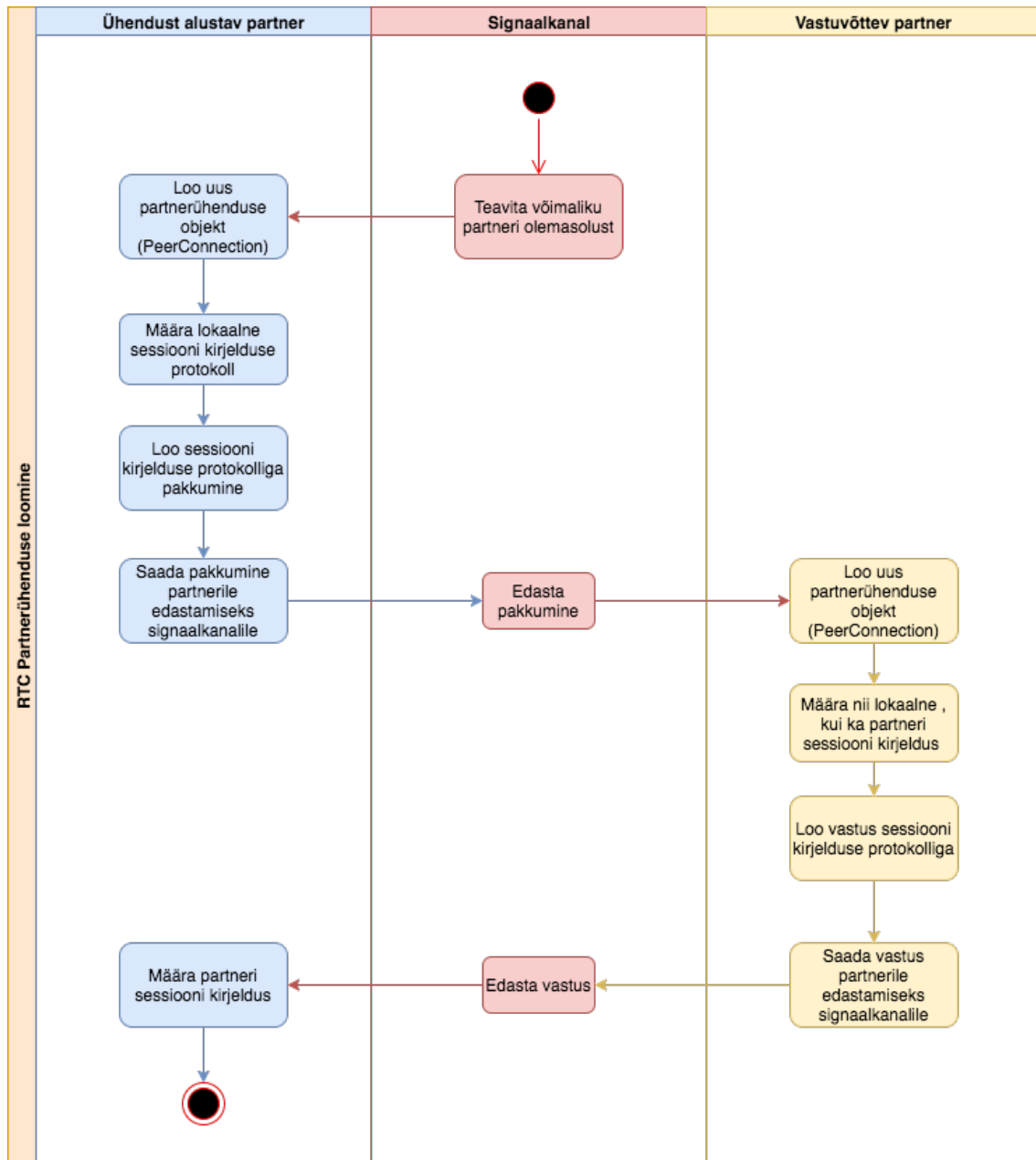
kommunikatsiooniks veebis, millele saab ligi läbi JavaScripti API, kui need komponendid brauseris implementeeritud on [1].

Vaatamata sellele, et WebRTC lubab rakenduse kasutajate brauseritel üksteisega suhelda saates informatsiooni üle partnerilt partnerile võrguühenduse (*peer-to-peer connection*), on siiski vaja ka välist signaalkanalit, et ühenduse alustamist koordineerida ning, et hakkama saada tulemüüride ja võrguaadresside tõlkimisega (NAT) [49].

Signaalkanal on vajalik reaalaja-kommunikatsiooniühenduste loomiseks võrgupartnerite vahel. Signaalkanal võib tehniliselt olla ka näiteks e-mail või tekstisõnum mobiilile kuid kasutajamugavuse ja ühenduste loomise kiiruse huvides kasutab rakendus signaalkanalina nii WebRTC-d kui ka WebSocket-eid.

WebRTC partnerühenduste tehnoloogiat kasutades peab arvestama brauseritega, millel puudub tugi antud tehnoloogiale. Tuntumatest platvormidest ja brauseritest ei toimi WebRTC nii iOS-i [50] kasutatavates seadmetes, kui ka macOS-il [51] Safari [52] brauseris. WebKit-i [2] funktsioonide seisundi informatsiooni kohaselt on WebRTC seisund WebKit-is “arendamisel” [53], seega võib tulevikus ka WebKit-i kasutatavates brauserites olla võimalik partnerühendusi luua.

Ühenduse loomist lihtsustatud kujul töö käigus valminud rakenduses kirjeldab alljärgnev joonis (Joonis 2). Joonisel on ignoreeritud ICE kandidaatide asünkroonset vahetust ja lisamist partnervõrgu ühendustele võrgupartnerite poolt.



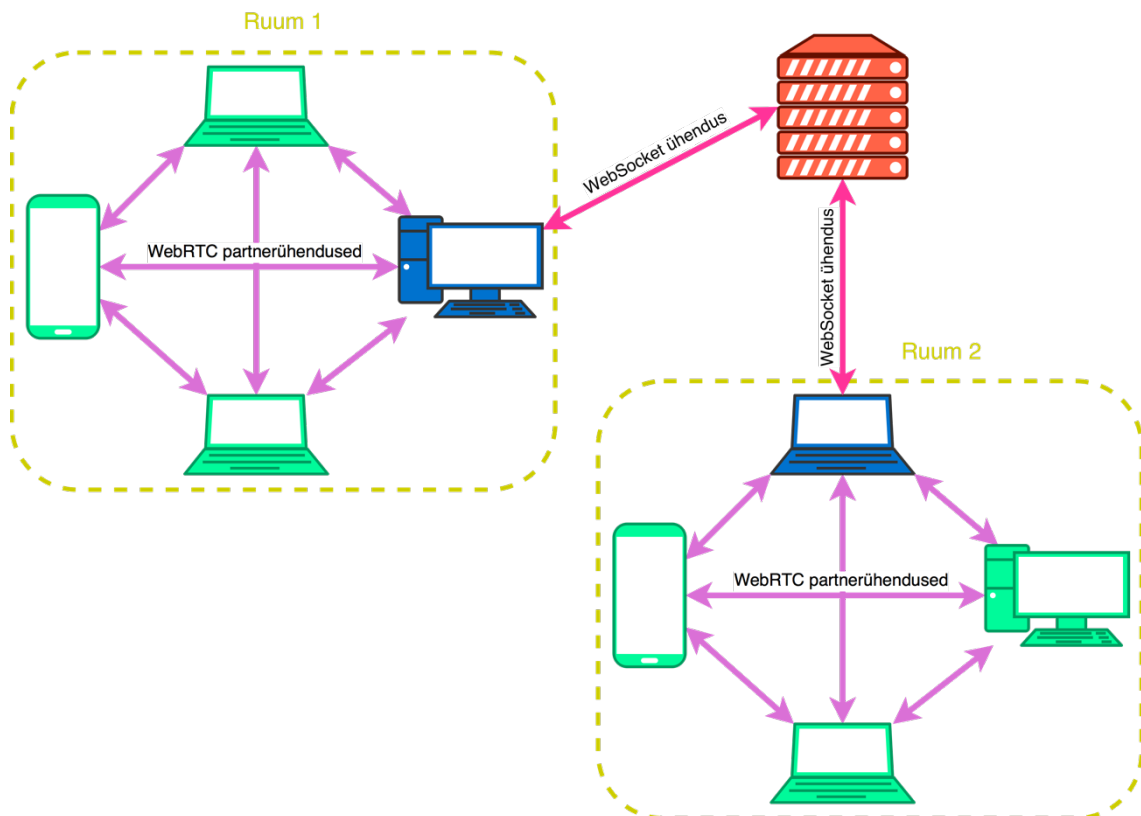
Joonis 2. RTC Partnerühenduse loomine.

3.5 Võrgutopoloogia

See peatükk kirjeldab kuidas vastavalt eelnevalt leitud nõuetele ja tehnoloogilistele piirangutele loodav partneritevaheline täisühendusvõrk välja näeb. Täisühendusvõrgu puhul ei teki olukorda, kus ühe võrguliikme lahkumisel võrk laguneks alamvõrkudeks. Tähtühenduse puhul laguneks kogu võrk juhul, kui lahkub keskne liige ning ringühendus võib kahe liikme lahkumisel enne ringiks taasühendamist laguneda kaheks

alamvõrguks. Täisühendusvõrgu eelis on ka võrgupartnerite võime saata informatsioon just seda vajavale võrgupartnerile ilma vahendajata. Täisühendusvõrkude puuduseks on vajadus omada ühendust iga võrgupartneriga, mistõttu on iga klienti poolt vajaminevate ühenduste arv võrdne teiste võrgupartnerite arvuga. Tähtühendus vajab vaid üht ja ringühendus kaht ühendust.

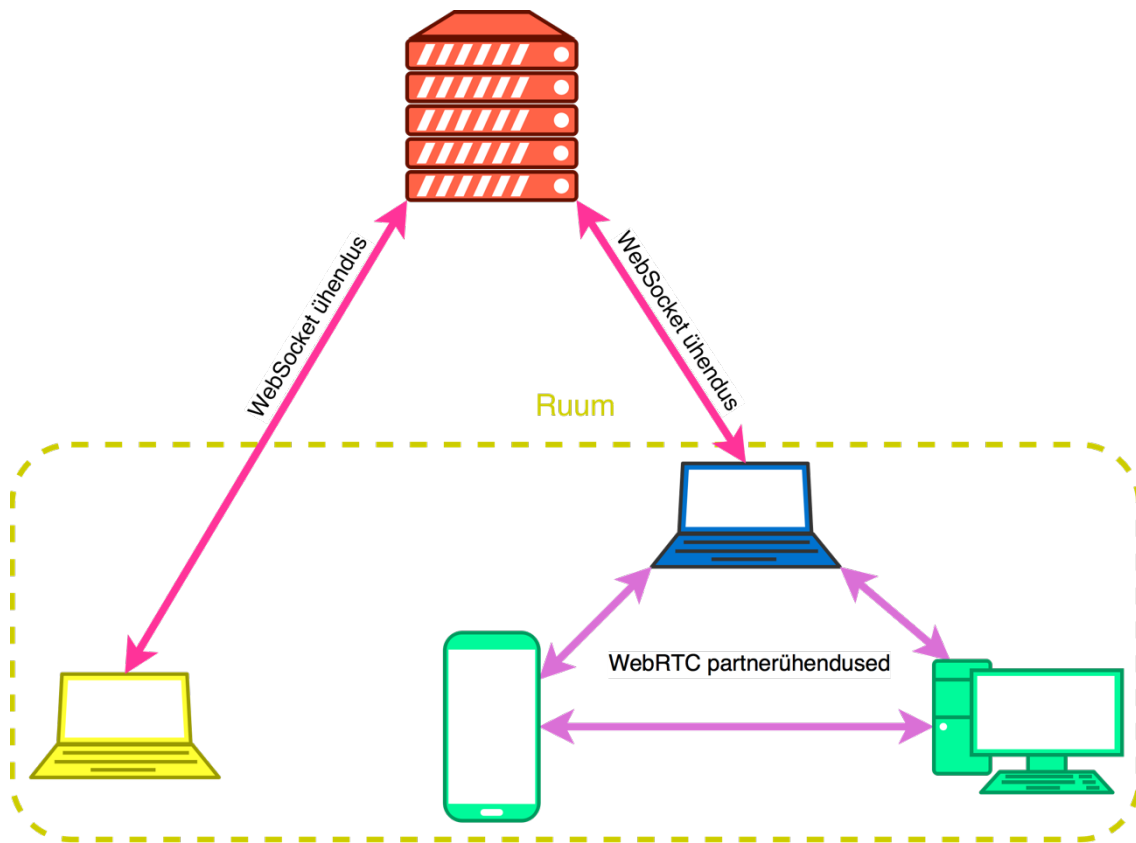
Võrgu topoloogia on kujutatud järgneval joonisel (Joonis 3).



Joonis 3. Võrgutopoloogia.

Rakenduse serveril (kujutatud joonisel punase värviga) on WebSocket ühendused (kujutatud roosa värviga) mänguruumide omanikega (kujutatud sinise värviga) WebRTC partnerühenduste loomiseks ning ruumidest suvalisel ajal lahkumise ning liitumise nõuete täitmiseks. Ruumiga liitumiseks edastab liituda sooviv kasutaja sõnumi rakenduse serverile, mis edastab soovi ruumi omanikule. Kasutades WebSocket serverit signaalkanalina luuakse partnerühendus ruumi omanikuga, misjärel suletakse liituva kasutaja ühendus tsentraalse serveriga ning luuakse partnerühendus kõigi teiste ruumi liikmetega kasutades signaalkanalina partnerühendust ruumi omanikuga.

Eelnevalt kirjeldatud võrgutopoloogia eeldab, et kõikide kasutajate brauserid toetavad kommunikatsiooni üle WebRTC. Toetamaks ka kasutajaid kelle veebibrauserid WebRTC-d ei toeta, pakub autor välja, et seesuguste kasutajate ja partnervõrku ühendatud kasutajate suhtlust võiks vahendada rakenduse server kasutades WebSocket tehnoloogiat. Sellise võrgu topoloogia kirjeldab alljärgnev joonis (Joonis 4).



Joonis 4. Võrgutopoloogia WebRTC toeta brauseri puhul.

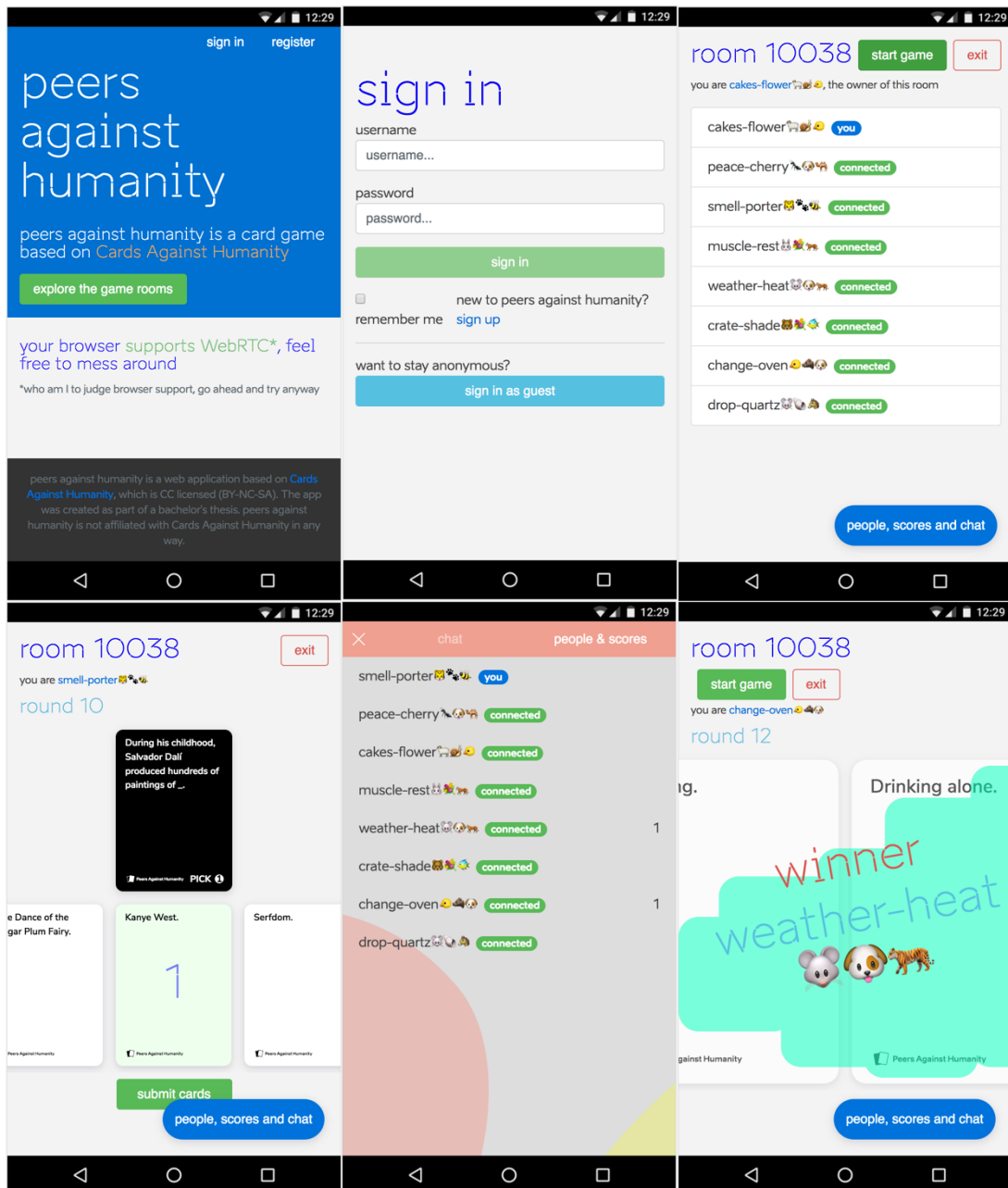
WebRTC tehnoloogiat mitte toetava brauseriga seade on joonisel kujutatud kollase värviga. WebRTC toeta brauserite puhul peab saatetavatel sõnumitel lisaks tüübile, saatjainfole ja sõnumi poolt sisaldatavale informatsioonile kaasas ka missugusele ruumile ja kellele ruumis sõnum mõeldud on kuna serveril puudub informatsioon ruumi liikmete kohta, kuna serveril on info ruumi omaniku kohta, võiks alternatiivina ruumi omanik partnerühenduseta kasutaja infot edasi saates edastada sõnumi kasutaja informatsiooni kasutades ja vastupidi. Joonisel 4 kujutatud viisi rakenduse ruumidega liituda rakenduses lõputöö raames implementeeritud ei ole.

4 Veebirakendus

Töö käigus valmis Cards Against Humanity lauamängul baseeruv veebirakendus. Rakendus implementeerib eelkirjeldatud võrgutopoloogiale vastavat täisühenduspartnervõrku ruumi liikmete vahel, kus ruumi omanik on ühendatud tsentraalse serveriga kasutades WebSocket ühendust. Kirjeldatud võimalus WebRTC toeta klientbrauseritel mänguruumidega ühenduda lõputöö käigus arendatud rakenduses puudub. Kogu mänguruumisisene suhtlus rakenduses toimub kasutades töö käigus arendatud täisühenduspartnervõrku. Tsentraalset serverit kasutab rakendus potentsiaalsete võrgupartnerite ühendamiseks ruumi omanikega, kasutajakontode ja ruumide haldamiseks ning mängukaartide hoiustamiseks ja jagamiseks ruumidele.

Arenduse käigus asetses töö fookus efektiivse ja lihtsa mitme võrgupartneri vahelist sidet algatava ruumide süsteemi loomisel. Seetõttu ei täida lõputöö raames arendatud rakendus mänguruumidele esitatud nõuet, mis ütleb, et peab olema võimalik luua privaatseid mängurume.

Valminud rakenduses implementeeritud funktsionaalsus on kasutatav ka WebRTC-d toetavatel mobiilibrauseritel ja rakenduse vaated on loodud eesmärgiga toetada erinevaid ekraanisuurusi. Alljärgnev joonis (Joonis 5) kirjeldab mõningate mobiilivaadete väljanägemist.



Joonis 5. Rakenduse mobiilivaated.

Mitmed mõjukad ja modernsed veebibrauserid, suuremad nende hulgas Safari ja mobiilne Safari, ei toeta veel WebRTC-d ning nende brauserite kasutajate jaoks tuleks luua võimalus üksteise vahel suhelda läbi tsentraalse serveri andmekanali. Lisaks toob partnervõrgu loomine ja kasutamine rakendusse väga palju lisaloogikat ja keerukust, kuna rakenduse olek sõltub mitmetest teistest võrgupartneritest, mitte tsentraalsele serverile teadolevast informatsioonist.

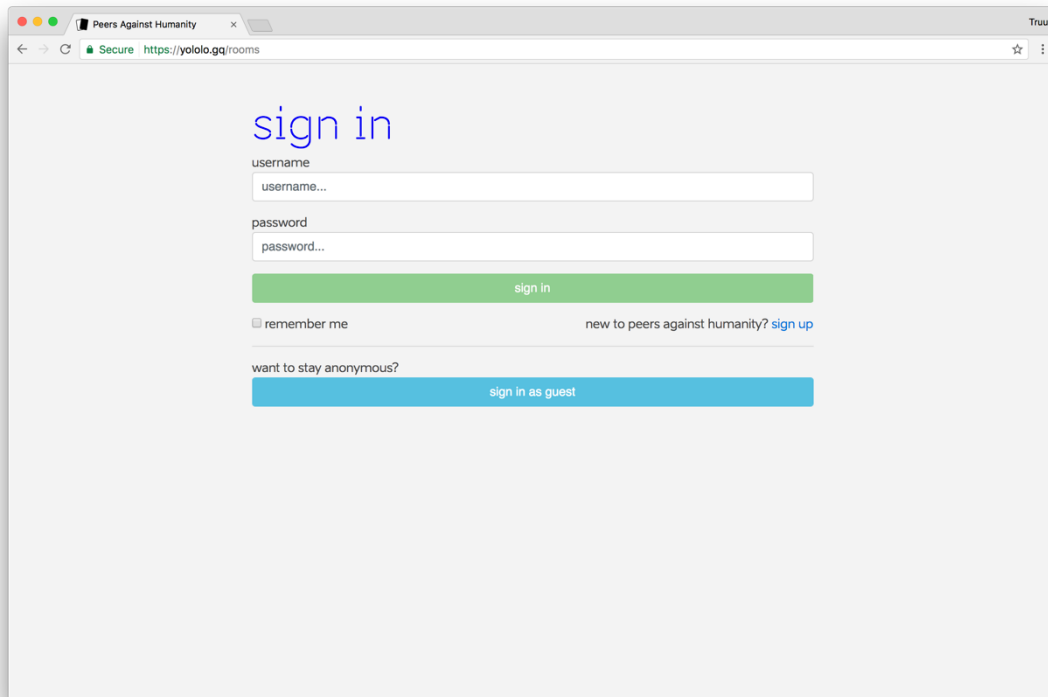
Kuna võrreldes WebSocket tehnoloogiaga on WebRTC brauserite toetus mitukümmend protsenti madalam ja partnervõrgu kasutamine nõuab ka teistsugust andmete lähenumist arendaja poolt, ei ole WebRTC veel iseseisvalt kasutamiseks reaaliajakommunikatsiooni nõudvates võrgurakendustes hea alternatiiv. Vaatamata sellele võimaldab WebRTC partnervõrgu loomine teha palju kasutajatevahelist suhtlust nõudvas rakenduses enamuse kommunikatsioonist pidada otse ja seeläbi mitte koormata serverit sõnumiedastusega. Kui tulevikus rohkemad brauserid WebRTC tehnoloogiat kasutada võimaldavad on see heaks lahenduseks mida kasutada soovides kulusid kokku hoida serveris, mille põhiline koormus tuleneb sõnumiedastusest.

4.1 Kasutajad

Rakendus võimaldab veebilehele sattunud kasutajatel sisse ja välja logida, soovi korral end külalisena sisse logida või registreerida uue salasõnaga konto. Loodud rakenduses on kasutajakontod põhiliselt vajalikud kasutajate leidmiseks teiste kasutajate poolt ja nende poolt loodud ruumide sidumiseks spetsiifiliste kasutajatega.

Rakendust edasi arendades saab kasutada kontosid ka kasutajate sidumiseks nende poolt loodud mängukaartidega ning mängutulemuste salvestamiseks globaalse skooritabeli jaoks. Rakenduse kasutamise käigus ilmnes, et külaliskonto loomise samm on ebavajalik, seda võiks teha alati kui kasutaja lehele ilma eelnevalt salvestatud kontota satub, et eemaldada üks samm kasutaja ja mänguni jõudmise vahelt. Kasutajal on alati võimalus hiljem soovi korral sama konto registreerida.

Külalisena sisse logimine loob uue salasõnata konto, ning kui seda brauseri poolt meelde ei jäeta, ei ole sama konto kasutamine järgnevatel sessioonidel enam võimalik. Kui kasutaja otsustab hiljem end registreerida brauseri poolt meelde jäetud külaliskontot omades määrab kasutaja registreerides külaliskontole salasõna ja kasutajanime misjärel on konto registreeritud. Kasutajanimed on rakenduses unikaalsed kuid hüüdnimel unikaalsuse piirang puudub. Eelistatuna näidatakse teistele klientidele kasutaja hüüdnimi, selle puudumisel konto kasutajanimi. Joonisel (Joonis 6) on kujutatud autentimist nõudvale veebiaadressi rajale sattudes autentimata kasutajatele näidatav vaade. Edukal autentimisel suunatakse kasutaja edasi lehele millele ta parajasti juurde pääseda soovib.



Joonis 6. Sisselogimise vaade.

4.2 Ruumid

Ruumide vaates tegutsemiseks peab kasutaja olema autenditud ja autoriseeritud. Vastavalt funktsionaalsetele nõuetele peavad kasutajad saama ruume luua, nendega liituda ja neist suvalisel ajal lahkuda. Ruumide vaade kasutab WebSocket serverit, et mitte ühes ruumis viibivatele kasutajatele informatsiooni reaalajas edastada. Kõik ruumidega seonduvad operatsioonid viiakse läbi kasutades kas WebSocket ühendust või WebRTC partnervõrku.

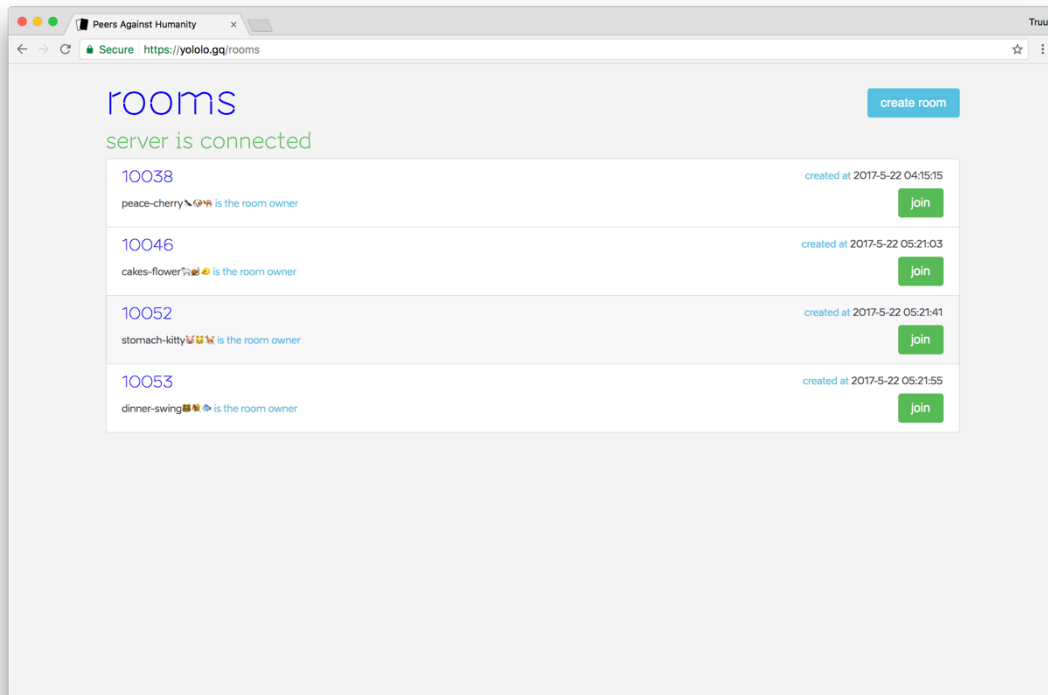
Esialgne plaan oli ruumi kohta hoida kogu vajaminevat informatsiooni mänguruumi oleku (kas mäng on alustatud, lõpetatud, kes on liikmed) kohta hoida tsentraalse serveri andmebaasis. Selline info dubleerimine oli arendusaega ja serveriressursse nõudev, seetõttu hoiab valminud rakenduses kogu ruumisiseselt vajalikku informatsiooni ruumi omanik. Server teab ainult mis on ruumi number, kes on omanik ning kas ruum on aktiivne, et olla võimeline ruumidega liituda soovijate ja ruumi omanike vahelisi

ühenduse algatamise sõnumeid vahetama. Selle valiku tõttu on rakendusel liiga vähe informatsiooni, et kuvada ruumide vaates olevatele kasutajatele välja nimekirjas olevate ruumide olek ja ruumi liikmete arv. Probleemi lahendamiseks peaks rakendust edasi arendades viima kogu ruumidega seonduva info ruumi liikmete kätte ja serverirakenduses genereerima ruumile vaid unikaalse identifikaatori. Vajadusel küsiks serverirakendus ruumide informatsiooni ruumi omanikelt. Ruumi omanikud saaksid ka ise uuenenud ruumist serverit teavitada, mis seejärel edastataks info ruumide vaates olevatele kasutajatele.

Uus mänguruum luuakse saates vastavat tüüpi sõnum rakenduse WebSocket serverile. Avatuna hoitavate WebSocket ühenduste hulga minimaalsena hoidmiseks ja sõnumite edastamiseks kindlatele kasutajatele on ühel kasutajakontol lubatud korraga lahti hoida vaid üht WebSocket ühendust. Kuna ruumi omanik olemine tähendab pidevat ühendust WebSocket serveriga, on sellest tulenevalt ühel kasutajal omada korraga maksimaalselt ühte ruumi. Juhul kui ilmub uus aktiivne ruum saadab server kõigile ruumide vaates olevatele (WebSocket ühendusega klientidele) sõnumi uue ruumi infoga.

Ruumide vaates näeb kasutaja nimekirja aktiivsetest ruumidest. Aktiivsete ruumide hulgas on ruumid, kus mäng on juba alanud ja ruumid, milles olevad kasutajad ootavad mängu alustamist. Kasutajal on lubatud liituda ruumidega milles mängimist veel alustatud ei ole või ruumidega milles on mängimist alustatud ja millest kasutaja mängu käigus lahkus. Kasutaja võib selles vaates luua ka uue ruumi. Ruumide vaates olevad kasutajad on alati ühendatud WebSocket serveriga.

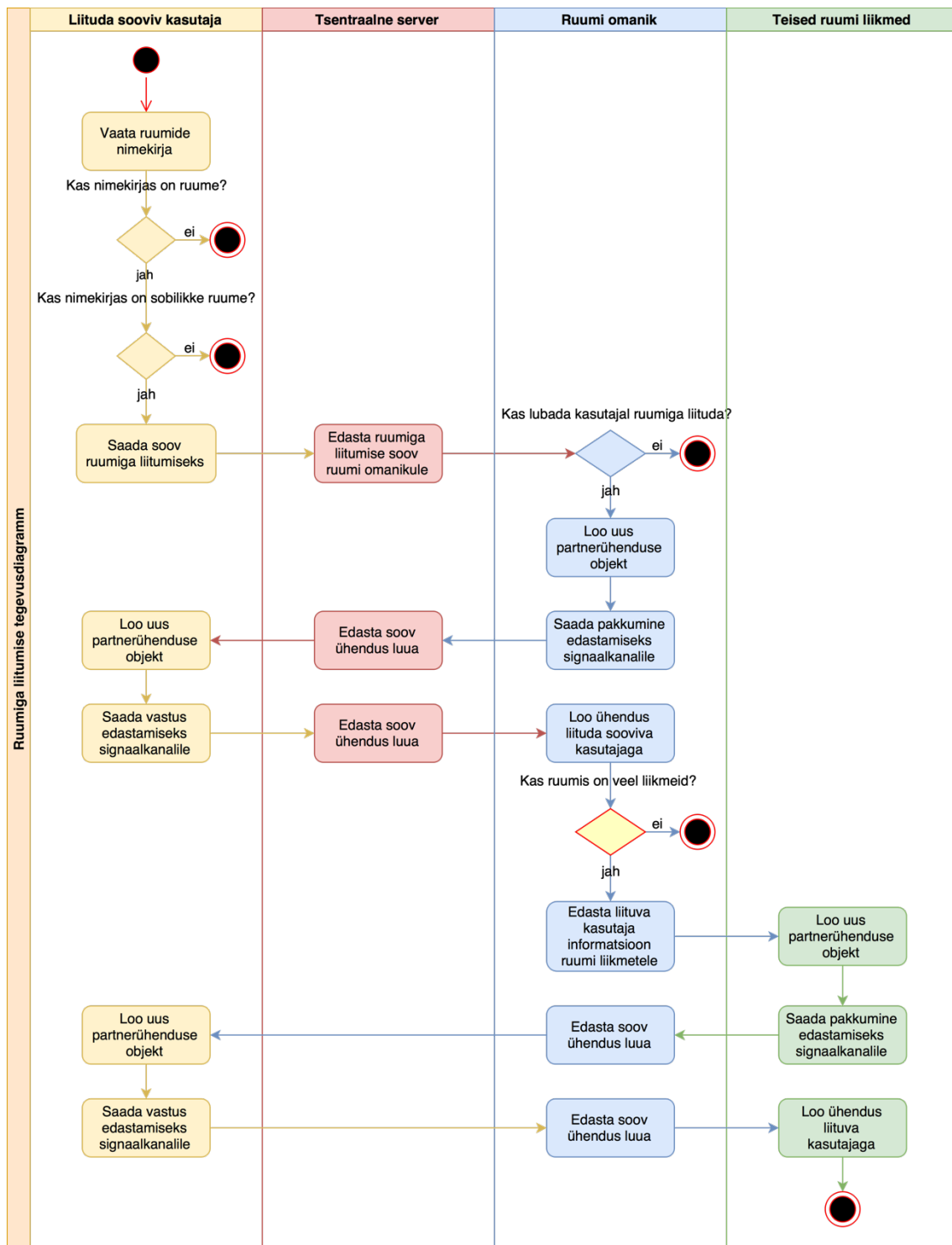
Ruumide vaatesse jõudnud kasutajale avaneb alljärgneval joonisel (Joonis 7) olev pilt.



Joonis 7. Ruumide vaade.

4.2.1 Ruumiga liitumine

Ruumiga liitumiseks saadab kasutaja sõnumi liitumissooviga WebSocket serverile. Server edastab ruumi omanikule kasutaja soovi ruumiga liituda. Kui ruumi omanik otsustab lubada kasutajal ruumiga liituda, loob ta kasutajaga uue partnerühenduse ning saadab kasutaja informatsiooni ka teistele ruumi liikmetele, kes omakorda loovad liituvat kasutajaga partnerühendused. Peale seda protsessi on kasutaja täisõiguslik ruumi liige. Ruumiga liitumise tegevusdiagramm on kujutatud alljärgneval joonisel (Joonis 8).



Joonis 8. Ruumiga liitumise tegevusdiagramm.

4.2.2 Ruumist lahkumine

Kasutaja lahkub ruumist kas omal vabal tahtel või juhul, kui sulgetakse brauseri aken või veebilehte sisaldav brauserikaart, mille tõttu sulgetakse kasutaja partnerühendused

ja olemasolul WebSocket ühendus serveriga. Vabatahtlikult ruumist lahkudes sulgeb kasutaja automaatselt partnerühendused kõikide ruumi liikmetega.

Kui ruumist lahkub ruumi omanik, edastab ta lisaks ühenduste sulgemisele ka tsentraalsele serverile soovi ruumist lahkuda. Ruum, millest lahkub ruumi omanik määratakse serveris mitteaktiivseks. Kui ruumis, millest lahkub omanik, on ka teisi liikmeid, võtab üks ruumi liikmetest automaatselt ruumi omaniku rolli üle, avades WebSocket ühenduse ning kooskõlastades ruumi ülevõtmise serveriga.

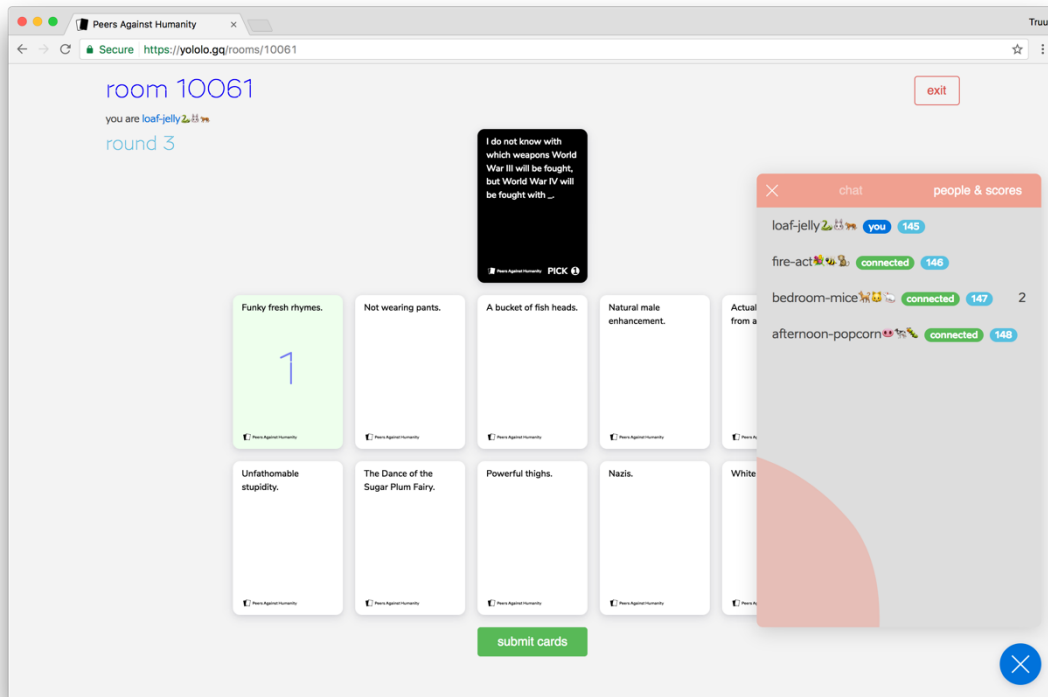
4.3 Mäng

Mängu vaatesse sattunud kasutaja on juba eelnevalt autenditud, autoriseeritud ja ruumi liige. Lõputöö käigus arendatud rakenduses implementeeritud mängu parameetrid on ettemääratud ning ei ole kasutajale seadistatavad.

Mängus eristatakse kahte tüüpi mängijaid: mänguringi hindaja ja tavaline mängija (valib kaarte mida hindaja hindab). Minimaalne mängijate arv mängu alustamiseks on kolm. Mänguringi hindaja ja tavaline mängija on üksteist välistavad rollid.

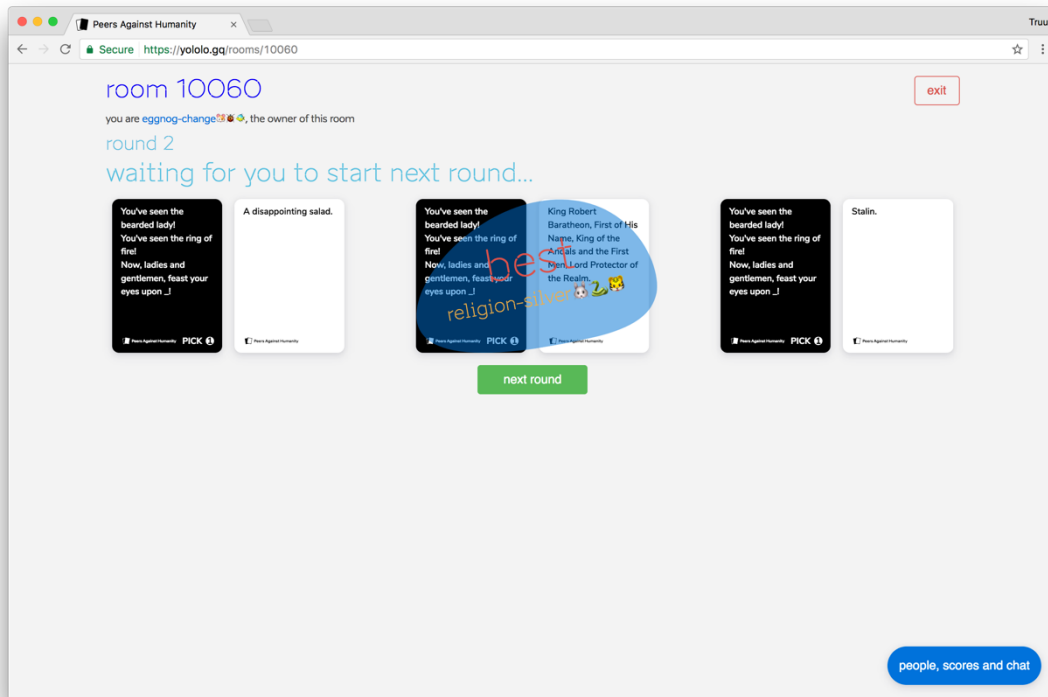
Mängu alustamisel valib ruumi omanik suvaliselt esimese hindaja ja mängijate järjekorra ning edastab selle info mängijatele. Eelnev hindaja alustab uut mänguringi, ehk valib endale määratud kaartide seast uue musta kaardi ning edastab selle mängijatele. Esimese ringi puhul alustab mänguringi ruumi omanik. Mängijad valivad kaardi või kaardid ning saadavad kõigile valitud kaardi või kaardid. Kõik mängijad segavad vastused ära. Kaardid näidatakse välja alles siis, kui kõikide aktiivsete mängijate kaardid on kohale jõudnud. Hindaja valib kõige naljakama kombinatsiooni ning edastab kõigile informatsiooni valiku kohta.

Mitte-hindajast mängija vaade on kujutatud alljärgneval joonisel (Joonis 9).



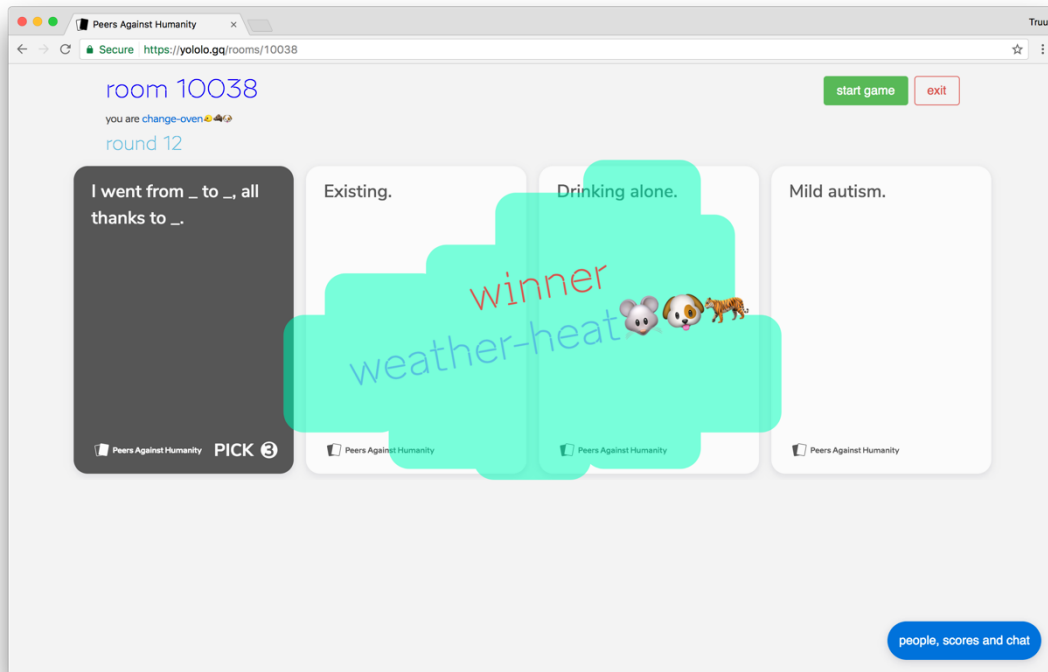
Joonis 9. Tavamängija vaade mängus.

Loodud rakenduses peab hindaja manuaalselt järgmise mänguringi alustama vajutades joonisel (Joonis 10) näha olevat rohelist nuppu. Hindaja vaade on kujutatud järgneval joonisel (Joonis 10).



Joonis 10. Hindajast mängija vaade mängus.

Kui üks mängijatest saab kokku 3 punkti kuvab rakendus välja võitja nime ning viimased kaardid mille võitja valis. Joonisel (Joonis 11) kujutatud mängu lõpu vaate kuvab rakendus kõigile mängijatele ning peale mängu lõpetamist on mänguruumi omanikul võimalus alustada samas ruumis uut mängu.



Joonis 11. Vaade peale võitja selgumist mängus.

4.3.1 Vajalik kaartide arv mängus

Kuigi rakenduses on olemasolevate kaartide arv piiratud, võib kaartide arv rakenduses olla kui tahes suur, sest neid võib alati andmebaasi juurde lisada. Igasse mänguruumi on vaja vaid alamhulka kõikidest olemasolevatest kaartidest ja üleliigse andmesidemahu vältimiseks peab välja arvutama kui palju kaarte on maksimaalselt vaja ühes mängus ja olema kindel, et kogus kaarte mängijate arvule on olemas. Lisaks peab vältima olukorda kus mängu kestel kaartide otsa saamise tõttu mängu jätkata ei saa. Maksimaalse vajaminevate kaartide arvu välja arvutamiseks peab kõigepealt leidma mitu mänguringi tehakse maksimaalselt teatud arvu punktideni mängimisel.

Kõigil mängijatel on võimalik saada maksimumpunktidest üks punkt vähem kuni üks mängijatest maksimumpunktid saab ja mängu võidab. Seega on maksimaalne mänguringide arv mängus, kus mängitakse teatud maksimaalse punktide arvuni arvutatav valemiga (1), kus “r” on mänguringide arv, “m” on mängijate arv ja “p” on maksimaalne punktide arv. Mängus vajatav mustade kaartide arv on võrdne mänguringide arvuga.

$$r = m(p - 1) + 1 \quad (1)$$

Mängu esimeses ringis jagatakse kõigile kümme kaarti, misjärel vajab iga mängija kaarte juurde peale nende esitamist, mida võib igas mänguringis olla maksimaalne mustal kaardil olev lünkade arv. Igas mänguringis on üks mängijatest hindaja, kes kaarte ei vaja. Valemi (2) abil saab välja arvutada maksimaalse vajamineva valgete kaartide arvu mängu jooksul. Valemis (2) on “ k_{max} ” maksimaalne vajaminev valgete kaartide arv, “ m ” on mängijate arv, “ r ” on mänguringide arv ja “ v_{max} ” maksimaalne mustal kaardil olev lünkade arv.

$$k_{max} = m(10 + (r - 1)v_{max}) - rv_{max} \quad (2)$$

Näiteks, kui mängus on 12 mängijat, kes mängivad kuni üks neist saab 5 punkti, on maksimaalne vajalik valgete kaartide arv mängus 1704 kaarti. Seega on igal mängijal vaja 142 erinevat kaarti.

Nende valemite põhjal küsib ruumi omanik serverilt mängu alguses vajaliku hulga kaarte ning jagab need mängijatele laiali.

5 Kokkuvõte

Käesoleva töö käigus analüüsiti olemasolevaid Cards Against Humanity-l baseeruvaid võrgurakendusi. Uuringu käigus selgus, et rakendustel on probleeme serverite koormuse madalana hoidmisega. Kõigis rakendustes oli kasutajakontode loomine tehtud väga lihtsaks. Kahes rakenduses kasutati kasutajale nähtavaid mänguruume, üks lõi ruumid ja sobitas kasutajaid automaatselt. Rakendused lubasid mänguruumidega liituda ka kasutades URL-i. Mängu ajal lubas mänguga liituda üks uuritud rakendustest.

Töö tulemusena töötati välja leitud nõuetele vastav reaaliajalist tekstipõhist andmeedastust võimaldava täisühendusvõrgu tehniline lahendus Redux-i vahevarana WebRTC partnerühenduste põhjal ja arendati seda lahendust kasutav Cards Against Humanity-l baseeruv võrgurakendus, milles kogu ruumisene klientbrauserite vaheline kommunikatsioon peetakse kasutades loodud partnervõrku. Rakenduse tsentraalne server tegeleb kasutajakontode ja ruumide haldamise, potentsiaalsete partnerite esmase ühendamise ja ruumi omanikele mängu alustamisel mängukaartide jagamisega.

Partnervõrgu loomine ja kasutamine toob rakendusse väga palju lisaloogikat ja keerukust, ning rakenduse olek sõltub mitmetest teistest võrgupartneritest, mitte tsentraalsele serverile teadolevast informatsioonist. Lisaks mõned mõjukad ja modernsed veebibrauserid ei toeta veel WebRTC-d ning nende brauserite kasutajate jaoks tuleks luua võimalus üksteise vahel suhelda läbi tsentraalse serveri andmekanali. Vaatamata sellele võimaldab tehnoloogia palju kasutajatevahelist suhtlust nõudvas rakenduses enamuse kommunikatsioonist pidada otse ja seeläbi mitte koormata serverit sõnumiedastusega.

Kasutatud kirjandus

- [1] The WebRTC project authors, „Frequent Questions | WebRTC,“ [Võrgumaterjal]. Available: <https://webrtc.org/faq/>. [Kasutatud 10 4 2017].
- [2] Apple Inc., „WebKit,“ [Võrgumaterjal]. Available: <https://webkit.org/>. [Kasutatud 09 05 2017].
- [3] Comodo CA Ltd., „HTTP to HTTPS | What is a HTTPS certificate,“ [Võrgumaterjal]. Available: <https://www.instantssl.com/ssl-certificate-products/https.html>. [Kasutatud 09 05 2017].
- [4] Mozilla Developer Network and individual contributors, „WebSockets - Web APIs | MDN,“ [Võrgumaterjal]. Available: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API. [Kasutatud 26 04 2017].
- [5] Mozilla Developer Network and individual contributors, „About Javascript - Javascript | MDN,“ [Võrgumaterjal]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript. [Kasutatud 09 05 2017].
- [6] Infospace Holdings LLC, „How Network Address Translation Works | HowStuffWorks,“ [Võrgumaterjal]. Available: <http://computer.howstuffworks.com/nat.htm>. [Kasutatud 14 05 2017].
- [7] Cards Against Humanity LLC., „Cards Against Humanity,“ [Võrgumaterjal]. Available: <https://cardsagainsthumanity.com/>. [Kasutatud 30 04 2017].
- [8] BoardGameGeek, LLC., „Cards Against Humanity | Board Game | BoardGameGeek,“ [Võrgumaterjal]. Available: <https://boardgamegeek.com/boardgame/50381/cards-against-humanity>. [Kasutatud 10 05 2017].
- [9] Creative Commons Corporation, „Creative Commons - Attribution-NonCommercial-ShareAlike 2.0 Generic - CC BY-NC-SA 2.0,“ [Võrgumaterjal]. Available: <https://creativecommons.org/licenses/by-nc-sa/2.0/>. [Kasutatud 19 05 2017].
- [10] The WebRTC Project authors, „WebRTC Home | WebRTC,“ [Võrgumaterjal]. [Kasutatud 21 05 2017].
- [11] A. Deveria, „Can I use... Support tables for HTML5, CSS3, etc,“ [Võrgumaterjal]. Available: <http://caniuse.com/#feat=rtcpeerconnection>. [Kasutatud 08 05 2017].
- [12] Google Inc, „Google,“ [Võrgumaterjal]. Available: www.google.com. [Kasutatud 14 04 2017].
- [13] Apple Inc., „Games - App Store Downloads on iTunes,“ [Võrgumaterjal]. Available: <https://itunes.apple.com/ee/genre/ios-games/id6014?mt=8>. [Kasutatud 09 05 2017].
- [14] Google Inc, „Android Apps on Google Play,“ [Võrgumaterjal]. Available: <https://play.google.com/store/apps>. [Kasutatud 09 05 2017].
- [15] Individual reddit users, „A Subreddit About a Card Game for Horrible People,“ [Võrgumaterjal]. Available: <https://www.reddit.com/r/cardsagainsthumanity/>.

- [Kasutatud 09 05 2017].
- [16] D. Whitfield, „Cards Against Originality,“ [Võrgumaterjal]. Available: <http://www.cardsagainstorignality.com/>. [Kasutatud 11 05 2017].
 - [17] Andy Janata, „Pretend You're Xyzzy,“ [Võrgumaterjal]. Available: <http://pretendyoure.xyz/zy/>. [Kasutatud 17 04 2017].
 - [18] M. Silverstein, W. Ngo, T. McGinnis ja D. Gonzales, „Cards for Humanity,“ [Võrgumaterjal]. Available: <http://cfh.io>. [Kasutatud 17 04 2017].
 - [19] azala.info (pseudonüüm), „World's Filthiest Cards Against Humanity Online,“ [Võrgumaterjal]. Available: azala.info. [Kasutatud 17 04 2017].
 - [20] azala.info (pseudonüüm), „Azala on Twitter: "Mobile View now scales correctly without destroying the layout for everything else on phones.",“ [Võrgumaterjal]. Available: <https://twitter.com/AzalaDotInfo/status/859012585244344320>. [Kasutatud 09 05 2017].
 - [21] Digital Ocean, Inc, „DigitalOcean: Cloud computing designed for developers,“ [Võrgumaterjal]. Available: <https://www.digitalocean.com/>. [Kasutatud 09 05 2017].
 - [22] Canonical Ltd, „The leading operating system for PCs, IoT devices, servers and the cloud | Ubuntu,“ [Võrgumaterjal]. Available: <https://www.ubuntu.com/>. [Kasutatud 19 05 2017].
 - [23] Software in the Public Interest, Inc., „Debian -- The Universal Operating System,“ [Võrgumaterjal]. [Kasutatud 21 05 2017].
 - [24] Software in the Public Interest, Inc. and others, „Debian -- Debian "Jessie" release information,“ [Võrgumaterjal]. Available: <https://www.debian.org/releases/jessie/>. [Kasutatud 14 05 2017].
 - [25] Docker Inc., „Docker - Build, Ship, and Run Any App, Anywhere,“ [Võrgumaterjal]. Available: <https://www.docker.com/>. [Kasutatud 14 05 2017].
 - [26] The PostgreSQL Global Development Group, „PostgreSQL: The world's most advanced open source database,“ [Võrgumaterjal]. Available: <https://www.postgresql.org/>. [Kasutatud 14 05 2017].
 - [27] NGINX Inc., „Welcome to NGINX Wiki! | NGINX,“ [Võrgumaterjal]. Available: <https://www.nginx.com/resources/wiki/>. [Kasutatud 12 05 2017].
 - [28] Node.js Foundation, „Node.js,“ [Võrgumaterjal]. Available: <https://nodejs.org/en/>. [Kasutatud 10 05 2017].
 - [29] R. Sharp, „remy/nodemon: Monitor for any changes in your node.js application and automatically restart the server - perfect for development,“ [Võrgumaterjal]. Available: <https://github.com/remy/nodemon>. [Kasutatud 14 05 2017].
 - [30] A. Strzelewicz, „PM2 - Advanced Node.js process manager,“ [Võrgumaterjal]. Available: <http://pm2.keymetrics.io/>. [Kasutatud 14 05 2017].
 - [31] C. Robbins ja Contributors, „foreverjs/forever: A simple CLI tool for ensuring that a given script runs continuously (i.e. forever),“ [Võrgumaterjal]. Available: <https://github.com/foreverjs/forever>. [Kasutatud 14 05 2017].
 - [32] freedesktop.org, „systemd,“ [Võrgumaterjal]. Available: <https://www.freedesktop.org/wiki/Software/systemd/>. [Kasutatud 18 04 2017].
 - [33] IBM, „IBM Knowledge center - nohup Command,“ [Võrgumaterjal]. Available: https://www.ibm.com/support/knowledgecenter/en/ssw_aix_71/com.ibm.aix.cmds4/nohup.htm. [Kasutatud 18 04 2017].

- [34] Canonical Ltd., „upstart - event-based init daemon,“ [Võrgumaterjal]. Available: <http://upstart.ubuntu.com/>. [Kasutatud 18 04 2017].
- [35] Docker Inc., „Post-installation steps for Linux - Docker Documentation,“ [Võrgumaterjal]. Available: <https://docs.docker.com/engine/installation/linux/linux-postinstall/#configure-docker-to-start-on-boot>. [Kasutatud 18 04 2017].
- [36] freedesktop.org, „journalctl,“ [Võrgumaterjal]. Available: <https://www.freedesktop.org/software/systemd/man/journalctl.html>. [Kasutatud 21 05 2017].
- [37] Ecma International, „ECMAScript 2015 Language Specification - ECMA-262 6th Edition,“ [Võrgumaterjal]. Available: <https://www.ecma-international.org/ecma-262/6.0/>. [Kasutatud 14 05 2017].
- [38] Facebook, „JSX in Depth - React,“ [Võrgumaterjal]. Available: <https://facebook.github.io/react/docs/jsx-in-depth.html>. [Kasutatud 21 05 2017].
- [39] W3C, „Extensible Markup Language (XML),“ [Võrgumaterjal]. Available: <https://www.w3.org/XML/>. [Kasutatud 21 05 2017].
- [40] S. McKenzie, „Babel • The compiler for writing next generation JavaScript,“ [Võrgumaterjal]. Available: <https://babeljs.io/>. [Kasutatud 14 05 2017].
- [41] Twitter, Inc, „Bootstrap • the most popular HTML, CSS, and JS framework in the world.,“ [Võrgumaterjal]. Available: <http://v4-alpha.getbootstrap.com/>. [Kasutatud 10 05 2017].
- [42] Facebook, Inc, „A declarative, efficient, and flexible JavaScript library for building user interfaces.,“ [Võrgumaterjal]. Available: <https://github.com/facebook/react>. [Kasutatud 10 05 2017].
- [43] D. Abramov, „Predictable state container for JavaScript apps.,“ [Võrgumaterjal]. Available: <https://github.com/reactjs/redux>. [Kasutatud 10 05 2017].
- [44] D. Abramov, „Thunk middleware for Redux,“ [Võrgumaterjal]. Available: <https://github.com/gaearon/redux-thunk>. [Kasutatud 10 05 2017].
- [45] Google Inc, „Chrome V8 | Google Developers,“ [Võrgumaterjal]. Available: <https://developers.google.com/v8/>. [Kasutatud 21 05 2017].
- [46] S. McKenzie, „CLI • Babel,“ [Võrgumaterjal]. Available: <http://babeljs.io/docs/usage/cli/#babel-node>. [Kasutatud 14 05 2017].
- [47] T. Griesser, „Knex.js - A SQL Query Builder for JavaScript,“ [Võrgumaterjal]. Available: <http://knexjs.org/>. [Kasutatud 10 05 2017].
- [48] A. Deveria, „Can I use... Support tables for HTML5, CSS3, etc,“ [Võrgumaterjal]. Available: <http://caniuse.com/#search=websocket>. [Kasutatud 9 05 2017].
- [49] S. Dutton, „WebRTC in the real world: STUN, TURN and signaling - HTML5 Rocks,“ [Võrgumaterjal]. Available: <https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/>. [Kasutatud 12 04 2017].
- [50] Apple, Inc., „iOS 10 - Apple,“ [Võrgumaterjal]. Available: <https://www.apple.com/lae/ios/ios-10/>. [Kasutatud 21 05 2017].
- [51] Apple Inc., „macOS - Apple,“ [Võrgumaterjal]. Available: <https://www.apple.com/lae/macos/sierra/>. [Kasutatud 21 05 2017].
- [52] Apple Inc., „macOS - Safari - Apple,“ [Võrgumaterjal]. Available: <https://www.apple.com/lae/safari/>. [Kasutatud 21 05 2017].

- [53] Apple Inc., „WebKit Feature Status | WebKit,“ [Võrgumaterjal]. Available: <https://webkit.org/status/#specification-webrtc>. [Kasutatud 12 04 2017].
- [54] Cards Against Humanity LLC., „Cards Against Humanity Rules,“ [Võrgumaterjal]. Available: http://s3.amazonaws.com/cah/CAH_Rules.pdf. [Kasutatud 02 05 2017].