

Petteri Vermilä
Tommi Keskitalo

Verkkopohjainen karaokejärjestelmä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

26.11.2015

Tekijät Otsikko	Tommi Keskitalo, Petteri Vermilä Verkkopohjainen karaokejärjestelmä
Sivumäärä Aika	52 sivua 26.11.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja	Lehtori Kimmo Sauren
<p>Insinööriyössä suunniteltiin ja dokumentoitiin uudenlainen karaokejärjestelmä, joka toisi palvelun suoraan kuluttajan henkilökohtaiseen äylaitteeseen.</p> <p>Työssä selvitettiin kilpailevien ratkaisujen kautta keskeiset karaokejärjestelmän ominaisuudet, joista koottiin moderni kokonaisuus, jonka lähtökohtana oli käyttäjälähtöisyys. Pääominaisuuksien kartoituksen jälkeen pystyttiin päättämään käytettävät tekniikat ja komponentit, kuten Bootstrap, AngularJS ja Symfony, sekä suunnittelemaan kilpailukykyinen tuote. Selvityksen aikana huomattiin nopeasti tämäntasoisissa projekteissa tuotteiden vaatimustason olevan erittäin haastava.</p> <p>Alun perin tuotetta suunniteltiin paikallisen lähiverkon palveluna, mutta kehityksen edetessä järjestelmä päätettiin toteuttaa pilvipalveluna, jotta siihen pääsisi käsiksi mistä tahansa ja milloin tahansa, kunhan käytössä on internetyhteys. Ensin suunnitelmasta toteutettiin prototyyppi, jossa on pelkistetty käyttöliittymä. Se toimi esittelyversiona ja todisteena konseptin toimivuudesta. Tämän jälkeen aloitettiin varsinaisen sovelluksen kehitys. Järjestelmään suunniteltiin paljon eri ominaisuuksia, joista muun muassa laulujono ja karaokekappaleiden toistaminen toteutettiin ensimmäiseen versioon ja loput dokumentoitiin jatkokehitystä varten.</p> <p>Suunnitteluun panostettiin parhaan käyttäjäkokemuksen saavuttamiseksi kaikilla tuetuilla päätelaitteilla, ja työn aikana kartutettiin arvokasta tietotaitoa pilvipalveluratkaisujen kehittämistä. Lisäksi tuotteistamisen havaittiin olevan iso osa kokonaisvaltaista tuotekehitystä, jota on mietittävä jo heti alkuvaiheessa — kuten kuka tai mikä on tuotteen kohderyhmä ja mitä vaatimuksia se asettaa kilpailutasoiselle tuotteelle.</p> <p>Insinööriyössä suunniteltu ja toteutettu karaokejärjestelmä soveltuu niin ammatti- kuin viihdekäyttöönkin ja tarjoaa mahdollisuuden tuottaa eritasoisia versioita eri käyttötarkoituksiin.</p>	
Avainsanat	KaraOkay, karaoke, pilvipalvelu, Raspberry Pi

Authors Title	Tommi Keskitalo, Petteri Vermilä A web-based karaoke system
Number of Pages Date	52 pages 26 November 2015
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Specialisation option	Software Engineering
Instructor	Kimmo Sauren, Senior Lecturer
<p>The thesis covers the design and documentation of a new kind of karaoke system, which would bring the service straight into personal handheld devices of consumers.</p> <p>A collection of central features in competing solutions were determined and used to assemble a plan for a modern, user oriented system. After charting the main features the project technologies and components, such as Bootstrap, AngularJS and Symfony, were decided and a competitive product could be designed. During the research it quickly became apparent that the standard for products of this amplitude is very challenging.</p> <p>The product was initially planned as a local network service, but as development progressed it was decided to change the system into a cloud service so that it could be accessed wherever and whenever as long as an Internet connection was available. The next step was implementing a prototype based on the plans and designs with a simplified user interface. This served as a demo version and as proof of concept. The actual system development started soon after. There were a lot of varying features planned for the system of which song queue and playback of karaoke tracks among others were included in the first version and the rest were documented for further development.</p> <p>The main emphasis was on planning to achieve the best possible user experience on every supported end device, and along the process a lot of valuable know-how regarding the development of cloud services was gained. It was also realized that productisation was a big part of comprehensive product development that had to be taken into account at an early stage — such as who or what is the target audience and what their expectations and requirements are for a competitive level product.</p> <p>The karaoke system designed and implemented in the project suits professional as well as entertainment use and offers the possibility of producing a variety of versions for different purposes.</p>	
Keywords	KaraOkay, karaoke, cloud service, Raspberry Pi

Sisällys

Lyhenteet ja sanasto

1	Johdanto	1
2	KaraOkayn kilpailevat järjestelmät	5
2.1	Hinnoittelu	6
2.2	Ominaisuudet	6
2.3	Tuotevertailu	13
2.4	Kohderyhmän asettamat vaatimukset	15
2.5	KaraOkayn tarjoamat edut	15
3	KaraOkay-järjestelmän suunnittelu	16
3.1	Karaokejärjestelmän määritelmä	16
3.2	Käyttäjätasot	17
3.3	Toiminnollisuudet	17
3.4	Käyttöliittymä	20
3.5	Siirtyminen KaraOkayhyn ja kappaleiden hankinta	30
3.6	Keskitetty asiakasrekisteri ja Single Sign-on-palvelu	31
4	Asiakaspäätelaitteet	32
4.1	Vaadittavat ominaisuudet ja mahdolliset laitevaihtoehdot	32
4.2	Kokoonpanot	32
4.3	Raspberry Pi keskuslaitteena	35
5	KaraOkayssä käytetyt tekniikat	36
5.1	Verkkosovellus vai työpöytäsovellus	36
5.2	Toteutus Single-Page Appina	36
5.3	Järjestelmän vaatimukset	38
5.4	Web-sovelluksen kommunikaatorajapinnat	39
5.5	Testaus	44
5.6	Mobiilisovellus	48
6	Jatkokehitys	49
6.1	HipHop Virtual Machine	49

6.2	Tulevat ominaisuudet	50
7	Yhteenveto	51
	Lähteet	53

Lyhenteet ja sanasto

ORM	Object-relational mapping. Oliomallin mukaisen esityksen kuvaus relaatiomallin mukaiseksi esitykseksi.
MVC	Model-view-controller. Ohjelmistoarkkitehtuurityyli, jonka tarkoituksena on erottaa sovelluksen liiketoimintalogiikka ja käyttöliittymä.
Some	Sosiaalinen media, kuten Facebook, Google+, Twitter, keskustelupalstat, tai jonkin palveluntarjoajan luomia yhteisöjä omille käyttäjilleen.
On Demand	Yrityksen ylläpitämä palvelu, jonka käytöstä maksetaan yleensä kuukausittainen veloitus.
ACL	Access Control List. Työkalu käyttöoikeuksien ja niiden periytyvyyden hallintaan (1).
DOM	Dokumenttioliomalli. On dokumentin rakenteen puumallinen kuvaus, joka kuvaa elementtien keskeisiä suhteita (2).

1 Johdanto

Insinööriyön tavoitteena on suunnitella ja kehittää kuluttajahintatavallinen ammattitason karaokejärjestelmä harraste- ja ammattikäyttöön. Työn kohderyhmänä ovat ikäiset karaoken laulannasta kiinnostuneet henkilöt.

Lopputuotteen on tarkoitus olla helposti lähestyttävä ja nopeasti opittava järjestelmä, joka tarjoaa ominaisuuksia, jotka kilpailevat ammattitason järjestelmien kanssa. Tuote kulkee projektissa kehitysnimellä KaraOkay, jolla viitataan Symfony-PHP-sovelluskehityksen päälle rakennettuun palveluun. Toistaiseksi tuotteelle suunnitellaan ainoastaan web-rajapinta, mutta myöhemässä vaiheessa mahdollisesti myös natiivit mobiilisovellukset.

Järjestelmän suunnittelussa otetaan huomioon Raspberry Pin kaltaiset asiakaspäätelaitteet, joiden käyttötarkoitus kokonaisuudessa on toimia etähallintalaitteena pilvipalvelulle. Konseptin pohjalta palvelun voi tuottaa ja mahdollisesti päästä markkinoille yhdeksi toimijaksi.

Karaoken historia

Karaoke on Japanissa 1970-luvulla kehittynyt taiteenmuoto, jossa musiikkikappaleet on nauhoitettu ilman alkuperäisvokalistiä ja kuka tahansa voi tulkita oman versionsa kappaleesta (3). Sana "karaoke" muodostuu Japanin kielen sanojen "karappo" ja "okesutara" alkuosien yhdistelmästä, joka tarkoittaa vapaasti suomennettuna "ilman orkesteria". Tällä viitataan yleensä musiikkinauhoitteisiin, joissa on ainoastaan äänitetty tunnettujen kappaleiden säestys. (4.)

On yleisesti tunnistettu karaoken saaneensa alkunsa Kobe City -nimisessä baarissa Japanissa. Kiertelevä kitaristi ei päässyt esiintymään sairastumisen tai jonkin muun syyn vuoksi, ja baarin omistaja varautui iltaan nauhoittamalla etukäteen kappaleiden säestyksiä, joiden päälle vokalisti kykeni laulamaan. Vokalisti nautti uudesta tavasta, ja vaikka tämä onkin legenda, se saattaa hyvin olla alku karaokelle, joka on sen jälkeen kaupallistettu ja on hyvin suosittu Japanissa sekä myös muualla ympäri maailmaa. (4.)

Karaoken syntyessä Japanissa se peri samalla myös japanilaisen kulttuurin, jossa juhlien ilmapiiri on ollut aina iloinen ja eloisa. Karaoke usein seuraa näitä jalanjälkiä, eikä

tärkeintä ole, laulaako laulaja hyvin vai huonosti. Vaikka esiintyjä laulaisi nuotin ohi, hänen esityksensä voi olla erinomainen. Japanilaisiin tapoihin kuuluu olla antelias kuunnellessa toisia laulajia ja esimerkiksi osallistua esitykseen taputtamalla käsiä. Näin ollen, kuka vain voi helposti esiintyä toisten edessä häpeilemättä laulutaidosta riippumatta. (4.)

Mikrofoniin laulaminen säästyksen kanssa esityksen jälkeen raikuvat aplodit saavat laulajan usein tuntemaan itsensä ammattilaulajaksi. Karaoke stimuloi ihmisten halua laulaa, mikä usein koetaan virkistäväksi stressin ympäröimässä yhteisössä. (4.)

Ensimmäiset karaokelaitteet käyttivät C-kasetteja, mutta teknologian kehittyessä CD:t, VCD:t, LD:t ja lopulta DVD:t korvasivat ne. Vuonna 1985 julkaistiin ensimmäinen CD+G-formaatin CD-levy (5). CD+G on CD-formaatin jatke, joka voi ääniraitojen lisäksi sisältää matalaresoluutioista grafiikkaa, kuten kappaleen sanoituksia. Mediasoitimen on erikseen tuettava formaattia, jotta se pystyy näyttämään grafiikan.

Karaokelaitteet olivat aluksi vielä isoja, sillä kappaleiden tallennus vei paljon fyysistä tilaa, ja ne olivat usein pyörillä liikuteltavia. Niissä oli näyttö, kappaleiden valintapaneeli, mikrofonit ja kaiuttimet kaikki samassa (kuva 1), ja niitä oli tyypillisimmin pubeissa tai ravintoloissa.



Kuva 1. Vanhanajan karaokelaite (6).

Vuoden 1997 loppupuolella ja vuoden 1998 alussa kehitettiin mediaformaatti MP3+G, joka on nykyään yleisin tallennusmuoto karaokekappaleille videoformaattien ohella. MP3+G yhdistää MP3-äänenpakkausmenetelmän ja CD+G-formaatin grafiikan. Vuodesta 2003 lähtien on julkaistu useita ohjelmia, jotka tuovat karaoken PC:lle. Satojen CD+G- ja LD-levyjen mukana kantamisen sijaan karaoke DJ:t saattoivat nyt siirtää koko kokoelmansa kovalevyille ja toistaa kappaleet ja sanat tietokoneeltaan. (7.)

Kappaleiden digitalisointi mahdollisti myös karaokelaitteiden fyysisen koon pienentymisen. Nykypäivän karaokelaitteita on monenlaista, ja niitä myydään lelukaupoista vähittäistavarataloihin ja aina alan erikoisliikkeisiin asti. Nykypäivänä kuluttajalla on valittavana pienistä karaokeratkaisuista, kuten mukana kulkeva moderni karaokejärjestelmä (kuva 2), aina isoihin tietojärjestelmiin asti.



Kuva 2. Esimerkki modernista karaokejärjestelmästä älypuhelimella (8).

Karaoken pohjalta on myös julkaistu useita videopelejä. Ensimmäinen tällainen videopeli oli Karaoke Studio, joka julkaistiin vuonna 1987 Japanissa. Se antoi pelaajalle pisteitä laulun täsmällisyyden perusteella, ja yksi tunnetuimmista tämän tyyppisistä pelisarjoista on Playstationille kehitetty SingStar-tuoteperhe. SingStar-peleihin tuli mukana erillinen mikrofoni, johon pelaajien tulee laulaa.

Älylaitteiden yleistyttyä myös monet valmistajat alkoivat tuottaa erilaisia karaoke-laulu-sovelluksia, joissa oli jo valmiina kaikki karaoken laitteiston perusedellytykset. SingStar hyödyntää älylaitteita muuttamalla ne langattomiksi lisämikrofoneiksi PS3:lle ja PS4:lle omilla Android- ja iOS -sovelluksilla. (9.)

2 KaraOkayn kilpailevat järjestelmät

Karaokemarkkinoilla on Suomessa paljon kilpailua niin koti- kuin ammattijärjestelmissä. Suomessa isoimmat toimijat ovat Melplay and KSF, joista molemmat tarjoavat omia tuotepakettejaan ja myyvät karaokeversioita kappaleista. Lisäksi internetissä on monia toimijoita, joilla on myös omat kappaletarjontansa ja tuotteet, esimerkiksi www.karaokeparty.com-sivusto, johon voi rekisteröityä ja laulaa kappaleita suoratoistona selaimella. Karaokeparty-sivustolla on muutamia ilmaisia esimerkkejä, joita voi laulaa ilmaiseksi, mutta päästäkseen käsiksi koko kirjastoon täytyy ostaa täysjäsenyys. Toinen paljon suosiota saanut palvelu on Karafun, joka on tuonut oman työpöytäsovelluksensa, jonka kautta käyttäjät pääsevät Karafunin omaan kappalekirjastoon tai laulaa omia kappaleita tietokoneelta. Karafun tarjoaa ilmaiseksi lyhyen näytteen kaikista kappaleistaan, ja omia kappaleitaan voi laulaa ilmaiseksi, mutta päästäkseen laulamaan Karafunin tarjoamia kappaleita, tulee myös hankkia täysjäsenyys.

Koska karaoke on hyvin yleistä kautta maailman, monet palveluntarjoajat ovat siirtyneet mainostamaan itseään sosiaaliseen mediaan, kuten YouTube-videopalveluun, jossa he julkaisevat omia videoita ja houkuttelevat niillä lisää uusia käyttäjiä. Lisäksi sovellusvalmistajat ovat tehneet älypuhelimille omia tuotteitaan karaokesovelluksina, kuten Smule Inc:n kehittämä ”Sing! Karaoke by Smule” -sovellus, jolla voi laulaa karaokea. Sovellus arvioi laulua ja antaa pisteitä pelimäiseen tapaan. Tällä tapaa voi kilpailla ystävien kanssa, jakaa tuloksiaan tai esityksiä sosiaalisessa mediassa.

Internetissä on myös tarjolla monia keskenään kilpailevia karaokesovelluksia, jotka eivät itse tarjoa lainkaan sisältöä, vaan ainoastaan sovelluksen karaoken laulamiseen. Yritykset ovat jättäneet kappaleiden hankinnan asiakkaan omalle vastuulle ja tavoittavat näin eri kuluttajaryhmän. Pelkän sovelluksen hankinnassa on etuna verrattuna palveluihin, jotka veloittavat asiakasta tarjoamistaan kappaleista se, että monet verkkosivustot tarjoavat pääosin englanninkielisiä kappaleita tai hyvin rajatun kotimaisen valikoiman. Hankkiessaan kappaleet itse käyttäjällä on parempi kontrolli laulettavien kappaleiden määrästä ja laadusta.

2.1 Hinnoittelu

Yleisesti ottaen Suomessa myytävät karaokekappaleet ovat DVD:itä, ja valtaosa vähittäiskaupoista myy niitä. Niissä kappaleiden hinta on noin 50 sentistä euroon, riippuen kokoelmasta. Useat suosikkikappaleet saattavat kuitenkin toistua monissa eri kokoelmissa. Kappaleita voi myös ostaa verkosta yksittäin, mikä on kappalehinnalta hieman kalliimpaa, mutta se on helpoin tapa löytää yksittäisiä toivekappaleita. Ammattikäyttöön kappaleiden hinnat ovat viidestä eurosta ylöspäin.

Riippuen paljon käyttötarkoituksesta ja aktiivisuudesta monesti on demand -ratkaisut voivat tulla halvemmaksi, koska niillä saa kiinteällä kuukausihinnalla koko kappalekirjaston käyttöön ilman erillisiä lisäkustannuksia sekä automaattisesti kaikki uudet julkaisut heti käyttöön. Tällainen ”avaimet käteen” -paketti soveltuu hyvin aloittelevalle harrastelijalle, sillä se ei edellytä isoja laitehankintoja tai pitkää sitoutumista. Pitkäaikainen harrastuneisuus usein kuitenkin edellyttää järeämmän kaluston hankkimista ja omaa kappalekirjastoa, jota voi laajentaa oman mielensä mukaan.

Melplay ja KSF myyvät DVD:iden lisäksi myös karaokelaitteita, kuten muun muassa mikseriä ja mikrofoneja sekä omia karaokeratkaisujaan, jotka koostuvat tietokoneesta, esiasennetusta karaokesovelluksesta ja kappalekirjastosta. Näiden hinta on kolmestahannesta eurosta kymmeneentuhanteen euroa ja on yleensä tarkoitettu ammatin harjoittamiseen. Aluksi voi valita kappalekirjaston koon viidestäsadasta kymmeneen tuhanteen kappaleeseen asti, ja myöhemmin voi ostaa lisää kappaleita ja päivittää kirjastoa.

2.2 Ominaisuudet

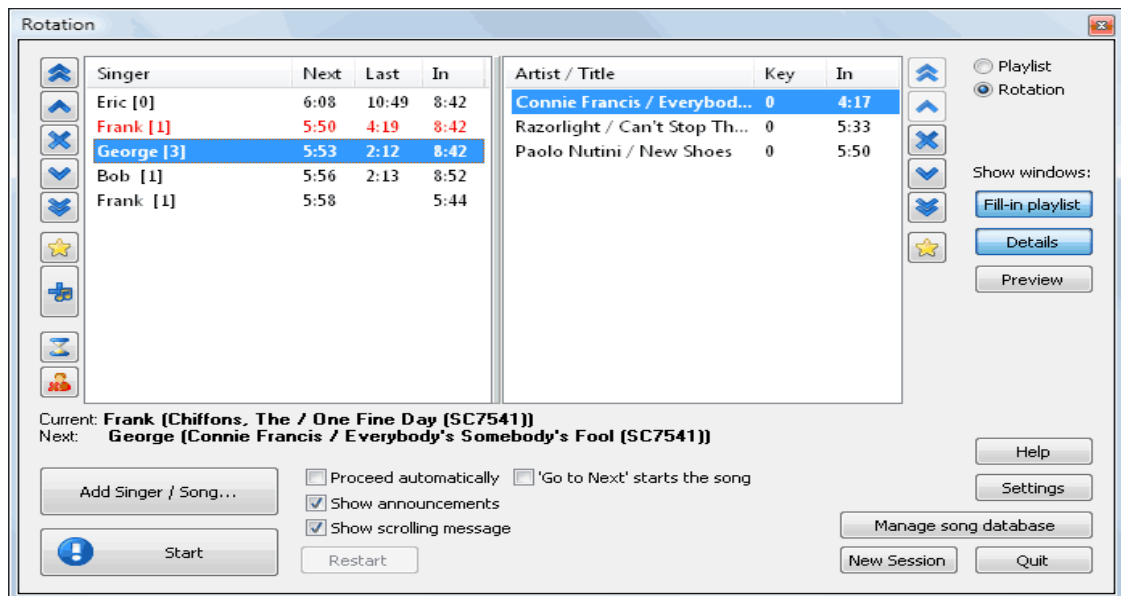
Käsiteltävillä ominaisuuksilla viitataan yksittäisiin toimintoihin, jotka voivat vaihdella eri järjestelmien välillä toteutukseltaan ja ulkoasultaan. On tärkeää huomioida vaihtoehtojen eroavaisuuksia hankinnan yhteydessä omien tarpeiden mukaan. Ominaisuuksien määrä ei välttämättä ole suoraan verrannollinen järjestelmän laatutasoon, ja tarpeettomat ominaisuudet voivat jopa osoittautua häiritseviksi hankaloittamalla sen käyttämistä. KaroOkayn suunnittelussa panostimme erityisesti ominaisuuksien yhteen liittämiseen, jolloin ne toimisivat joustavasti yhtenä kokonaisuutena ja järjestelmä näin soveltuisi moniin eri käyttömieltymyksiin.

Laulujono

Yleisesti kaikki kilpailevat tuotteet ovat keskitetysti määrätystä laitteesta hallittavia ratkaisuja ja näin ollen fyysisesti paikkaan ja valvojaan sidottuja. Ainoastaan KFS tarjoaa 12,40 euron kuukausiveloituksella lisäpalveluna järjestelmiinsä laulajille mahdollisuuden lisätä kappaleita laulujonoon etänä oman äylaitteen kautta (10). Tämä mahdollistaa tuotteelle selkeän edun muihin tarjolla oleviin vaihtoehtoihin nähden.

Tuotteet voidaankin helposti jakaa kahteen kategoriaan hallintaominaisuuksien perusteella: jonolliset ja jonottomat. Jonottomat ovat yksinkertaisia: niissä valitaan haluttu kappale ja järjestelmä alkaa toistaa kappaletta. Hyvänä esimerkkinä jonottomista on Playstation-pelikonsolille tehty SingStar-peli. Jonottomat ratkaisut sopivat hyvin harjoitteluun, yksityiskäyttöön tai pienen ryhmän illanistujaisiin kotona, kun kenenkään ei tarvitse hallinnoida usean laulajan vuoroja tai kilpailla lauluvuoroista. Kovemman käytön vaatimuksena usein onkin kappaleiden ja laulujonon hallinta. Tällöin karaokeisäntä voi kontrolloida, kuka laulaa seuraavana, ja lisätä kappaleita etukäteen listalle laulettavaksi ja järjestelmä automaattisesti ilmoittaa, kenellä on lauluvooro.

Kehittyneemmissä järjestelmissä on myös automaattinen laulajien kierron hallinta, jolloin järjestelmä valvoo laulajien laulujärjestystä. Järjestelmä pitää huolen siitä, että illan aikana kullakin laulajalla on mahdollisuus laulaa yhtä usein, ja luo kullekin laulajalle laulettavista kappaleista oman henkilökohtaisen jonon, josta laulaja voi laulaa vain yhden kappaleen kerralla (11). Tämä on usein toteutettu lisäämällä kappaleiden sijasta laulaja jonoon, kuten kuvasta 3 näkyy, ja laulajille on määritelty omat henkilökohtaiset kappalelistat.



Kuva 3. Siglos Karaoke Professional Singer rotation management (12).

Lauluvuoron jälkeen laulaja putoaa listan pohjalle. Verrattuna yksilotteiseen kappaleiden listaukseen laulajien kierrättäminen mahdollistaa usean kappaleen lisäämisen kerralla yhdelle laulajalle ja järjestelmä huolehtii automaattisesti siitä, että kukin saa laulaa omalla vuorollaan ilman, että sitä tarvitsee erikseen vaihtaa.

Laulujonon hallinta

Hyvin oleellinen osa laulujonoa on myös sen ylemmän tason hallinta ja oikeellisuuden valvonta muuttuvissa tilanteissa. Kun kyseessä ovat ihmiset ja heidän kanssaan toimiminen, ei tietokone osaa ennakoida kaikkia ja karaokeisännällä on oltava oikeus muokata laulujonoa. Karaokeisäntä voi poistaa ja lisätä varauksia, muokata esittäjiä tai vaihtaa kappaleita ennen lauluvuoroa, ohittaa lauluvuoron tai siirtää jotakin myöhemmäksi, jotta laulujono pysyy koko ajan järjestyksessä. Vaikka laulajille annetaan mahdollisuus perua oma lauluvuoronsa tai vaihtaa kappaletta ennen omaa vuoroaan, ei järjestelmä voi täysin luottaa sen käyttäjiin. Lisäksi karaokeisännän tulee käynnistää jokainen lauluvuoro ja valvoa mikrofoniin siirtymistä aina seuraavalle vuorossa olevalle.

Karaokeisännällä on myös oikeus estää laulajaa enää lisäämästä kappaleita jonoon, jos hänen käytöksensä ei ole tilanteeseen sopivaa.

Kappalelista

Jonon hallinnan lisäksi kappalelistan ylläpitäminen voi osoittautua työlääksi: Paperitulosteiden uudelleen tulostaminen tiheään tahtiin päivittyvässä kappalekirjastossa ei ole kovin luontoystävällistä. Jos kappaleita halutaan listata esittäjän ja kappaleiden mukaan aakkosjärjestyksessä ja vielä mieluiten useana kopiona, voi tulla paljon paperijätettä, kun kappaleita on satoja tai jopa tuhansia. Monesti karaokebaareissa näkee tulostettuja lehtiä kappalelistoista ja paperilappuja toiveita varten. Paperille kirjoitetaan oma nimi ja kappaletoive listalta ja toimitetaan karaokeisännälle, joka lisää laulutoiveen järjestelmän laulujonoon. Käytäntö toimii myös jonottomissa järjestelmissä, kunhan joku ottaa isännän roolin ja ylläpitää listaa manuaalisesti.

KaraOkay pohjautuu kappalelistojen hallintaan ja niiden digitalisointiin sekä tehokkaaseen hyödyntämiseen. Tarjoamalla laulajille mahdollisuus selata kappalelistaa, etsiä suosikkikappaleensa, seurata jonon etenemistä ja lisätä laulutoiveita suoraan omalta tai karaokepaikan tarjoamalta yleiseltä älylaitteeltaan kevennetään karaoke-isännän työtaakkaa ja annetaan hänelle enemmän aikaa keskittyä muihin työtehtäviin. Järjestelmä kykenee toimimaan lähes itsenäisesti, jolloin se ei välttämättä vaadi täystyöllistettyä karaokeisäntää.

Tilastokeskuksen väestön tieto- ja viestintätekniikan käyttö -tutkimuksen mukaan vuonna 2014 älypuhelinia käyttävien suomalaisten osuus nousi edellisestä vuodesta 10 prosenttiyksikköä 60 prosenttiin (13). Tämän pohjalta olisi turvallista olettaa, että suurimmalla osalla laulajista, tai jollain heidän ystävällään, olisi käytettävissä älypuhelin, jonka avulla he voivat käyttää palvelua. Lisäksi karaokepaikka voi tarjota yleisiä älylaitteita, joiden kautta kaikki voivat käyttää järjestelmää.

Välimusiikki-soittolista

Usein karaokebaareissa laulettavien kappaleiden välissä soi täytemusiikkia. Tämän voi toteuttaa monella tavalla, kuten esimerkiksi vaihtamalla äänisisäntulon stereoissa karaokejärjestelmästä radioon. Kehittyneimmissä järjestelmissä on kuitenkin usein vaihtoehtona lisätä omaa musiikkia tai streamata internetistä taustamusiikkia. Kappaleet lähtevät automaattisesti soimaan, kun karaoke-esitys päättyy, ja päättyy, kun uusi esitys alkaa.

Kappalekirjastot

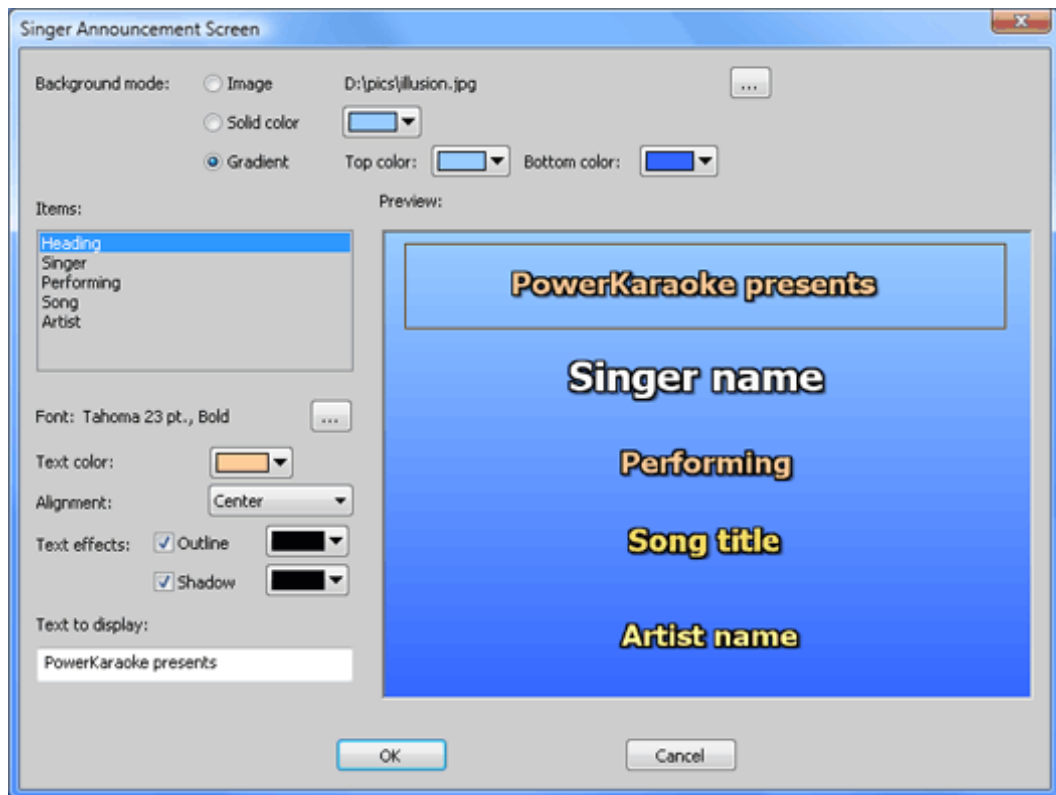
Kappalekirjastolla tarkoitetaan kussakin järjestelmässä tarjolla olevia kappaleita, joita laulajat voivat valita laulettaviksi. Kirjastoja on lähinnä kahdenlaisia tai niiden sekoitus:

- suoratoistettavia, kuten YouTube- tai Karaokeparty-sivustot, joiden tarjonta on täysin sivustojen kontrolloitavissa
- paikallisia, kuten KSF, Siglos Karaoke professional, PyKaraoke tai SingStar, joissa kappaleet tulee hankkia erikseen. KSF ja SingStar tukevat vain oman kauppansa tarjoamia kappaleita, mutta molemmissa on laaja valikoima kappaleita tarjolla, ja siitä voi räätälöidä sopivan kirjaston.

Sekoitus on yhdistelmä molempia, tästä hyvänä esimerkkinä Karafun, joka tarjoaa laajan englanninkielisen kappalekirjaston suoratoistettavana, mutta työpöytäsovellus tukee myös paikallisten tiedostojen toistamista suoraan koneelta.

Laulajien hallinta

Laulajien hallinta on tarpeellinen ominaisuus järjestelmissä, joissa on kehittynyt laulujono. Laulajien hallinta sisältää muun muassa laulajan lauluhistorian, kutsumanimen ja järjestelmästä riippuen mahdollisia lisätietoja. Siglos Karaoke professional tarjoaa käyttäjille mahdollisuuden muokata yksilöllisen ilmoitusruudun, jossa voi muun muassa vaihtaa oman kuvansa taustakuvaksi (kuva 4).



Kuva 4. Siglos Karaoke Professional Singer Announcement Screen (14).

Jos karaokejärjestelmä käyttää esittäjien kiertoa, on järjestelmään lisättävä käyttäjiä etukäteen, jotta se kykenee muodostamaan listan kaikista laulajista. Karaokeisäntä kykenee luomaan uusia käyttäjiä jonoon ja poistamaan vanhoja.

Ilmoitusnäkyä ja jonon esittäminen

Hyvin useasti karaoketilaisuuksissa kappaleiden välissä esitetään ilmoitusnäkyä. Näytöillä esitetään jonosta kolme tai viisi seuraavaa nimeä, jotta kukin esiintyjä osaa valmistautua tulevaan lauluvuoroonsa ja voi seurata jonon etenemistä, tai kuvan 4 mukainen seuraavan laulajan ilmoitus. Esitystapoja on monia, ja jossain tapauksissa myös lauluvuoron aikana ruudun ala-osassa saattavat vierähtää laulujonossa seuraavina olevat laulajat. KaraOkayn ideana on tarjota tietoa suoraan laulajan älylaitteelle sekä arvio siitä, koska hänen oma seuraava lauluvuoronsa alkaa. Ilmoitusnäky voi myös olla räikeä, tai sitä ei välttämättä ole laisinkaan jonottomissa järjestelmissä, jolloin tehtävä jää karaokeisännän vastuulle.

Korkeuden ja nopeuden muuttaminen

Koska kaikkia lauluja ei ole kirjoitettu kaikkien laulettavaksi ja ihmisten äänialat vaihtelevat, hyvä karaokejärjestelmä kykenee vaihtamaan musiikin äänialaa tai nopeutta, jotta laulettava kappale sopisi paremmin laulajan omaan laulutyyliin. Tavallisesti äänialan vaihteluväli on miinus kolmesta plus kolmeen alkuperäisestä sävellajista. DVD:llä äänialan vaihtaminen on toteutettu eri ääniraidoilla, sillä sen muokkaaminen toiston aikana ei ole kovin yleinen laitetason ominaisuus. Äänialaa voi myös muokata koneellisesti joissakin järjestelmissä.

Digitaalisen videon nopeuden muuttaminen on yleisesti ottaen harvinaista, mutta mp3-tiedostoilla se on helposti toteutettavissa. Tämän vuoksi on harvinaista nähdä suomalaisessa karaokessa nopeuden vaihtelua, ja hienosäätö pohjautuukin usein ainoastaan äänialan vaihteluun ja taajuuskorjaukseen.

Käyttäjien keskinäinen kommunikaatio

Nykypäivänä käyttäjien vuorovaikutus on iso osa markkinointia ja käytettävyyttä. Monet sovellukset tarjoavat sosiaalisen median integraatiota tai muita siihen rinnastettavia ominaisuuksia, kuten esityksen jakamista, julkaisemista tai toisten suoritusten tykkäämisestä ja kommentoimista. Lisäksi jotkut järjestelmät tallentavat esityksen, jotta sen voi uudelleen esittää tai julkaista myöhemmin.

SingStar, PartyKaraoke ja Sing! Karaoke myös arvioivat ja pisteyttävät laulusuorituksen tarkkuuden perusteella, kuinka hyvin laulaja osuu nuotteihin. Tämä muodostaa luonnollisen kilpailuympäristön, jossa käyttäjät koettavat päihittää toisensa ja esimerkiksi PartyKaraoke julkaisee sivuillaan parhaiten menestyneet laulajat, mikä motivoi käyttäjiä kehittämään itseään. Palvelut ovat kuitenkin saaneet jonkin verran kritiikkiä kyvyttömyydestä erotella huonot laulajat oikeasti hyvistä. (15.)

Sivustoille ja älylaitesovelluksille on hyvin yleistä luoda pelihenkinen kokemus, joissa laulua arvioidaan koneellisesti ja se voidaan jakaa yhteisössä, missä muut jäsenet voivat kommentoida ja arvioida toisten suorituksia.

Liikuteltavuus

Sing! Karaoke on onnistunut hyvin toteuttamaan älylaitteille karaokesovelluksen, missä voi laulaa lempikappaleitaan missä tahansa. Vastapuolella ovat järeät pöytätietokone-ratkaisut, joiden siirtäminen on työlästä ja hidasta. Sing! Karaoke ei välttämättä sovellu hyvin karaokeiltojen viihdykkeeksi, mutta esimerkiksi Playstationin SingStar on suhteellisen helposti siirrettävä järjestelmä, joka saattaa kulkea hyvin ystävän luokse tai muulle juhlapaikalle.

Rekisteröinti

Monet verkossa olevat palvelut edellyttävät käyttäjää kirjautumaan tai pyrkivät jollain muulla tapaa yksilöimään käyttäjät. Sivustot usein sallivat rajatun käytön anonyymeille, jotta he pääsevät tutustumaan järjestelmään, mutta esimerkiksi YouTubesta voi kuka tahansa laulaa kappaleita ilman rajoituksia. Maksullisissa sovelluksissa on oletettavasti oltava käyttäjien todennus ja turvalliset maksuominaisuudet.

Järjestelmän on kuitenkin hyvä tukea tuntemattomia käyttäjiä, koska rekisteröinti lisää aina käyttöönottokynnystä. Verkkoselaimilla tämä voi kuitenkin osoittautua isoksi haavoittuvuudeksi, koska ilkeiden aiheuttajia on aina internetissä etsimässä seuraavaa kohdetta. Suljetuissa järjestelmissä ongelma on suhteessa paljon pienempi, sillä ongelmatekijät on rajoitettu fyysiseen sijaintiin.

2.3 Tuotevertailu

Kilpailevien järjestelmien ominaisuuksia selvitellessä teimme vertailua paremmin havainnoivan taulukon sovellusten ominaisuuksista (taulukko 1). Tavoitteena oli saada mahdollisimman laaja otanta eri alustoille ja kohderyhmille toteutetuista mobiili-, selain- ja työpöytäsovelluksista.

Taulukko 1. Kilpailevien järjestelmien ominaisuudet (9; 10; 12; 16; 17; 18; 19; 20).

	KSF Player	KaraokeParty	Sing! Karaoke by Smule	Karafun	Siglos Karaoke professional	PyKaraoke	Youtube	SingStar	KaraOkay
Kohderyhmä	Ammatti	Koti	Koti	Koti	Ammatti Koti	Koti	Koti	Koti	Ammatti Koti
Alusta	-	Flash	Android IOS	Windows	Windows	Linux Windows	Flash	PS 2 / 3 / 4	Selain
Esittäjien jonon hallinta	on	ei	ei	on	on	ei	ei	ei	on
Soittolistat	ei	ei	ei	ei	ei	on	on	ei	ei
Kappalelista	on	on	on	on	on	on	ei	on	on
Välimusiikki-soittolista	on	on	ei	ei	on	ei	ei	ei	on
Kappalekauppa	on	ei	on	ei	ei	ei	ei	on	on
Kappalevalikoima	n. 10 000	1800+	700+	20 000+	ei	ei	Ei tietoa	Ei tietoa	Ei tietoa
Laulajaprofiilit	ei	on	on	ei	on	ei	on	on	ei
Ilmoitusnäkyvä	on	ei	ei	on	on	ei	ei	ei	on
Kappaleen sävelen muuttaminen	on	on	on	osittain	osittain	ei	ei	on	ei
Kappaleen nopeuden muuttaminen	ei	ei	ei	ei	osittain	ei	on	ei	ei
Yhteisöt	ei	on	on	ei	ei	ei	ei	on	on
Rekisteröinti	ei	on	on	ei	ei	ei	on	on	on
Ilmainen	ei	osittain	osittain	osittain	kokeilu	on	on	ei	ei
Älypuhelin yhteensopiva	osittain	ei	on	ei	ei	ei	on	osittain	on
Yhtäaikaista käyttäjiä	1+	1	1	1-2	1-2	1	1	2	1+

Erikoisin tuotteista on varmaankin Youtube-videopalvelu, josta löytyy erilaisia karaoke-versioita kappaleista. Tämä ei tee siitä suoraan karaokesovellusta, mutta mahdollistaa silti laulannan, ja esimerkiksi Siglos tukee Youtubea vaihtoehtoisena videolähteenä. Tämän vuoksi koimme sen olevan looginen osa vertailua.

Merkittävimmäksi eroiksi osoittautuvat kappaleiden hankintaan vaikuttavat tekijät, kuten kappalekauppa ja -valikoima ja niiden hallinta. Isoimmat tekijät, kuten SingStar ja KSF Player ovat tuoneet jo henkilökohtaiset älylaitteet osaksi laulamista, vaikka niiden lähestymismalli on täysin eri. SingStar muuttaa älypuhelimien mikrofoniksi, ja KSF Player tarjoaa lisäominaisuutena mahdollisuuden ilmoittautua laulajaksi omalta älylaitteelta.

Toinen huomattava ero on yhtäaikaisten käyttäjien määrä eli se, kuinka moni voi samaan aikaan käyttää järjestelmää, laulaa tai isännöidä tapahtumaa. Useimmat tukevat vain yhtä laulajaa kerralla, kuten Partykaraoke, PyKaraoke tai KSF Player, vaikka järjestelmään voi lisätä useamman mikrofonin ja jakaa lauluvuoron. SingStar tukee itsestään duettoja. Muutamat järjestelmät tukevat samanaikaista monikäyttöä, mutta edellyttävät tällöin kahta monitoria – tällöin toisessa esitetään sanat ja toisessa voidaan hallinnoida jonoa.

2.4 Kohderyhmän asettamat vaatimukset

KaraOkayn tuli olla kilpailukykyinen muiden laitteiden ja järjestelmien rinnalla. Tästä syystä sen tuli myös kyetä vastaamaan kohderyhmän asettamia vaatimuksia. Järjestelmien vertailussa tuli huomioida myös laitteistoon vaikuttavat tekijät, kuten mikrofonien lukumäärä, näyttöliitännät, käytettävyys ja laajennettavuus. Työn tavoitteena oli rakentaa järjestelmä, joka toimisi kaikilla yleisimmillä käyttöjärjestelmillä. Tämän vuoksi KaraOkayn tuotekehitys kohdistui sivuston suunnitteluun pilvipalveluna silmälläpitäen laitekohtaisia rajoituksia. KaraOkay suunniteltiin pilvipalveluna, minkä vuoksi sitä voi käyttää kaikilla päätelaitteilla, joissa on internetyhteys.

KaraOkayta kehittäessä panostettiin monipuoliseen sovelluskokonaisuuteen, joka kykenee tarjoamaan kaikki tavanomaiset karaokejärjestelmien ominaisuudet kilpailukykyiseen hintaan. Esimerkiksi tavallisen kannettavan tietokoneen saa laajennettua keskiveroksi karaokeviihdelaiteeksi liittämällä siihen mikrofonin.

2.5 KaraOkayn tarjoamat edut

Isoimpana etuna kilpailijoihinsa nähden KaraOkaylla on internetyhteyden hyödyntäminen hallinnoinnissa, ja laulajien on mahdollista kirjautua palveluun langattomasti ja sen kautta hallinnoida omia tietojaan järjestelmässä. Lisäksi järjestelmä on tavallista ympäristöystävällisempi, sillä se ei edellytä paperilehtiöiden tulostusta ja rohkaisee kaikkia käyttäjiä siirtymään käyttämään digitalisoitua kappalehakua ja lauluvuoroon ilmoittautumista.

Pilvipalveluna KaraOkay on myös laiteriippumaton: sitä voi käyttää missä tahansa jo olemassa olevassa järjestelmässä, eikä se näin vaadi isoja laiteinvestointeja, kunhan päätelaitteessa on internetyhteys. Järjestelmä tarjoaa myös paljon uusia mahdollisuuksia, joita muissa kilpailevissa järjestelmissä ei ole aikaisemmin ollut, kuten oman lauluvuoronsa siirtämisen myöhemmäksi, suosikkikappaleiden hallinnan, tehokkaamman jonon seuraamisen reaaliajassa sekä ilmoitukset omista tapahtumista omaan henkilökohtaiseen äylaitteeseen.

Koska järjestelmä ei ole laitesidonnainen, sen käyttöönottokynnys on hyvin alhainen ja se soveltuu hyvin kaikentasoisille käyttäjille harrasteliijoista ammattilaisiin. Järjestelmän käyttöönoton jälkeen laitteistoa on helppo päivittää, sillä sovellus toimii kaikilla alustoilla.

Useissa kilpailevissa järjestelmissä on karaokeisännän manuaalisesti hallinnoima laulujono. KaraOkay sen sijaan ylläpitää laulujonoa automaattisesti laulajien lisäämien toiveiden pohjalta. Lisäksi KaraOkayssa kappalekirjasto on digitalisoitu, ja käyttäjät voivat selata sitä omalla äylaitteellaan ja merkitä lempikappaleensa suosikeiksi sekä tarkkailla oman lauluvuoronsa lähestymistä. Kilpailevissa järjestelmissä usein kappalekirjastot ovat joko CD- tai DVD-levyillä tai digitalisoituna. Lisäksi baareissa kappalelistat esitetään laulajille paperilehtiönä, joista he voivat hakea laulettavien kappaleiden koodit ja esittää ne karaokeisännälle, joka lisää laulajan ja kappaleen jonoon. KaraOkay poistaa tarvittaessa välikäden yksinkertaistamalla prosessia ja tarjoamalla laulajalle mahdollisuuden omalla lauluvuorollaan aloittaa esityksen omasta äylaitteestaan.

3 KaraOkay-järjestelmän suunnittelu

KaraOkayn suunnittelussa kartoitettiin eri kilpailevien järjestelmien ominaisuuksia ja tämän hahmottelemalla pohjalta alustavia käyttöliittymäkuvia, joita myöhemmin käytettiin pohjana tärkeimpien ominaisuuksien sijoittelussa ja ulkoasussa. Kuvat ja käyttäjätarinat tarjosivat pohjan sille, miten ominaisuudet rakentavat miellyttävän kokonaisuuden ja mitkä tiedot ovat keskeisiä millekin näkymälle ja roolille.

3.1 Karaokejärjestelmän määritelmä

Jotta järjestelmä voidaan luokitella karaokejärjestelmäksi, siinä tulee olla karaoketiedostojen toisto-ominaisuus sekä mahdollisuus valita halutut kappaleet. Yksinkertaisimmillaan karaokejärjestelmään kuuluu tavanomainen DVD-toistin ja äänimikserin kautta lisätty äänentoistojärjestelmä ja mikrofoni. Kehittyneemmissä järjestelmissä kappaleet on digitalisoitu keskitetysti hallittavaan tietojärjestelmään, josta kappaleet voidaan toistaa suoraan. Kaikissa järjestelmissä yhteistä on laulajan äänen yhdistäminen instrumentaaliseen versioon tunnetuista kappaleista sekä näytölle ilmestyvät kappaleiden sanat. Laulajan tulee siis pystyä valitsemaan haluamansa kappale tulkittavissa olevista karaoke-versioista.

3.2 Käyttäjätasot

KaraOkayssa on kolme käyttäjätasoa tai roolia: anonyymi, laulaja ja valvoja, joista kaksi jälkimmäistä vaativat rekisteröitymisen tunnistautumiseen. ACL (Access control list) valvoo kunkin käyttäjän pääsyä kuhunkin näkymään käyttäjätason mukaan. Ylemmän tason käyttäjät perivät alemman tason oikeudet ja antavat näin kattavamman pääsyn järjestelmään. Lisäksi eri käyttäjätasoilla näkymät saattavat poiketa toisistaan, esimerkiksi valvojalla saattaa olla näkymässä useita toimintoja käytettävissään.

3.3 Toiminnollisuudet

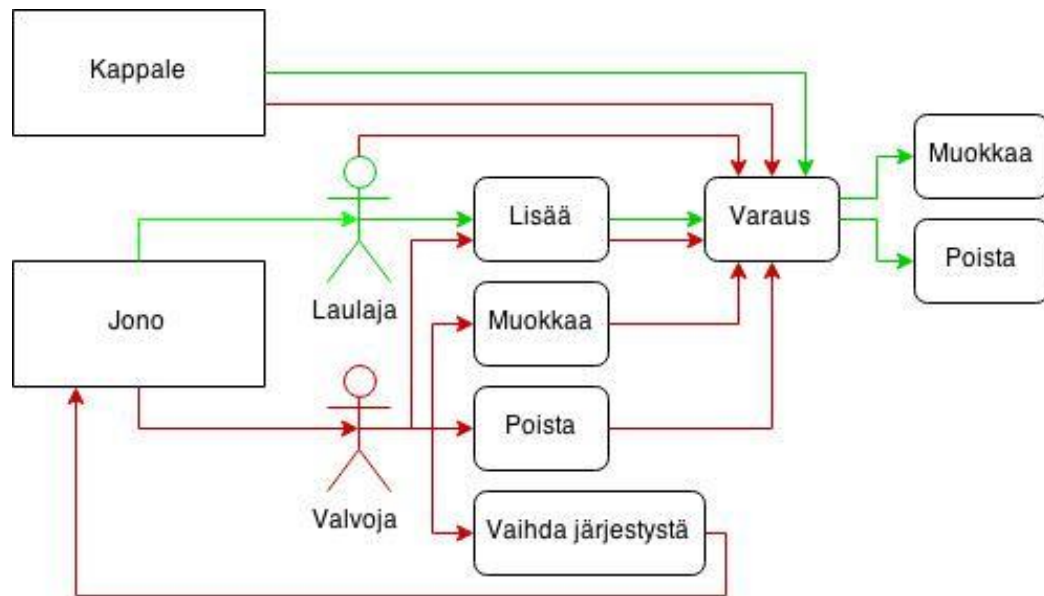
Järjestelmän toiminnallisuudet ovat sille ominainen osa, jotka määrittelevät laajuuden ja käytettävyyden pohjalta olemuksen, joka erottaa sen kilpailijoista. Käyttäjä yleensä haluaa sovelluksen ratkaisevan tietyn ongelman tai yksinkertaistavan monimutkaisen prosessin helposti ymmärrettävään konseptiin.

KaraOkayn suunnittelussa panostettiin käytettävyyteen enemmän kuin toiminnallisuuksien määrään, ja jokainen toiminnallisuus suunniteltiin edesauttamaan käyttökokemusta.

Laulujono

Lauluvuorot perustuvat yksinkertaiseen jonoon, jossa uudet varaukset lisätään pohjalle ja otetaan pois päältä. Laulajalla (vihreä tikku-ukko, kuva 5) on oikeus lisätä varaus jonoon, jossa hän siis itse on laulajana, ja muokata omia varauksiaan. Valvoja (punainen tikku-ukko, kuva 5) voi lisätä, muokata ja poistaa varauksia, mutta voi myös valita itsensä sijasta jonkun toisen laulajaksi. Valvojalla on myös oikeus muokata jonossa olevien varusten järjestystä, kuten siirtää saman esiintyjän lauluvuoroja myöhemmäksi tai siirtää pohjalta uusia esiintyjä ylemmäs. Varausta tehtäessä varaus tarvitsee tiedon laulajasta

ja kappaleesta. Oletuksena järjestelmä lisää laulajaksi tekijän ja valvojalla on myös oikeus vaihtaa laulajaa tarvittaessa.



Kuva 5. Jonon käsittely KaraOkayssa.

Laulajien priorisointi

Jokaisella laitteella on oma yksilöllinen tunniste ja yksilöllinen laulujono. Jos useamalta laitteelta lisätään kappaleita, kaikki tulevat saman tunnisteeseen jonoon, vaikka laulajia olisi oikeasti useampia. Laulujono hakee kullakin laitteelta yhden biisin ja rakentaa jonon niin, että yhden kierron aikana, kukin aktiivinen laite saa yhden lauluvuoron.

Kappalekirjasto

KaraOkay-yksiköillä on oma kappalekirjasto, joka muodostuu pilven tarjoamista kappaleista ja ravintolan omista kappaleista.

Kappalehaku

Käyttäjä voi hakea kappaleita kunkin yksikön, baarin tai ravintolan, kappalekirjastosta käyttöliittymän etsi-näkymän kautta. Hakukriteereinä voi olla joko kappaleen alkuperäinen esittäjä tai kappaleen nimi. Myöhemmässä versiossa kappaleita voi myös hakea hyräilemällä tai laulamalla älylaitteelle etsimäänsä kappaletta, ja järjestelmä tunnistaa

näytteiden perusteella halutun kappaleen tai ehdottaa vaihtoehtoja. Kappalehaun tavoitteena on esitellä laulajalle käytettävissä olevia kappaleita, joista hän voi esittää omia versioita.

Kappaleiden lisäys jonoon onnistuu helpoiten haun kautta ja tarjoaa hyvän kuvan laulajalle siitä, mitä kappaleita on tarjolla.

Työpöytä

Karaokeisännällä on käytössään työpöytä, joka muistuttaa paljon julkisesti näkyvillä olevaa inforuutua. Työpöydällä karaokeisännälle esitetään keskeisimmät tiedot järjestelmän tilasta, kuten

- viisi seuraavaa esiintyjää ja heidän valitsemansa kappaleet
- jonon pituus, niin ajallisesti kuin määrällisesti
- nyt vuorossa olevan laulajan nimimerkki ja kappaleen tiedot
- tapahtumasyöte
- pikanäppäimet.

Karaokeisälle tarjotaan myös pikatoiminnot työpöydällä, mistä hän voi esimerkiksi siirtää myöhemmäksi jotain laulajaa.

Inforuutu

Inforuutu on näkymä, joka esitetään karaokeruudulla lauluvuorojen välissä. Näkymä tarjoaa yleistä informaatiota, kuten KaraOkayn isännän työpöydällä pelkistettynä. Inforuudulla on myös vaihtoehtoista tietoa, ja se on yksilöitävissä kunkin asiakkaan mieleiseksi. Asiakas voi muokata vapaasti mallia, kuten vaihtaa taustakuvan, lisätä omaa tekstiä tai muokata tekstien värejä. Näkymässä esitetään jonon kolme seuraavaa esiintyjää sekä tietoa jonon pituudesta. Näytöllä on myös yksilöllinen QR-koodi, joka toimii linkkinä palvelun sivuille.

Oman lauluvuoron aloittaminen

Yhtenä erikoisominaisuutena KaraOkay tarjoaa esiintyjälle mahdollisuuden aloittaa oma lauluvuoronsa. Tämä vähentää karaokeisännän työtaakkaa, sillä oman lauluvuoronsa saatuaan esiintyjä voi hakea mikrofonin, ja kun hän on valmis aloittamaan, hän voi käynnistää kappaleen toiston omasta älylaitteestaan. Kun oma lauluvuoro on aktiivinen, näytölle ilmestyy painike, joka aloittaa esityksen.

3.4 Käyttöliittymä

Käyttöliittymä on laitteen tai ohjelmiston osa, jonka kautta käyttäjä käyttää tuotetta. KaraOkayn käyttöliittymä on toteutettu verkkosivustona, jonka käyttöön tarvitaan verkkoseläin. Sivuston käyttöliittymä ja näkymät toteutettiin Bootstrapillä, joka on avoimen lähdekoodin kehys, joka tarjoaa laajat responsiiviset, eli mobiilikäyttöön skaalautuvat, CSS-, HTML- ja JS-komponenttikirjastot. Koska Bootstrap perustuu avoimeen lähdekoodiin, voimme muokata sen tarjoamia valmiita komponentteja soveltumaan ratkaisuihimme.

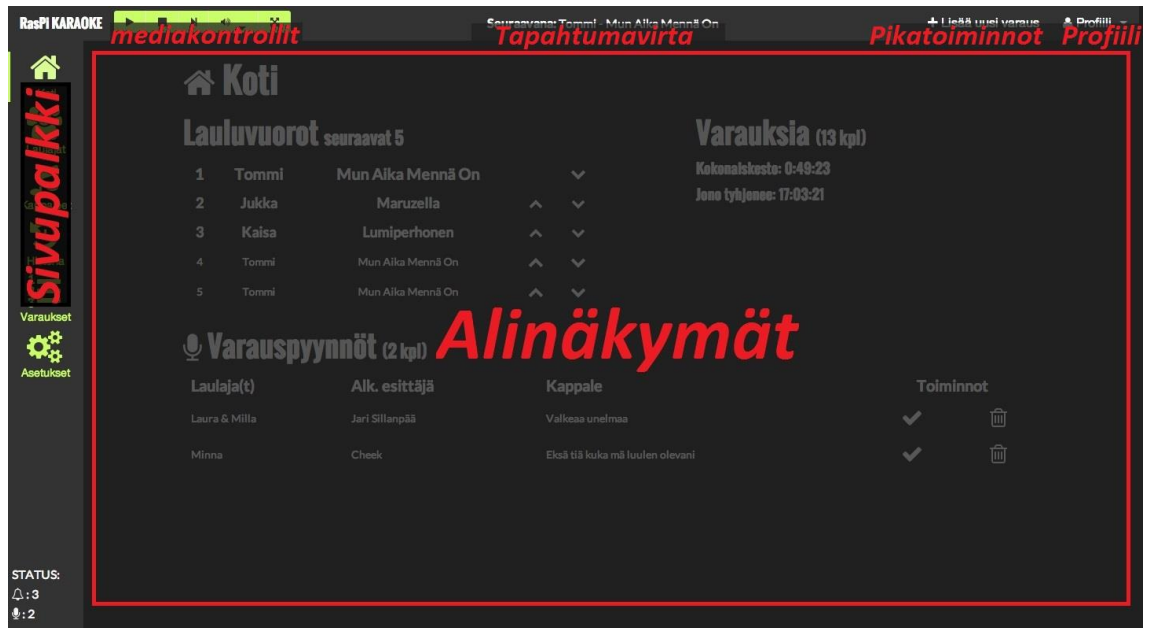
Alustavat käyttöliittymäkuvat

Suunnittelun alussa toteutettiin alustavia suunnitelma siitä, miltä sivusto tulee näyttämään. Näkymissä ei ole paljoa toiminnollisuutta, sillä niiden tehtävä on vain havainnollistaa sitä, millainen lopputuote tulee olemaan.

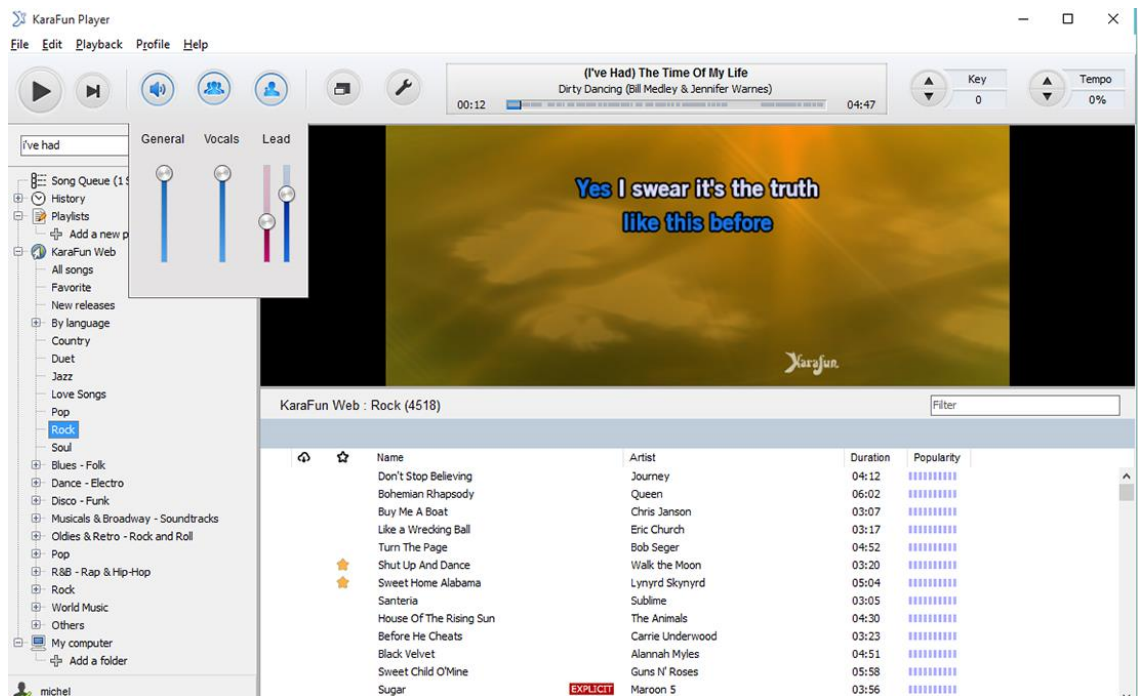
Aluksi sivustosta tehty suunnitelma sisälsi vain paikkamerkintöjä, minkä jälkeen voitiin toteuttaa raaka versio käyttöliittymäkomponenttien ulkoasusta. Suunnittelussa suurimman prioriteetin saivat näkymät, joissa käyttäjä tulisi oletetusti viettämään eniten aikaa.

Valvojan käyttöliittymärunko

Valvojan käyttöliittymä oli aluksi suunniteltu keskikoisille päätelaitteille näkymissä esitetyn suuren tiedon määrän vuoksi (kuva 6). Käyttöliittymärunko mallintaa paljon Karafun Playeriä sijoittelussa (kuva 7), paitsi videon toisto on eri näytöllä. 11-tuumaiselle tai isomalle näytölle valvoja saa kaikki tiedot yhtäaikaisesti näkyviin, mikä helpottaa sovelluksen käyttöä. Sisältö skaalautuu myös pienemmille näytöille, mutta tällöin valvoja joutuu vierittämään sivua tai avaamaan valikon nähdäkseen osan sisällöstä.



Kuva 6. KaraOkayn valvojan käyttöliittymärungon elementit.



Kuva 7. Karafun työpöytäsovellus (19).

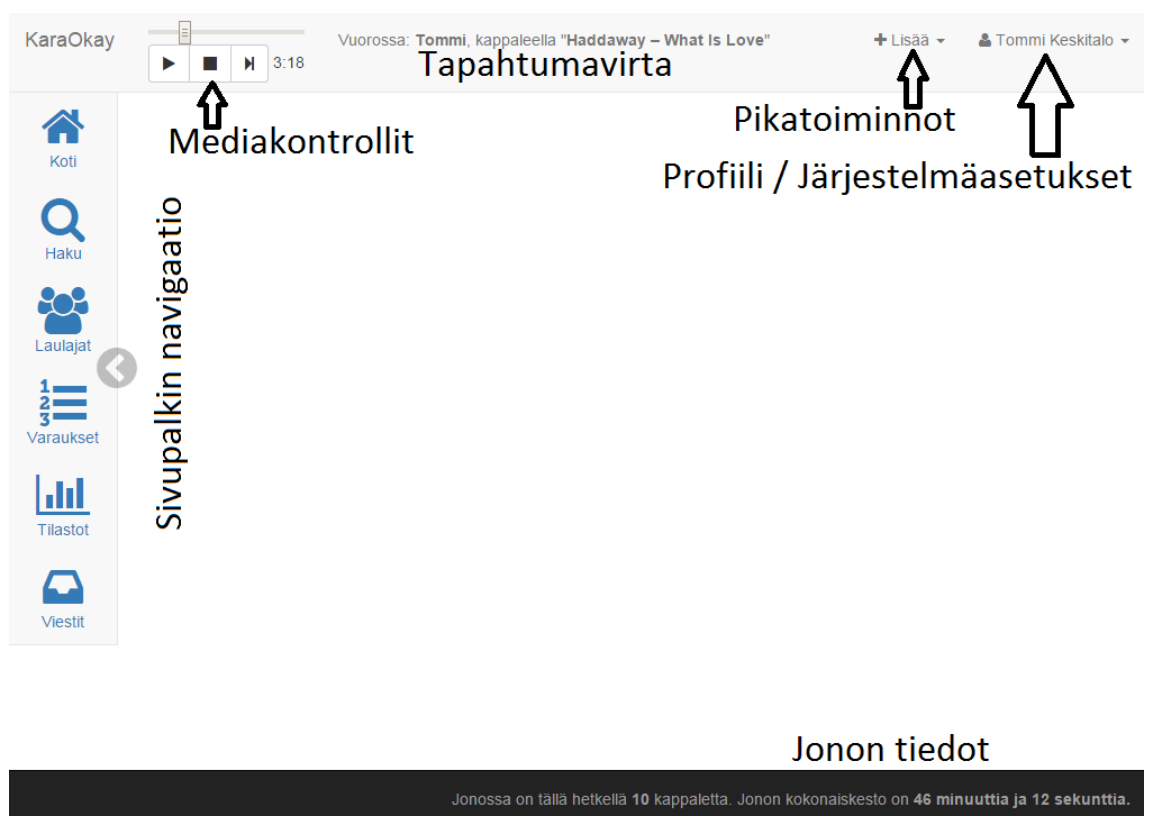
Valvojan käyttöliittymärunko (kuva 6) koostuu kolmesta pääkomponentista:

- Oikealla on sivupalkki, joka sisältää navigointipalkin. Navigaatiopalkissa on kuvattu toiminnollisuudet kuvakkeilla kategorioiden mukaan. Esimerkiksi Laulajainäkymässä ovat laulajiin kohdistuvat toiminnollisuudet.

- Yläpalkissa ovat mediakontrollit, keskitettynä tapahtumavirta, jossa esitetään jonon senhetkinen tila, sekä pikänäppäimet yleisimmille toiminnoille. Yläpalkissa on myös profiilivalikko, josta pääsee muun muassa käyttäjän asetuksiin.
- Näkymät esitetään päänäkymän paikkamerkinnässä alanäkymänä.

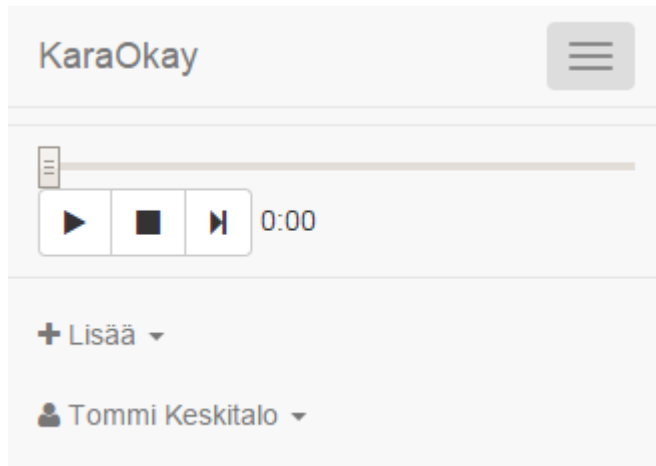
Ylä- ja sivupalkki ovat siis kiinteitä, eivätkä ne liiku vieritettäessä muun sivuston mukana, eikä niitä ladata uudestaan sovelluksen tilaa vaihdettaessa. Jos näkymässä on paljon sisältöä, vieritys keskittyy vain alanäkymään.

Prototyypissä teema on muutettu vaaleaksi ja mediapainikkeiden määrää on vähennetty yhdistämällä niiden toiminnollisuuksia. Samoin äänen voimakkuuden säätö poistettiin käyttöliittymästä ja jätettiin laitteiston hoidettavaksi. Lisäksi sivupalkki on muutettu piilouttavaksi, jotta sivusto skaalautuu paremmin mobiililaitteille. Isoilla näytöillä käyttöliittymä ei muuten eroa paljoa alkuperäisestä (kuva 8).



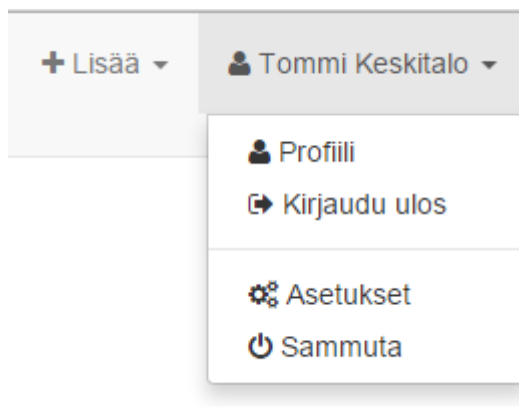
Kuva 8. Valvojan työpöytänäkö KaraOkayn prototyypissä.

Mobiililaitteilla kuitenkin jouduttiin karsimaan ominaisuuksia: esimerkiksi yläpalkin tapahtumavirta piilotetaan tilanpuutteen vuoksi ja yläpalkin kohdat muutetaan pudotusvalikoksi (kuva 9). Yläpalkissa on edelleen mediakontrollit, pikatoiminnot ja profiilivalikko, mutta ne aukeavat pienellä näytöllä painikkeen alta.



Kuva 9. Supistettu valvojan yläpalkki KaraOkayn mobiilinäkymässä.

Lisäksi varaukset-osio siirrettiin valvojan työpöydältä alapalkkiin käyttöliittymärunkoon, joka on korostettu mustalla taustalla (kuva 8). Myös asetukset siirrettiin yläpalkkiin profiilivalikon alle (kuva 10) sammuta-painikkeen yläpuolelle. Laitekohtaiset toiminnollisuudet ovat eroteltuna erottimella. Tämä vähentää sivupalkin korkeutta ja jättää sivupalkin ainoastaan karaokeen liittyvien toimintojen navigointiin.



Kuva 10. Profiilivalikko KaraOkayn prototyypissä.

Valvojan työpöytänäkymä

Valvojan työpöytä on yksi sovelluksen kriittisimmistä näkymistä, sillä se sisältää yhteenvedon kaikesta, mitä sivustolla tapahtuu. Näkymässä on paljon erilaista tietoa senhetkisestä tilasta, ja se on reaaliajassa päivittyvä. Työpöydän avulla valvoja saa helposti hahmotettavan kuvan sovelluksen senhetkisestä tilasta, kuten laulujonon seuraavat vuorot, avoimet varauspyynnöt, ilmoitukset, varauksien lukumäärän, laulujonon kokonaiskeston ja sen, kuinka paljon on vielä vapaata lauluaikaa (kuva 11).

Koti

Lauluvuorot seuraavat 5

Nro	Nimi	Mun Aika Mennä On	↑	↓
1	Tommi	Mun Aika Mennä On		↓
2	Jukka	Maruzella	↑	↓
3	Kaisa	Lumiperhonen	↑	↓
4	Tommi	Mun Aika Mennä On	↑	↓
5	Tommi	Mun Aika Mennä On	↑	↓

Varauksia (13 kpl)

Kokonaiskesto: 0:49:23
 Jono tyhjenee: 17:03:21
 Lauluaikaa jäljellä: 5h 56m 39s (23:00)

Varauspyynnöt (2 kpl)

Laulaja(t)	Alk. esittäjä	Kappale	Toiminnot
Laura & Milla	Jari Sillanpää	Valkeaa unelmaa	✓
Minna	Cheek	Eksä tiä kuka mä luulen olevani	✓

Ilmoitukset (3 kpl)

- 16:05: Laura & Milla pysyvät lauluvuoroo
- 16:02: Minna pysyvät lauluvuoroo
- 15:58: Marko perui oman lauluvuoronsa

Kuva 11. Valvojan työpöytänäkymä KaraOkayn käyttöliittymäsuunnitelmassa.

Työpöytänäkymä ei kuitenkaan ole vain tarkastelua varten, sillä valvoja voi myös hyväksyä varauspyyntöjä työpöytänäkymästä tai muokata jonon järjestystä. Toiminnoille on myös omat näkymänsä, joissa on enemmän toiminnollisuutta ja tarkempaa informaatiota, mutta työpöydässä on tarkoitus kyetä nopeasti hallinnoimaan perusominaisuuksia. Näin peruskäyttöön riittää pelkästään työpöydän tunteminen, eikä näkymien välillä tarvitse jatkuvasti hyppiä.

Valvojan kappalekirjastonäkymä

Valvojalla on työpöydän lisäksi myös muita näkymiä, jotka sisältävät eri sovelluksen osakokonaisuuksia, kuten esimerkiksi Kirjasto-näkymä, jonka kautta valvoja voi selata kappalekirjastoa, hakea kappaleita ja lisätä kappaleita laulajille jonoon tai muokata niiden tietoja (kuva 12).

Esittäjä	Kappale	Kesto	Hakemistopolku	Toiminnot
Juha Tapio	TSNEH	3:11	D:\Videot\kara files	+ ✎ i
Samuli Putro	Anna nyt	3:52	D:\Videot\kara files	+ ✎ i
Haloo Helsinki	Huuda!	4:05	D:\Videot\kara files	+ ✎ i
Cheek	Kuka muu muka	3:50	D:\Videot\kara files	+ ✎ i
Nopsajalka	Lupa olla	4:30	D:\Videot\kara files	+ ✎ i
Jenni Vartiainen	Nettiin	3:41	D:\Videot\kara files	+ ✎ i
Juju	Onnelliseksi	3:36	D:\Videot\kara files	+ ✎ i
Unikki	Pojat on poikki	4:13	D:\Videot\kara files	+ ✎ i
Samuli edelmann	Se viimeinen	3:53	D:\Videot\kara files	+ ✎ i

Osumia yhteensä 9 kappaletta, kokonais kesto: 00:34:51

Kuva 12. Kirjasto-näkymä KaraOkayn käyttöliittymäsuunnitelmassa.

Näkymässä on hakulomake kappaleen esittäjän tai kappaleen nimen mukaan ja taulukko tuloksista (kuva 12). Oletuksena haku on tyhjä, ja käyttäjälle listataan kaikki kappaleet. Taulukon alla on sivutus ja tuloksien tiedot, kuten osumien määrä ja kokonaiskesto.

Varauksien hallintanäkymä

Toinen valvojalle tärkeä näkymä on varauksien hallinta, josta hän näkee koko varausjonon ja käsittelemättömät varauspyynnöt. Käsittelemättömät varauspyynnöt ovat tunnistautumattomien käyttäjien lähettämiä lauluvuoropyyntöjä, jotka valvojan tulee hyväksyä, jotta ne lisätään jonoon. Valvoja voi joko hyväksyä tai hylätä ne ja saada lisätietoa varauspyynnön lähittäneestä laitteesta (kuva 13).

Varaukset

Käsittelemättömät varauspyynnöt

Laulaja(t)	Alk. esittäjä	Kappale	Toiminnot		
Laura & Milla	Jari Sillanpää	Valkeaa unelmaa	✓	?	✕
Minna	Cheek	Eksä tiä kuka mä luulen olevani	✓	?	✕

« 1 2 3 4 ... 12 13 »

Varaukset

Laulaja Hae Alk.esittäjä Hae Kappale Hae

#	Laulaja	Alk. esittäjä	Kappale	Kesto	Toiminnot
1	Tommi	Esittäjä	Mun Aika Mennä On	123	↓ ↗ 🗑
2	Jukka	Esittäjä	Maruzella	123	↑ ↓ ↗ 🗑
3	Kaisa	Esittäjä	Lumiperhonen	123	↑ ↓ ↗ 🗑
4	Tommi	Esittäjä	Mun Aika Mennä On	123	↑ ↓ ↗ 🗑
5	Tommi	Esittäjä	Mun Aika Mennä On	123	↑ ↓ ↗ 🗑
6	Tommi	Esittäjä	Mun Aika Mennä On	123	↑ ↓ ↗ 🗑
7	Tommi	Esittäjä	Mun Aika Mennä On	123	↑ ↓ ↗ 🗑
8	Tommi	Esittäjä	Mun Aika Mennä On	123	↑ ↓ ↗ 🗑
9	Tommi	Esittäjä	Mun Aika Mennä On	123	↑ ↓ ↗ 🗑
10	Tommi	Esittäjä	Mun Aika Mennä On	123	↑ ↓ ↗ 🗑

« 1 2 3 4 ... 12 13 »

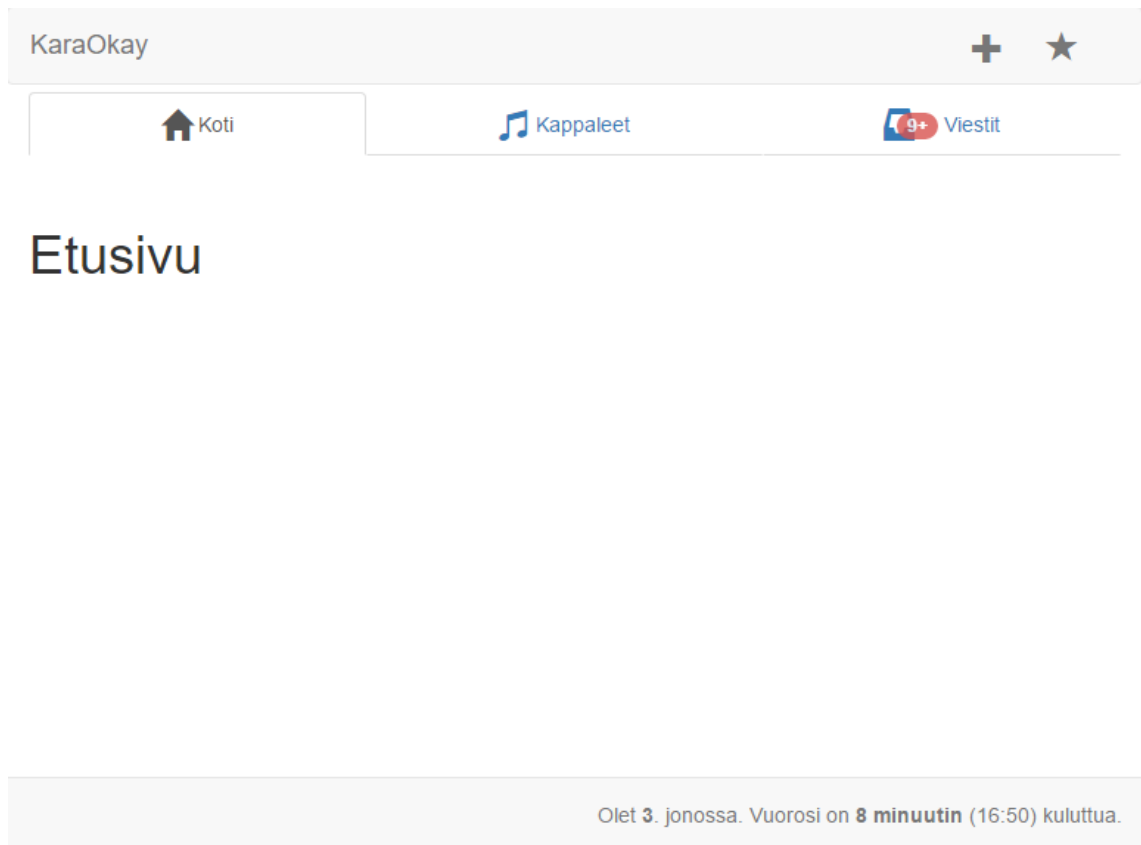
Kuva 13. Varaukset-näkymä KaraOkayn käyttöliittymäsuunnitelmassa.

Valvoja voi myös vaihtaa laulujonon järjestystä, muokata jonon varauksia tai poistaa varauksia jonosta. Varaukset on oletuksena järjestetty lauluvuoron mukaan, mutta tuloksia voi myös suodattaa laulajan, esittäjän tai kappaleen mukaan ja järjestystä voi vaihtaa (kuva 13).

Laulajan mobiilikäyttöliittymä

Laulajan mobiilikäyttöliittymästä haluttiin mahdollisimman yksinkertainen. Käyttöliittymäsuunnittelu aloitettiin miettimällä, mitä ominaisuuksia laulaja tarvitsee eniten. Mietimme, mitkä ovat laulajan keskeiset eli eniten käyttämät ominaisuudet, joita tulisi korostaa käyttöliittymässä.

Laulajan käyttöliittymä peri samoja elementtejä valvojan käyttöliittymästä, kuten ylä- ja alapakit, joissa on pikatoiminnot ja laulujonon tieto (kuva 14). Laulajan alapalkin laulujonon tiedot ovat kuitenkin käyttäjäkeskeisiä ja kertovat, milloin laulajan oma seuraava lauluvuoro on tulossa.



Kuva 14. Laulajan käyttöliittymä KaraOkayn prototyypissä.

Alapakissa laulajalle esitetään hänen seuraavan kappaleensa sijoitus jonossa sekä arvio, kuinka kauan seuraavaan lauluvuoroon on vielä aikaa, ja arvioitu esityksen alkamisaika.

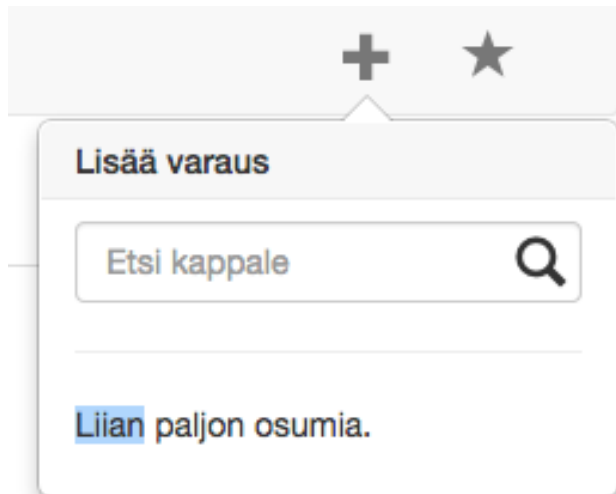
Laulajan valikko ja pikatoiminnot

Laulajan valikossa on kolme vaihtoehtoa: koti, kappaleet ja viestit:

- Koti-näkymä on valvojan tapainen työpöytä, jossa on laulajalle tärkeää tietoa laulujonosta ja tulevista lauluvuoroista.
- Kappaleet-näkymässä laulaja voi etsiä kappaleita ja lisätä kappaleita lauluvuoroon tai tallentaa kappaleita suosikeiksi.
- Viestit-näkymässä ovat järjestelmän viestit, kuten jos valvoja poistaa laulajan lauluvuoron, siirtää sitä myöhemmäksi tai joku hänen edeltään peruu lauluvuoron.

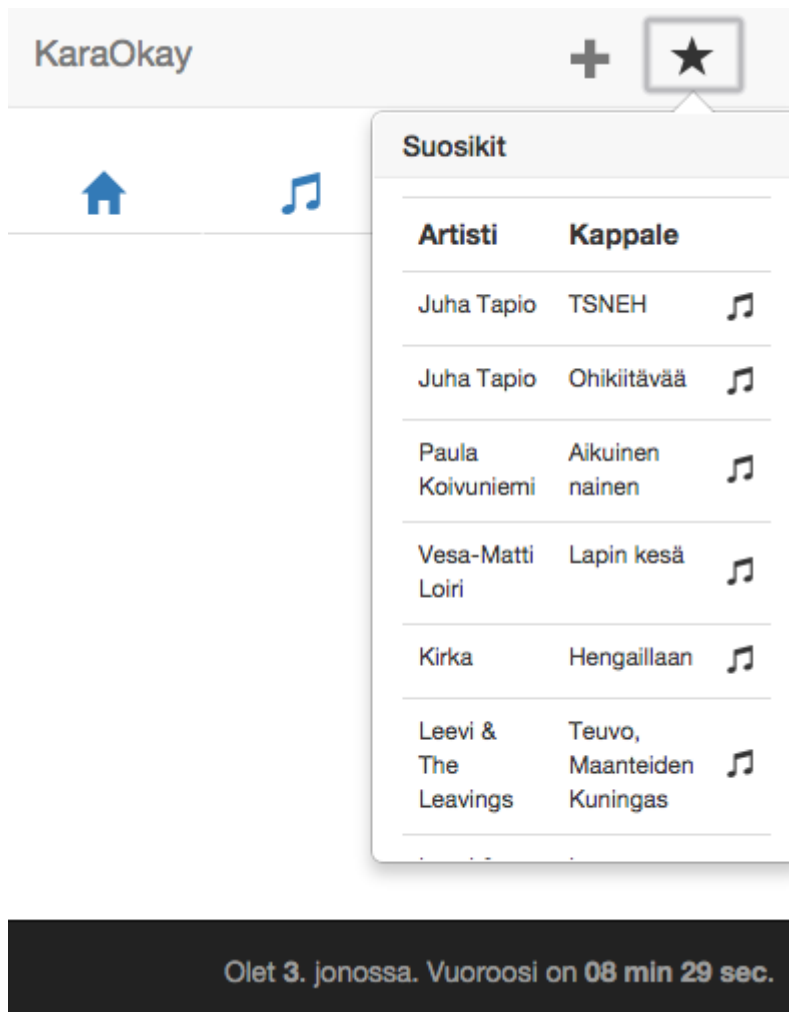
Viestien kohdalla navigaatiopalkissa on myös uusien viestien indikaattori, joka näyttää niiden lukumäärän.

Laulaja voi myös lisätä kappaleen laulujonoon pikatoimintojen avulla. Plus-toiminto avaa kappalehaun (kuva 15) pieneen paneeliin, jonka avulla laulaja voi suoraan hakea haluamansa kappaleen ja lisätä sen jonoon.



Kuva 15. laulajan pikatoiminto varauksen lisäykseen KaraOkayn prototyypissä.

Toinen laulajan pikatoiminto on suosikit (kuva 16), jonka avulla laulaja voi lisätä suoraan suosikeiksi merkitsemistään kappaleista haluamansa laulujonoon. Suosikeille ei ole omaa näkymää, vaan kappaleet lisätään suosikeiksi kappaleet-näkymän (kuva 17) kautta.



Kuva 16. Laulajan suosikit pikatoimintona KaraOkayn prototyypissä.

Laulajan kappalehaku

Kappaleet-näkymässä laulaja voi hakea kappaleita karaokepaikan kappalekirjastosta ja joko lisätä niitä suosikeiksi tai varata lauluvuoron kappaleella. Kappalelistauksessa (kuva 17) listataan kappaleiden perustiedot sekä toimintopainikkeet kullekin kappaleelle.

KaraOkay + ★

[Koti](#) [Kappaleet](#) [Viestit 9+](#)

Etsi kappale Q

Artisti	Kappale	Kesto	
System	Jee	13:19	+★
System1	Jee	13:17	+★
System4	Jee	13:14	+★
System5	Jee	13:13	+★
System	Jee	13:21	+☆
System1	Jee	13:18	+☆
System2	Jee	13:16	+☆
System3	Jee	13:15	+☆

Olet 3. jonossa. Vuorosi on 8 minuutin (16:50) kuluttua.

Kuva 17. Laulajan kappalelista KaraOkayn prototyypissä.

Toimintopainikkeissa oleva tähti kertoo, onko kappale lisätty laulajan suosikkeihin, ja se toimii päälle-/pois-kytkimenä. Musta tähti tarkoittaa kappaleen olevan suosikki. Plus-painike lisää kappaleen suoraan laulujonoon.

3.5 Siirtyminen KaraOkayhyn ja kappaleiden hankinta

Yleisimmistä kilpailevista järjestelmistä ei voi suoraan siirtää kappaleita, koska ne ovat usein kopiosuojattuja eikä niitä voi laillisesti toistaa erillisessä järjestelmässä. DVD:t voi kääntää mp4-tiedostoiksi huomioiden DVD:n omat tekijänoikeusehdot.

Pilvipalvelun kautta asiakas voi ostaa valmiita kappalepaketteja tai yksittäisiä kappaleita. Vaihtoehtona käyttäjä voi hankkia kappaleita myös itse joko DVD:nä ja kääntää ne tuettuun formaattiin. Järjestelmä tukee yleisimpiä karaokessa käytettyjä tiedostomuotoja, kuten mp3+cdg ja mp4.

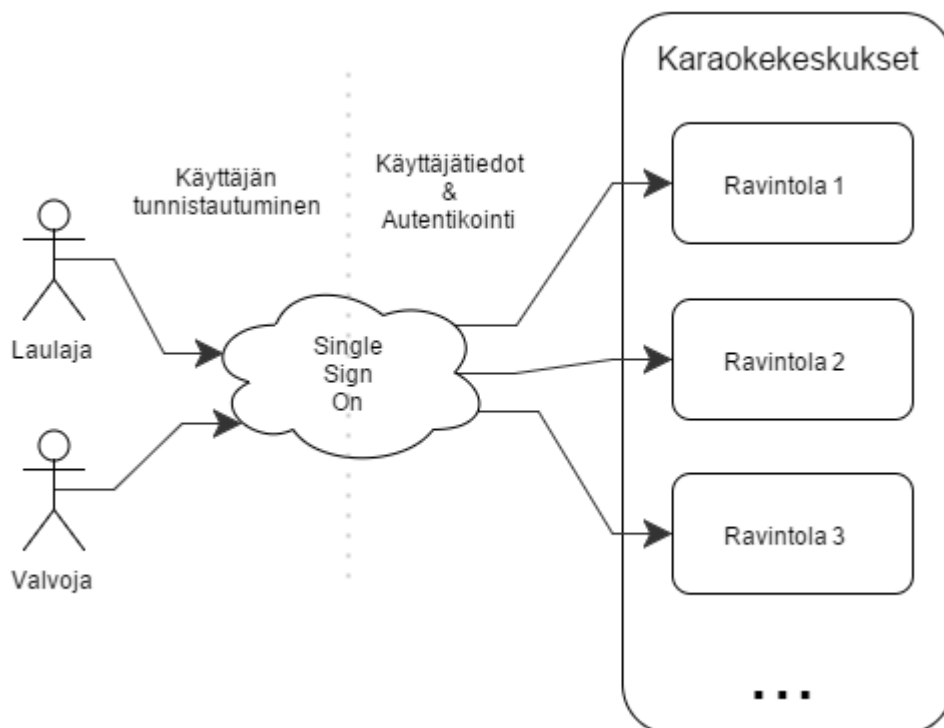
3.6 Keskitetty asiakasrekisteri ja Single Sign-on-palvelu

KaraOkay palveluntarjoajana ylläpitää keskuspalvelinta, joka tarjoaa laulajille Single Sign-On-palvelun asiakasjärjestelmiin sekä palveluntarjoajan omat sivustot, jossa on muun muassa lista kaikista asiakasjärjestelmistä. Järjestelmän palveluntarjoaja toimittaa asiakasjärjestelmän ylläpitäjälle baarin nimen mukaisen alidomain-nimen.

Palveluntarjoajan ylläpitämällä listassa on esillä baarin yleiset tiedot, kuten osoite, puhelinnumero, linkki sivustolle sekä pikalinkki reittiohjeisiin.

Käyttäjätietojen hallinta

Uusi käyttäjä rekisteröityy KaraOkay-järjestelmän palveluntarjoajan keskuspalvelimen Single Sign-On- eli kertakirjautumispalveluun, joka välittää käyttäjätiedot kullekin karaokebaarin asiakasjärjestelmälle (kuva 18), joka luo oman kopionsa käyttäjätilistä. Tämä mahdollistaa samojen käyttäjätunnusten käyttämisen eri baareissa, joissa on käytössä sama järjestelmä, vaikka ne ovat muuten itsenäisiä eivätkä kommunikoi toistensa kanssa.



Kuva 18. Single Sign-On -autentikoinnin periaate järjestelmässä.

Palvelun saavutettavuus

Laulajat löytävät kaikki asiakasjärjestelmät palveluntarjoajan verkkosivustoilta. Lisäksi asiakasjärjestelmän ylläpitäjä voi tulostaa yksilöidyn QR-koodin, joka on myös esillä ilmoitusruudulla.

4 Asiakaspäätelaitteet

KaraOkay-asiakasjärjestelmän ylläpitäjältä edellytetään päätelaite, johon on liitetty vähintään yksi ääni- ja videoulostulo. Tavallisesti ääniulostulon ja kaiutinjärjestelmän väliin on liitetty mikseri, joka yhdistää mikrofonit ja palvelusta toistettavan kappaleen. Videoulostulon kautta kappaleiden sanat voidaan esittää laulajille näyttöpäätteessä.

4.1 Vaadittavat ominaisuudet ja mahdolliset laitevaihtoehdot

KaraOkayssa keskityimme kevyiden asiakaspäätteiden ratkaisuihin, joiden päätarkoituksena on palvella käyttäjää yhteydenpitovälineenä isompaan ja tehokkaampaan keskustietokoneeseen. Täällaisesta on esimerkkinä Raspberry Pi, jota markkinoidaan yhden piirilevyn tietokoneena.

Raspberry Pi 2:ssa on HDMI-liitäntä ja 3,5 mm:n ääniliitäntä, johon voi liittää mikserin mikrofoneja varten. Raspberryyn voi tarvittaessa lisätä myös USB-WLAN-adapterin, jos laitetta haluaa käyttää langattomasti verkossa. Tuotteessa on integroituna Ethernet-liitin perinteistä, kiinteää laajakaistaa varten.

Käytännössä kuitenkin mikä tahansa internetyhteydellä varustettu moderni tietokone soveltuu päätelaitteeksi ja sitä voidaan laajentaa käyttäjän tarpeisiin mukautuvasti.

4.2 Kokoonpanot

Perinteisesti karaokelaitteissa on aina jonkinlainen mikseri, jolla saadaan yhdistettyä mikrofoneilta tuleva ääni ja esitettävän kappaleen taustääniraita. Mikseri on yhdistet-

tynä äänentoistojärjestelmään, joka vaihtelee tapauskohtaisesti tilan ja käyttötarkoituksen mukaan. Yksinkertaisimmillaan äänentoistojärjestelmäksi riittää yksittäinen vahvistimeen liitetty kaiutin, johon mikseristä tulevan yhdistetyn ääniulostulon voi liittää.

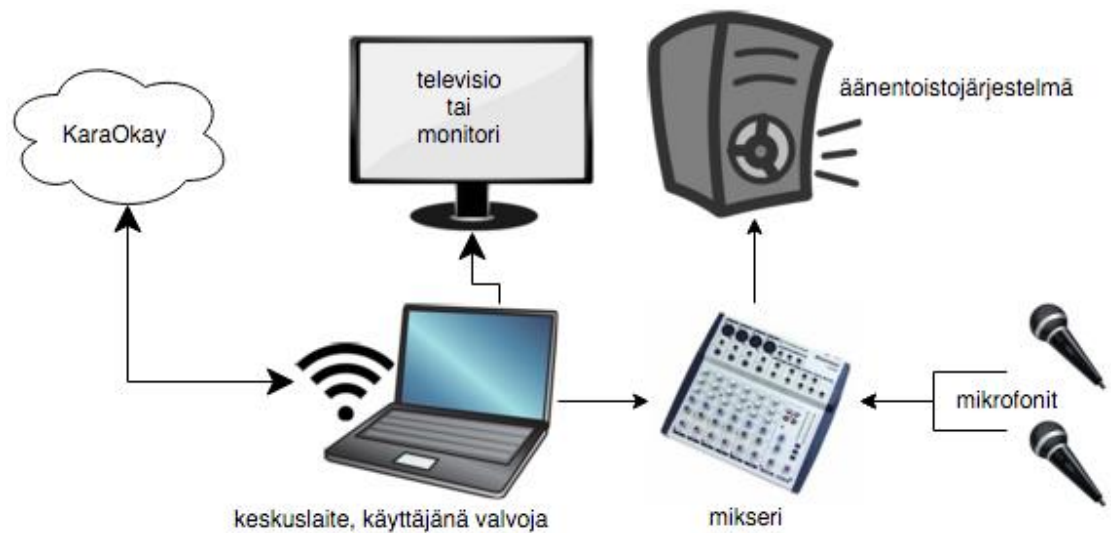
Mikseriin siis liitetään kaikki mikrofonit sekä äänisisäntulot. KaraOkayssa ääni tulee keskuslaitteesta, esimerkiksi Raspberry Pista, jonka päätehtävän on toistaa karaokekappaleet ja esittää yleisölle inforuutu kappaleiden välissä. Mikserin avulla voidaan kontrolloida yksittäisien tulojen äänenvoimakkuuksia. Kehittyneemmissä mikserissä voi myös esimerkiksi yhdistellä eri sisään- ja ulostuloja, prosessoida äänen taajuuskorjaimen läpi, lisätä tehosteita tai ohjata videotuloja.

Monissa karaokebaareissa näkee useita näyttöjä, joista voi seurata esityksiä ja jonoja. KaraOkayssä tähän pureudutaan toistimien kautta. Keskuslaitteesta voi videotoistimen avulla peilata kuvan useampaan näyttöpäätteeseen ja asettaa niitä eri puolille rakennusta.

Kaikissa KaraOkayn kokoonpanoissa tulee olla internetyhteys. Peruskomponentteina ovat keskuslaite, mikseri ja äänentoistojärjestelmä. Erillinen lisänäyttö ei ole pakollinen, mutta sen käyttö on suositeltavaa.

Suppea kokoonpanoratkaisu

Suppeassa kokoonpanossa keskuslaitteen ääniulostulo liitetään mikserin kautta äänentoistojärjestelmään ja videoulostulo monitoriin tai televisioon (kuva 19).

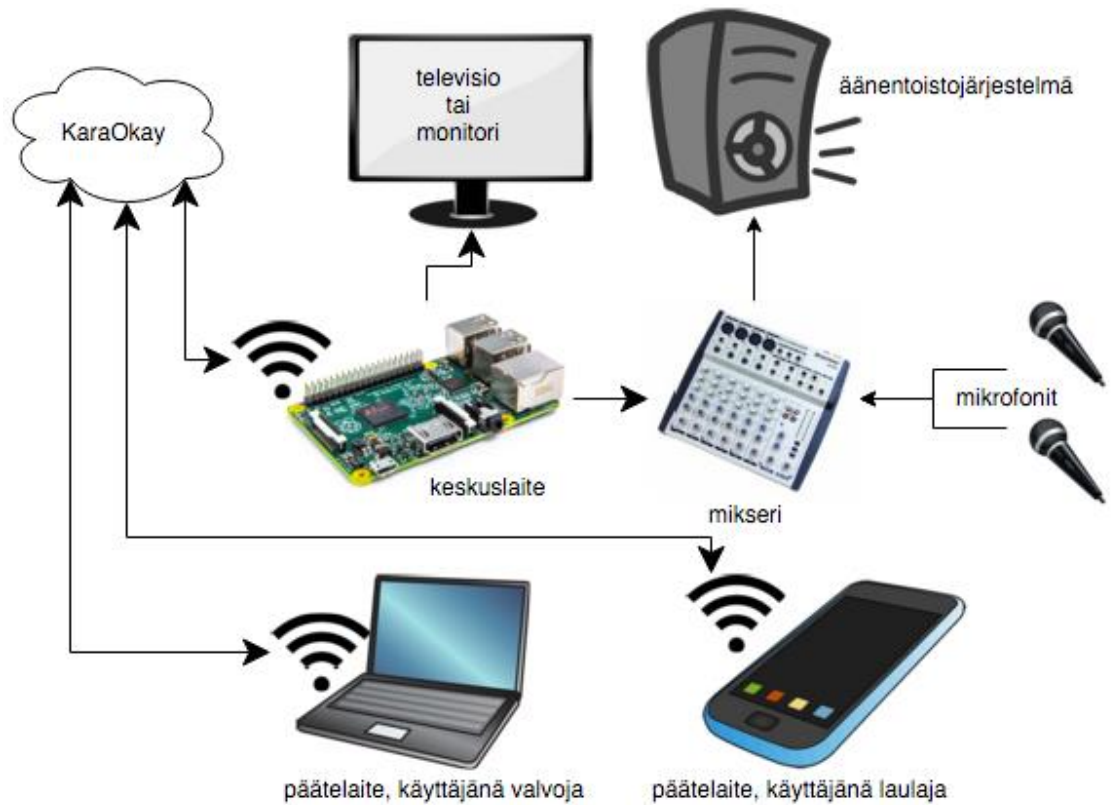


Kuva 19. Suppean KaraOkayn kokoonpanon kytkentäkaavio.

Keskuslaitteen kautta esitetään kappaleiden sanat videoulostulon kautta laulaessa, minkä vuoksi on hyvä varata toinen näyttö. Tämä mahdollistaa samanaikaisen käytön esityksen aikana.

Laajennettu kokoonpano

Verrattuna suppeaan kokoonpanoon laajennetussa kokoonpanossa on erillinen asiakaspäätelaite keskuslaitteena, jonka kautta kappaleet toistetaan. Verkon kautta KaraOkayyn on myös liitetty päätelaitteita, joita voi olla rajaton määrä ja ne voivat joko olla laulajia tai valvoja (kuva 20).



Kuva 20. Laajennetun KaraOkayn kokoonpanon kytkentäkaavio.

4.3 Raspberry Pi keskuslaitteena

Laitteistona Raspberry Pi 2 tarjoaa HDMI-liitännän, joka tukee Full HD -resoluutiota, sekä 3,5 mm:n stereo ääniulostulon, johon voi liittää omia tarpeitaan vastaavan mikserin mikrofoneja varten. Lisälaiteliitännöjä varten Raspberry Pi 2:ssa on neljä USB 2.0 -porttia, joihin voi liittää siirrettävän massamuistilaitteen kappaleita varten ja WLAN-adapterin, jos kiinteä internetyhteys ei ole mahdollinen. Laitteessa on myös Ethernet-portti, jonka kautta laitteen voi liittää internetiin, ja usein tämä on myös suositeltavampi vaihtoehto langattomuuden rajoitteiden takia. Verkkoysteys on kuitenkin pakollinen, koska järjestelmä on selainpohjainen ja sitä on hallinnoitava etänä toisella verkon laitteella, kuten tabletilla tai kannettavalla tietokoneella.

5 KaraOkayssä käytetyt tekniikat

Päätimme toteuttaa karaokesovelluksen PHP:lla hyödyntäen muun muassa Symfony 2 -sovelluskehystä, Doctrine ORM- ja AngularJS-kirjastoja sekä MySQL-tietokantaa. Angular mahdollistaa sovelluksen toiminnan Single-page appina, eli sivun ensilatauksen jälkeen ainoastaan päivitetään näkymää, mikä pienentää siirrettävän tiedon määrää verrattuna perinteiseen toteutustapaan, jossa koko HTML-sivun sisältö päivitetään kunkin pyynnön seurauksena. Lisäksi tämä mahdollistaa sulavamman oloisen käyttökokemuksen. (21, s. 497.)

5.1 Verkkosovellus vai työpöytäsovellus

Yleisimmin käytettyjen karaokeratkaisujen käyttöliittymä on toteutettu työpöytäsovelluksena, mutta jotta asiakkaiden olisi mahdollista käyttää palvelua omalla äylaitteellaan, tulisi jokaiselle tuettavalle alustalle lisäksi toteuttaa oma mobiilisovellus tai vaihtoehtoisesti sisällyttää järjestelmään verkkokäyttöliittymä. Toisaalta, jos järjestelmän toteuttaa kokonaan verkkosovelluksena, asiakaskäyttöliittymän rakentamiseen ei tarvitse nähdä niin paljoa vaivaa, koska yhteensopiva taustajärjestelmä on jo olemassa. (22.)

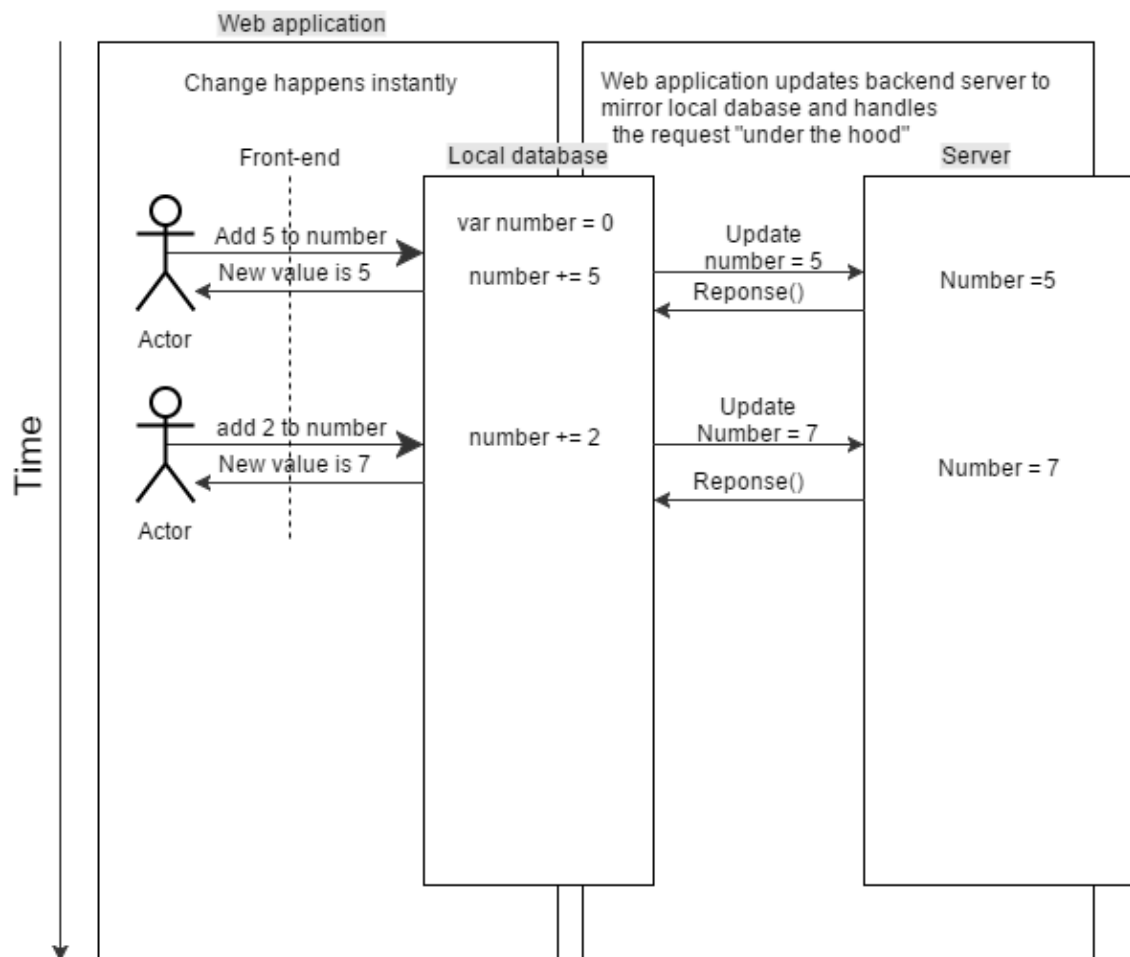
Eri alustojen tukemisen voi pienemmässä mittakaavassa rinnastaa verkkoselainten erilaisuuksiin. Suosituimmat mobiilikäyttöjärjestelmät edellyttävät sovellusten toteutuksen olevan tietyllä kielellä – Androidilla Java (23), iOS:lla Objective-C (24), Windows Phonnella JavaScript, C#, Visual Basic tai C++ (25). Selaimet sen sijaan enimmäkseen noudattavat W3C:n määrittämää spesifikaatiota, ja vaikka toteutukset eivät täysin vastaa sitä, kieli on kuitenkin sama ja tietyillä selaimilla ilmenevät ongelmat saa usein korjattua helposti (26). Selainriippumattomuutta voi myös huomattavasti parantaa kirjastoilla, jotka kykenevät toimimaan yhtenäisesti suosituimmilla selainmootoreilla, kunhan tiedostaa kunkin oikut ja omituisuudet. Ulkoasua yhtenäistävistä kehyksistä suosituin lienee Bootstrap, jota myös KaraOkay hyödyntää. Lisäksi verkkosovelluksen voi helposti siirtää natiiveille mobiilikäyttöjärjestelmille Apache Cordova -alustan avulla.

5.2 Toteutus Single-Page Appina

Single-page app (SPA) on verkkosovellus tai -sivusto, joka koostuu yksittäisestä verkkosivusta ja pyrkii tarjoamaan sulavamman käyttökokemuksen. SPA-sovelluksen sisältö

ladataan joko kokonaisuudessaan sivunlatauksen yhteydessä tai dynaamisesti tarpeen tullen käyttäjän toimien perusteella (21, s. 497). Sivun ei kuitenkaan missään vaiheessa lataudu uudelleen tai siirry toiselle sivulle prosessin aikana. Siinä, missä perinteiset verkkosovellukset siirtyisivät toiselle sivulle, SPA:n käyttöliittymä kommunikoi verkkopalvelimen kanssa Ajax-kyselyillä ja päivittää näkymää JavaScriptilla.

Single-page appit, joissa on tavoiteltu sulavampaa käyttökokemusta, on usein toteutettu stateful-tyyppisellä ratkaisulla, jossa palvelinpyynnöt korvataan sivuston tiloina. Tilan muutos tapahtuu välittömästi, ja muutos näkyy heti käyttäjälle. Sivusto saattaa kuitenkin vielä taustalla ladata tai tallentaa dataa palvelimelle ja synkronoida lokaaleja muutoksia keskuspalvelimelle. Verkkosovelluksella on siis oma kopio käytettävästä datasta, jonka muutokset peilataan palvelimelle ja pyynnöt toteutetaan sisäisesti Ajax-kyselyillä (kuva 21).



Kuva 21. Stateful-periaate ohjelmoinnissa.

Järjestelmä ylläpitää omaa tietokantaa ja hoitaa automaattisesti synkronoinnin palvelimen kanssa. Virhetilassa palvelimen tieto on aina prioriteetti, ja sovellus reagoi siihen esimerkiksi virheilmoituksella ja pyytää käyttäjää toistamaan toimenpiteen ja päivittämään käyttöliittymään tehdyt muutokset.

5.3 Järjestelmän vaatimukset

KaraOkay rakennettiin Symfony-PHP-sovelluskehiksen ympärille. Se vaatii tiettyjä ominaisuuksia palvelinympäristöltä toimiakseen haluamallamme tavalla. Koska kyseessä on verkkopalvelu, järjestelmä tarvitsee verkkopalvelimen. Kehityksen aikana käytimme Apache httpd:ta, mutta esimerkiksi myös nginx toimii (27).

Symfony hyödyntää reitityksissä Front Controller -periaatetta, joka ohjaa kyselyt saman PHP-tiedoston kautta. Jotta osoitteista saisi miellyttävämmän näköiset, järjestelmä käyttää Apachen moduulia "mod_rewrite". Rewrite-moduuli uudelleenohjaa RewriteCond-direktiivillä määritellyn ehdon mukaan kyselyt, jotka eivät vastaa olemassa olevaa tiedostopolkua DocumentRoot-direktiivin mukaisessa hakemistossa, ja RewriteRule ohjaa ne app.php-tiedostoon säilyttäen alkuperäisen URL-muodon (kuva 22).

```

1  <VirtualHost *:80>
2      ServerName karaokay.localhost
3      DocumentRoot "C:/www/karaokay/source/web"
4
5      <Directory "C:/www/karaokay/source/web">
6          AllowOverride All
7          Require all granted
8
9          RewriteEngine On
10         RewriteCond %{REQUEST_FILENAME} !-f
11         RewriteRule ^(.*)$ app_dev.php [QSA,L]
12     </Directory>
13
14     ErrorLog "logs/karaokay-error.log"
15 </VirtualHost>
16

```

Kuva 22. Apache httpd -palvelimen vhost-konfiguraatio KaraOkayssä.

Symfony-projekteissa käytetään usein Composer-riippuvuuksien hallintatyökalua, jonka kautta voi hallita sovelluksen eri komponenttien versioita ja joka pitää huolen niiden yhteensopivuuksista.

5.4 Web-sovelluksen kommunikaatorajapinnat

KaraOkay hyödyntää useita PHP-sovelluksen kehitystä helpottavia komponentteja, joista isoimmassa roolissa on Symfony-sovelluskehys. Symfony tarjoaa MVC-mallin mukaisen arkkitehtuurirakenteen ja sitoo yhteen kaikki muut sovelluksessa käytettävät kolmannen osapuolen komponentit bundle-kokonaisuuksien avulla, jotka mahdollistavat komponenttien käytön DI-kontainerin (Dependency Injection) kautta.

Dependency Injection on ohjelmistosuunnittelumalli, joka pyrkii ratkaisemaan komponenttien riippuvuudet käyttäen Inversion of Control -menetelmää (IoC). Menetelmän mukainen ratkaisu saa riippuvaisuutensa toteutuksen ulkopuolelta sen sijaan, että se itse päättäisi toteutuksesta (28). Perinteisessä ohjelmassa jokin luokka saattaisi kutsua funktiota, joka kirjoittaa lokitapahtumia tiedostoon. IoC-menetelmää hyödyntävällä luokalla sen sijaan olisi riippuvuus lokitapahtumia kirjaavaa luokkaa kuvaavaan rajapintaan. Rajapinnalla saattaa olla useita toteutuksia, joista yksi kirjoittaa tapahtumat tiedostoon ja toinen vaikkapa tietokantaan. Luokka käsittelee lokien kirjausta rajapinnan kautta, eikä sen tarvitse välittää, minkä toteutuksen funktiota se kutsuu. Toteutus injektoidaan luokalle tyypillisesti sen instantioinnin yhteydessä — tätä kutsutaan konstruktori-injektioksi — tai tarpeen mukaan suorituksen aikana setteri-injektiolla.

DI-kontainer on komponentti, jolle on konfiguroitu, miten mikäkin komponentti tulee rakentaa ja mitä riippuvuuksia sillä on. Näin se kykenee tarjoamaan instanssin siitä muualla ohjelmassa. Riippuvuuksien injektointi lisää huomattavasti sovelluksen testattavuutta muun muassa yksikkötesteillä verrattuna sovellukseen, joka kutsuisi suoraan muiden luokkien staattisia metodeja tai hakisi luokan instanssin käyttäen Singleton-anti-suunnittelumallia, koska tällöin luokalle voidaan injektoida oikean toteutuksen sijaan testikorvike eli rajapinnan toteuttava valeluokka, joka ohjaa kutsuvan luokan käyttäytymistä esimerkiksi palauttamalla tietynlaisen arvon tai heittämällä poikkeuksen. (29.)

Seuraavaksi tärkein ominaisuus on datan tallennus ja nouto tietokannasta. Varsinainen kommunikointi tietokannan kanssa on ulkoistettu Doctrine ORM -komponentille (tästä eteenpäin Doctrine). Sovelluksessa luodaan MVC-mallin model-osan mukaisia luokkia, joille määritellään Doctrinen tarjoamilla annotaatioilla säännöt, miten luokan ja sen attribuuttien tulisi vastata tietokantataulua. Luokkaa, jolla on tietokantaan peilattavia ominaisuuksia, kutsutaan entityksi (30). Kuvassa 23 on esimerkkinä Queue-entity eli jonossa

oleva lauluvuoron varaus, jolle on määritelty tietokantataulun nimeksi "Queue" sekä kolme kenttää:

- id: uniikki kokonaisluku, joka kasvaa yhdellä jokaista tietokantataulun uutta riviä kohden
- song: relaatio Song-entiteettiin, joka sisältää laulettavan kappaleen tiedot ja viittauksen kappaleen tiedostopolkuun
- singer: bi-direktionaalinen eli kaksisuuntainen relaatio Singer-entiteettiin, joka sisältää laulajan tiedot. Singer-entiteetillä on vastaavasti relaatio Queue-entiteettiin.

```

1  <?php
2
3  namespace Kara\QueueBundle\Entity;
4
5  use Kara\SongBundle\Entity\Song;
6  use Kara\SingerBundle\Entity\Singer;
7  use Doctrine\ORM\Mapping as ORM;
8
9  /**
10   * @ORM\Entity
11   * @ORM\Table(name="Queue")
12   */
13  class Queue
14  {
15      /**
16       * @ORM\Id
17       * @ORM\Column(type="integer")
18       * @ORM\GeneratedValue(strategy="AUTO")
19       */
20      private $id;
21
22      /**
23       * @ORM\ManyToOne(targetEntity="Kara\SongBundle\Entity\Song")
24       * @ORM\JoinColumn(nullable=false)
25       * @var Song
26       */
27      private $song;
28
29      /**
30       * @ORM\ManyToOne(targetEntity="Kara\SingerBundle\Entity\Singer",
31       *     inversedBy="queues"
32       * )
33       * @ORM\JoinColumn(nullable=false)
34       * @var Singer
35       */
36      private $singer;
37  }

```

Kuva 23. KaraOkayn Queue-entity. Attribuuttien setteri- ja getteri-metodit on jätetty pois kuvasta.

Doctrine generoi annotaatioiden pohjalta tietokantataulun ja sen kenttiin kohdistuvat avaimet ja rajoittimet (engl. key ja constraint). Kuvassa 24 on Queue-entityn pohjalta luotu taulu MySQL-tietokannassa. Song- ja singer-relaatiot tallennetaan kokonaisluokina, jotka toimivat viittauksina kunkin vastaavan taulun id-kenttään. Sisäisesti Doctrine hoitaa viittauksien ratkaisun automaattisesti, jolloin varsinaisen sovelluksen PHP-koodissa ei tarvitse käsitellä id-tietoja lainkaan. Rajoittimet pitävät huolen tietokannan eheydestä: jos taulusta Singer tai Song yrittää poistaa rivin, johon on viittaus Queue-tilussa, tietokanta palauttaa virheen. Jotta rivi voi poistaa, on viittaus poistettava ensin tai vaihtoehtoisesti Queue-tilun voisi muuttaa poistamaan vastaavat rivit, jos viitattava rivi

poistetaan toisesta taulusta. KaraOkayssä kuitenkin halutaan säilyttää historiatiedot, joten rivejä ei tulla poistamaan.

```
CREATE TABLE `queue` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `song_id` int(11) NOT NULL,
  `singer_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `IDX_BE3C5067A0BDB2F3` (`song_id`),
  KEY `IDX_BE3C5067271FD47C` (`singer_id`),
  CONSTRAINT `FK_BE3C5067271FD47C` FOREIGN KEY (`singer_id`) REFERENCES `singer` (`id`),
  CONSTRAINT `FK_BE3C5067A0BDB2F3` FOREIGN KEY (`song_id`) REFERENCES `song` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
```

Kuva 24. Doctrine ORM:n generoima MySQL-tietokantauseke KaraOkayn Queue-taulusta.

Symfony-sovelluskehiksen lisäksi järjestelmässä on käytössä toinen sovelluskehys, AngularJS, joka on Googlen kehittämä ja ylläpitämä JavaScript-kehys. Angular toimii välikätenä Symfonyn ja selaimen välillä, ja viimekädessä vastaa siitä, mitä käyttäjälle näytetään. Sovelluskehiksen käyttämät HTML-mallinteet on esitallennettu kehiksen välimuistiin, jolloin palvelimelta tarvitsee ensipyyynnön jälkeen pyytää vain oleellinen tieto, mikä minimoi siirrettävän datan määrän. Kuvassa 25 on SongController.list-kontrolleri ja sen route-määrittäjä. Käyttäjän siirtyessä sovelluksen polkuun /song/list hänelle renderöidään list.html-mallinteen mukainen HTML-näkymä — joka on tallennettu välimuistiin käyttäen Symfonyyn asennettua kolmannen osapuolen AppVentus/AsseticInjectorBundlea — ja samanaikaisesti SongServiceä käyttäen lähetetään palvelimelle pyyntö saatavilla olevien kappaleiden tiedoista. Vastauksen saavuttua kappaleiden tiedot päivitetään näkymään. Mikäli käyttäjä haluaa luoda lauluvuoron varauksen tietyllä kappaleella, QueueService lähettää sen palvelimelle toisessa pyynnössä, joka sisältää kappaleen uniikin id-tiedon. Järjestelmään kirjautuneen laulajan tiedot sen sijaan kulkevat mukana jokaisessa pyynnössä.


```

1  angular.module('kara.song.controller', ['ngRoute'])
2
3  .controller('SongController.list', ['$scope', 'songService', 'queueService',
4      function($scope, songService, queueService) {
5          songService.getSongs(function(songs) {
6              $scope.songs = songs;
7          });
8
9          $scope.queue = function(song) {
10             queueService.create(song);
11         };
12     }]);
13
14 .config(['$routeProvider',
15     function($routeProvider) {
16         $routeProvider
17             .when('/song/list', {
18                 name: 'app_song_list',
19                 controller: 'SongController.list',
20                 templateUrl: 'KaraSongBundle/Song/list.html'
21             });
22     }]);

```

Kuva 25. AngularJS:n MVC-mallin mukainen SongController KaraOkayssä.

Symfony ohjaa Angular-puolen QueueServicen lähettämä pyynnön QueueControllerin createAction:iin, joka noutaa viittauksen kappaleeseen tietokannasta (kuva 26). Tietokannasta noudettavan datan kommunikaatio on abstraktoitu Doctrinen EntityRepository-luokkaan, joka Song-entytyn tapauksessa on nimeltä SongRepository. Tämän jälkeen varaus tallennetaan käyttäen QueueServiceä, jota ei pidä sekoittaa Angularin puolella olevaan QueueServiceen. Service-luokat sisältävät järjestelmälogiikkaa ja ovat yleisesti vastuussa tietojen tallennuksesta tietokantaan käyttäen Doctrinen työkaluja. QueueServicen koodin voisi periaatteessa laittaa suoraan QueueControlleriin, mutta erillään ne voidaan myös yksikkötestata erikseen: QueueServicen testeissä varmistettaisiin tallennusprosessin toimivuus ja kontrollerin testeissä vastauksen oikeellisuus songId-tiedosta riippuen.

```

1 <?php
2
3 namespace Kara\QueueBundle\Controller;
4
5 use Kara\FrameworkBundle\Controller\Controller;
6
7 class QueueController extends Controller
8 {
9     public function createAction($songId)
10    {
11        $song = $this->getSongRepository()->findById($songId);
12        $this->getQueueService()->create($song);
13
14        return $this->createJsonResponse(array());
15    }
16 }

```

Kuva 26. Symfony Frameworkin MVC-mallin mukainen QueueController KaraOkayssä.

Lopuksi createAction lähettää vastauksena tyhjän JSON-olion. Vastauksen sisällöllä ei tässä tapauksessa ole merkitystä, vaan sillä, että Angularin kontrolleri saa tiedon, että pyyntö on mennyt onnistuneesti läpi.

5.5 Testaus

Järjestelmän kehityskaaren aikana sitä testataan jatkuvasti manuaalisesti käyttöliittymän kautta ominaisuuksia kehitettäessä, mutta myös ohjelmallisilla testeillä. Automatisoimalla testausprosessi voidaan helpommin varmistaa, että tehdyt muutokset eivät riko sovelluksen muita osa-alueita. Testit eivät kuitenkaan voi taata sovelluksen toimintaa kaikissa tilanteissa, mutta se takaa, että sovellus toimii tietyllä tavalla tietyissä tilanteissa. Testatun ominaisuuden toteutusta voi teoriassa muuttaa vapaasti, kunhan testi menee vielä läpi muutosten jälkeen. (31.)

Testien automatisoinnilla tarkoitetaan yleensä ohjelmallisten testien suorittamista ennen muutosten tallennusta kehityspalvelimelle tai välittömästi tallennuksen jälkeen. Jälkimmäisessä tapauksessa on suotavaa, että kehittäjä myös itse ajaa testit ennen tallennusta. Ideaalisesti testit ajetaan aina ympäristössä, joka muokkaa tuotantoympäristöä.

Yksikkötestaus

Yksikkötestaus on sovelluksen testausmenetelmä, jossa testataan yksittäisien sovelluskomponenttien käyttökelpoisuutta eristetyssä ympäristössä. Olio-ohjelmoinnissa yksikkötesti yleensä kattaa yhden haaran yksittäisestä julkisesta metodista. Alemman käyttöoikeustason, kuten private ja protected metodit testataan epäsuorasti julkisten metodien testeissä. (32, s. 426.)

Dependency injection -menetelmän, eli riippuvuuksien injektoinnin, hyödyntäminen lisää luokan testattavuutta erottamalla komponentit, joiden kanssa se kommunikoi, omiksi parametreikseen. Kuvassa 27 on esimerkki dependency injectionista FlashMessenger, jolle injektoidaan SessionInterface- ja TranslatorInterface-rajapinnat toteuttavat komponentit. Luokan tehtävänä on tarjota helppo ja yhtenäinen tapa toimittaa käyttäjälle virheilmoituksia kontrollerista käsin. Luokan ainoalle metodille annetaan parametrina käännettävän tekstin tunniste ja mahdolliset käännöksen käyttämät parametrit, jotka se syöttää kääntäjäkomponentille ja sen jälkeen lisää palautusarvon istuntokomponentin ilmoituksille varattuun "säkkiin".

```

1 <?php
2
3 namespace KaraOkay\MainBundle\Controller\Helper;
4
5 use Symfony\Component\HttpFoundation\Session\SessionInterface;
6 use Symfony\Component\Translation\TranslatorInterface;
7
8 class FlashMessenger
9 {
10     /**
11      * @var SessionInterface
12      */
13     private $session;
14
15     /**
16      * @var TranslatorInterface
17      */
18     private $translator;
19
20     /**
21      * @param SessionInterface $session
22      * @param TranslatorInterface $translator
23      */
24     public function __construct(SessionInterface $session, TranslatorInterface $translator)
25     {
26         $this->session = $session;
27         $this->translator = $translator;
28     }
29
30     /**
31      * @param string $translationKey
32      * @param array $parameters
33      */
34     public function showErrorMessage($translationKey, array $parameters = array())
35     {
36         $message = $this->translator->trans($translationKey, $parameters);
37         $this->session->getFlashBag()->add('error', $message);
38     }
39 }

```

Kuva 27. KaraOkayn FlashMessenger, jolle injektoidaan konstruktorissa Session ja Translator.

Luokkaa instantioidessa sille voidaan syöttää parametrina mitkä tahansa vaadittavat rajapinnat toteuttavat luokat. Yksikkötesteissä voi luoda riippuvuuksista testituplia eli valeluokkia, jotka pystytään konfiguroimaan käyttäytymään sen mukaan, mitä milloinkin halutaan testata. FlashMessengerin testeissä (kuva 28) instanssi on päätetty koostaa käännöskomponentin testikorvikkeesta ja käyttää istuntokomponentin oikeaa toteutusta. Tämä siksi, että käännökset on määritetty toisaalla eikä haluta, että testi hajoaa, jos käännöksiä muutetaan. Istuntokomponentti sen sijaan käytännössä vain säilöö arvon, ja sen saa myös helposti haettua ulos, joten ei ole syytä monimutkaistaa testiä luomalla siitäkin testikorviketta. Komponentti haetaan DI-containerilta käyttäen Symfony-sovelluskehityksen sille antamaa tunnistetta.

```

1 <?php
2
3 namespace KaraOkay\MainBundle\Tests\Controller\Helper;
4
5 use KaraOkay\FrameworkBundle\Test\ContainerTestCase;
6 use KaraOkay\MainBundle\Controller\Helper\FlashMessenger;
7 use Symfony\Component\HttpFoundation\Session\Session;
8 use PHPUnit\Framework\MockObject\MockObject;
9
10 class FlashMessengerTest extends ContainerTestCase
11 {
12     /**
13      * @var FlashMessenger
14      */
15     private $target;
16
17     /**
18      * @var Session
19      */
20     private $session;
21
22     /**
23      * @var PHPUnit\Framework\MockObject\MockObject
24      */
25     private $translator;
26
27     protected function setUp()
28     {
29         parent::setUp();
30
31         $this->session = $this->getContainer()->get('session');
32         $this->translator = $this->getMock('Symfony\Component\Translation\TranslatorInterface');
33         $this->target = new FlashMessenger($this->session, $this->translator);
34     }
35
36     /**
37      * @test
38      */
39     public function showError_message_should_translate_message_and_add_it_to_session_flash_bag()
40     {
41         $this->translator
42             ->expects($this->once())
43             ->method('trans')
44             ->with('build.error.insufficientPylons')
45             ->will($this->returnValue('You must construct additional pylons!'));
46
47         $this->target->showErrorMessage('build.error.insufficientPylons');
48
49         $messages = $this->session->getFlashBag()->get('error');
50
51         $this->assertEquals('You must construct additional pylons!', $messages[0]);
52     }
53 }

```

Kuva 28. KaraOkayn FlashMessengerin testiluokka.

Testin alussa annetaan translator-korvikkeelle ohjeet, miten sen tulee käyttäytyä, kun metodia kutsutaan tietyllä parametrilla. Tämän jälkeen kutsutaan testattavaa metodia ja haetaan istunokomponentilta säkki, jonne viestin odotetaan päätyneen. Lopuksi varmistetaan, että viestin sisältö täsmää testikorvikkeen "kääntämän" tekstin kanssa.

Testillä varmistetaan nyt, että metodi syöttää käännetyn tekstin istunnon säkkiin, mutta se ei tarkoita, että viesti päätyisi käyttäjän ruudulle asti. Luokan tehtävä ei kuitenkaan

ole varsinaisesti näyttää viestejä, vaan käyttää sovelluskehityksen tarjoamaa valmista mekanismia. Tämän mekanismin toimivuus ei kuulu luokan tai sen testin piiriin, vaan sille on omat testinsä sovelluskehityksen koodissa. Yksikkötestien näkökulmasta voidaan vain olettaa sen toimivan odotetusti.

Integraatiotestaus

Yksikkötestauksen eristetyn testiympäristön vastaparina on integraatiotestaus, jossa pyritään varmistamaan komponenttien välisen kommunikaation yhteistoiminta. Integraatiotestissä voidaan käyttää joko muutaman komponentin oikeaa toteutusta yhdessä tai vaikkapa koko järjestelmää sellaisenaan. Isoissa integraatiotesteissä tulee kuitenkin ongelmaksi virheen paikantaminen tilanteessa, jossa testi ei menekään läpi. Tästä syystä ne usein toimivat tukena yksikkötesteille eivätkä päätoimisena testausmenetelmänä.

KaraOkayssä kokonaisvaltainen integraatiotesti voisi tehdä HTTP-pyyynnön QueueControlleriin yrittäen lisätä kappaleen laulujonoon ja sen jälkeen varmistaa lopputulos tietokannasta. Sovelluksessa ei kuitenkaan juuri hyödynnetty integraatiotestausta, koska ne ovat työläämpiä toteuttaa kuin yksikkötestit ja hajoavat helpommin toiminnallisuuksien muuttuessa.

Käyttöliittymättestaus

Käyttöliittymättestaus on integraatiotestauksen muoto, jossa keskitytään käyttöliittymään. Aiempaa kappaleen lisäyksen integraatiotestiä tukemaan voisi lisätä käyttöliittymätestin, jossa simuloidaan linkin tai napin painallusta, joka saa selaimen lähettämään HTTP-pyyynnön sovellukselle. Testissä voitaisiin lisäksi varmistaa, että käyttäjän näkymä päivittyy odotetusti onnistuneen pyynnön seurauksena.

5.6 Mobiilisovellus

Mobiili- eli älylaitteella käytettävistä sovelluksista puhuttaessa tarkoitetaan niin sanottuja natiiveja sovelluksia, jotka on toteutettu käyttöjärjestelmän tarjoaman rajapinnan päälle. Android-ympäristössä se on toteutettu tavallisesti Javalla ja XML:llä, iOS:ssä Objective-C:llä tai Swiftillä. Mobiilisovelluksiksi luokitellaan myös verkkosivustot, joita voi käyttää älylaitteen selaimella, tai hybridit, jotka ovat näiden kahden yhdistelmiä.

Yleisesti ottaen tärkeimpänä syynä valita laiteläheiset ratkaisut perinteisen sivuston sijasta on rajapinnan tarjoamat järjestelmän komponentit, kuten push-notifikaatiot tai pääsy laitteiston komponentteihin. Hybridisovellukset tarjoavat osittaisen pääsyn laitteistoon sovelluksesta.

KaraOkayssa valitsimme toteutukseksi responsiivisen verkkosovelluksen, joka siis toimii useimmilla älylaitteilla ja selaimilla. Sivuston toteutuksessa käytetty Bootstrap on suunniteltu mobile first -periaatteella. Mobile first -periaattella kuvataan sivun rakennetta, joka on suunniteltu toimimaan lähtökohtaisesti mobiililaitteilla ja skaalautuvan täysimittaiseksi verkkosivustoksi. Verkkosovellus rajoittaa vähän sen laiteominaisuuksia, sen pyöriessä selaimella, mutta tulevaisuudessa se on helppo siirtää hybridisovellukseksi eri alustoille, jolloin sovellukseen tulee myös laiteläheiset ominaisuudet. KaraOkayn loppukäyttäjän rajapinta imitoi hyvin laiteläheisiä ulkoasuratkaisuja, mikä mahdollistaa hyvät jatkokehitysmahdollisuudet.

6 Jatkokehitys

Koska KaraOkayn alkuperäisen suunnitelman mukainen toteutus oli täysin käyttökelpoinen, jouduimme aikataulun vuoksi jättämään joitakin käyttökokemusta parantavia ominaisuuksia pois ensimmäisestä versiosta. Nämä ominaisuudet säilyvät edelleen työlliställä, ja niiden toteutus on määrää aloittaa julkaisun jälkeen, priorisoiden käyttäjäpalautteen ja verkkoanalytiikan perusteella.

6.1 HipHop Virtual Machine

Koska PHP on skriptikieli, sen suoritusnopeus on heikko verrattuna ohjelmointikieliin, joissa koodi käännetään konekielelle, koska PHP-koodi joudutaan parsimaan joka pyynnöllä uudelleen. Suoritusnopeutta voi jossain määrin parantaa erilaisilla välimuistiratkaisuilla.

Facebookin kehittämä HipHop Virtual Machine (HHVM) lähestyy ongelmaa toisesta näkökulmasta. HHVM on sovelluspohjainen virtuaalikone, joka perustuu ajonaikaiseen kääntämiseen (JIT, Just-In-Time). HHVM kääntää PHP-koodin x86-64-konekielelle, minkä jälkeen koodi suoritetaan natiivisti (33). Virtuaalikone on yhteensopiva suosituim-

pien verkkopalvelimien kanssa, kuten Apache, nginx ja Lighttpd, ja se on suoritusteholtaan vertailuanalyyysien perusteella jopa kymmeniä kertoja nopeampi kuin PHP yksinään (34).

Jotta järjestelmän saa toimimaan virtuaalikoneen kanssa, siihen voi kuitenkin joutua tekemään joitain muutoksia, koska HHVM käyttäytyy joissain tapauksissa eri tavalla kuin PHP (35). Tämän lisäksi PHP:sta on lähiaikoina tulossa versio 7, joka yhteisön suorittamien suorituskykytestien perusteella vaikuttaisi olevan varsin varteenotettava haastaja HHVM:lle (36). PHP7:lla on myös niin sanottu kotikenttäetu, koska se on PHP:n ylläpitäjien kehittämä ja näin ollen todennäköisemmin vastaa dokumentoituja ominaisuuksia.

6.2 Tulevat ominaisuudet

Sovelluksen suunnittelun ja kehityksen aikana tulee jatkuvasti uusia ideoita ja kehitysehdotuksia. Kaikkia niitä ei kuitenkaan voida toteuttaa ensimmäiseen versioon, koska muuten julkaisupäivä siirtyisi vastaavasti aina vain myöhemmäksi. Seuraavassa on listattuna joitakin hyväksi toteamiamme mutta ei-kriittisiä ominaisuuksia, joita yleisesti kutsutaan nice-to-have-ominaisuuksiksi. Nämä ominaisuuden mitä luultavimmin toteutetaan sovelluksen myöhemmässä versiossa.

- Älykäs laulajien kiertojärjestelmä rytmittäisi lauluvuorot siten, että mahdollisimman moni pääsisi laulamaan.
- Laulaja voi saada alkuperäisen esityksen tai coverin kuulokkeiden kautta omasta älylaitteestaan streamattuna oman lauluvuoronsa aikana, mikä helpottaa esittämistä ja ajoittamista.
- Laulaja voi esikuunnella kappaaleen tai osan siitä kappaletta valitessaan, mikä on mahdollista toteuttaa esimerkiksi Spotifyn Web API:n kautta.
- Laulaja voi esimerkiksi hyräilemällä tai laulaen osan kappaaleesta etsiä kappaleita ja järjestelmä ehdottaa kappaleita, jotka muistuttavat näytettä. Mitä paremmin laulaja osaa imitoida kappaletta, sitä varmemmin järjestelmä tunnistaa sen.

- Laulaja voi itse aloittaa oman laulunsa omasta äylaitteestaan oman lauluvuoronsa alkaessa. Yleensä karaokeisäntä käynnistää laulun, joten tällä ominaisuudella voitaisiin vähentää isännän työtaakkaa.

7 Yhteenveto

Insinööriyön tavoitteena oli suunnitella ja kehittää kuluttajahintaystävällinen ammattitason karaokejärjestelmä harraste- ja ammattikäyttöön. Suunnitteluvaiheessa perehdyttiin kilpaileviin järjestelmiin ja kartoitettiin niiden keskeisimmät ominaisuudet. Kartoituksen jälkeen määritettiin halutut ominaisuudet uudenlaiseen järjestelmään.

Alustavan suunnittelun pohjalta päädyttiin kehittämään verkkopohjainen karaokesovellus käyttäen Symfony-sovelluskehystä taustajärjestelmänä ja AngularJS:ää käyttöliittymäkehystenä. Käytössämme olivat myös ketterät kehitysmenetelmät ja versionhallinta, jotka tekivät yhdessä kehittämisestä virtaviivaista. Ohjelmoinnissa pyrimme noudattamaan Lean-periaatteita, eli keskityimme siihen, mikä tuottaa tuotteelle arvoa ja yritimme minimoida ylimääräisen työn tekemistä.

Projektin edetessä sen toimintaperiaate kehittyi laitelähtöisestä toteutuksesta pilvipalveluksi, mutta tämä ei varsinaisesti vaikuttanut jo suunniteltuun kokonaisuuteen. Saimme paremman kuvan siitä, mikä olisi kuluttajalle houkutteleva ratkaisu. Näkökulman muutos ei varsinaisesti vaatinut toimenpiteitä teknisen puolen osalta, sillä palvelu ei itsessään muuttunut vaikka se paketoitiin erilaiseksi lopputuotteeksi.

Projektin alussa asetetut tavoitteet saavutettiin onnistuneesti. Työssä tehtiin kilpailevien tuotteiden vertailu, ja sen pohjalta kriittisistä ja toissijaisista ominaisuuksista kootusta dokumentaatiosta saatiin kattava. Olemme siihen tyytyväisiä. Lisäksi kehittäessämme prototyyppiä löysimme komponentit, jotka muodostaisivat toimivan ja yhtenäisen pohjan varsinaiselle toteutukselle.

Verkkosovelluksiin verrattuna pilvipalveluratkaisujen kehittäminen on moniosaisempaa, koska ne toimivat hajautetusti eri päätelaitteilla. Tuotteistaminen taas on iso osa kokonaisvaltaista tuotekehitystä, ja heti alkuvaiheessa on mietittävä kysymyksiä, kuten kuka tai mikä on tuotteen kohderyhmä ja mitkä ovat sen asettamat vaatimukset kilpailutasoiselle tuotteelle.

KaraOkayn kehitystyötä jatketaan tulevaisuudessa, ja kunhan sen todetaan vastaavan kaupallisen käytön vaatimuksia, voidaan sen markkinointi aloittaa. Pilvipalveluna karaoken laulantaan suunnattu järjestelmä toimii erinomaisesti: jälleenmyynti on yksinkertaista, koska käyttöönottokustannukset ovat hyvin pienet verrattuna laitekeskeisiin ratkaisuihin ja nykyään kaikilla on älypuhelin kaikkialla mukana, joten päätelaitteista ei ole pulaa.

Lähteet

- 1 Access Control Lists – OpenSUSEa suomeksi. 2013. Verkkodokumentti. openSUSE. <http://opensuse.fi/Access_Control_Lists>. Luettu 15.7.2015.
- 2 DOM (JavaScript ja VbScript) opas : Dokumenttioliomali. 2015. Verkkodokumentti. 2kmediat.com. <<http://www.2kmediat.com/dhtml/dokumenttimalli.asp>>. Luettu 4.11.2015.
- 3 MICHAEL KENNEDY and JOYCE BOURNE. "karaoke." The Concise Oxford Dictionary of Music. 1996. Verkkodokumentti. Encyclopedia.com. <<http://www.encyclopedia.com/doc/1O76-karaoke.html>>. Luettu 18.4.2015.
- 4 History of Karaoke | Karaoke Scene Magazine Online |. .: History Of Karaoke :.. 2015. Verkkodokumentti. Karaoke Scene Magazine Online. <<http://karokescene.com/history/>>. Luettu 5.6.2015.
- 5 Brewer, Bryan & Key, Edd. L.1987. The Compact Disc Book. s.l. : Harcourt Brace Jo-vanovich, s. 96.
- 6 Machine for karaoke von. 2004. Verkkodokumentti. U.S. Army. <<https://commons.wikimedia.org/wiki/File:KaraokeMachine.JPG> >. Luettu 24.11.2015.
- 7 History of the MP3+G® Format. 2011. Verkkodokumentti. tricerasoftware.com. <<http://tricerasoftware.com/history.html>>. Luettu 21.10.2015.
- 8 Karaoke USA GF830 DVD/CDG Karaoke Machine with Bluetooth & SD Slot. 2015. Verkkodokumentti. Ace Karaoke Corporation. <<http://ep.yimg.com/ay/acekaraoke/karaoke-usa-gf830-dvd-cdg-karaoke-player-with-sd-slot-1.gif>>. Luettu 24.11.2015.
- 9 SingStar™ Mic. SingStar. 2014. Verkkodokumentti. Sony Computer Entertainment Europe. <<http://www.singstar.com/fi/app.html>>. Luettu 21.10.2015.
- 10 eSongBook - What is eSongBook?. 2015. Verkkodokumentti. KSF Entertainment Group Ltd. <<http://www.esongbook.net/shop>>. Luettu 15.4.2015.
- 11 Siglos Karaoke Professional - Playback management. 2009. Verkkodokumentti. Siglos Karaoke Professional. <http://www.powerkaraoke.com/help/siglospro/source/idh_playback_management.htm>. Luettu 15.4.2015.
- 12 Karaoke Software for hosting your own shows: Siglos Karaoke Professional. 2015. Verkkodokumentti. Siglos Karaoke Professional. <http://www.powerkaraoke.com/src/prod_siglos-karaoke-professional.php>. Luettu 24.11.2015.

- 13 Tilastokeskus - Väestön tieto- ja viestintätekniiikan käyttö 2014. 2014. Verkko-dokumentti. Väestön tieto- ja viestintätekniiikan käyttö. <http://www.stat.fi/til/sutivi/2014/sutivi_2014_2014-11-06_tie_001_fi.html>. Päivitetty 25.11.2014. Luettu 13.4.2015.
- 14 Siglos Karaoke Professional - Singer Announcement Screen. 2015. Verkko-dokumentti. Siglos Karaoke Professional. <http://www.powerkaraoke.com/help/siglospro/source/idh_singer_announcement_screen.htm>. Luettu 24.11.2015.
- 15 Peliarvostelu: SingStar SuomiBileet. 2015. Verkkodokumentti. Oy Entertainment Media. <<http://www.xtv.fi/pelit/peliarvostelu-singstar-suomibileet>>. Luettu 16.4.2015.
- 16 Products | KSF Player. 2015. Verkkodokumentti. KSF Entertainment Group Ltd. <<http://www.ksfplayer.com/info/shop/>>. Luettu 24.11.2015.
- 17 Karaokeparty.com - Online Karaoke nettiselaimessasi. 2015. Verkkodokumentti. KaraokeParty Online Gaming AB. <<http://www.karaokeparty.com/fi/>>. Luettu 24.11.2015.
- 18 Sing! Karaoke by Smule – Android-sovellukset Google Playssa. 2015. Verkko-dokumentti. Smule, Inc. <<https://play.google.com/store/apps/details?id=com.smule.singandroid&hl=fi>>. Luettu 24.11.2015.
- 19 Free Karaoke Software - KaraFun Player. 2015. Verkkodokumentti. Recisio. <<http://www.karafun.com/karaokeplayer/>>. Luettu 24.5.2015.
- 20 PyKaraoke - Karaoke for Linux. 2015. Verkkodokumentti. pykaraoke. <<http://ki-bosh.org/pykaraoke/faq.php>>. Luettu 24.5.2015.
- 21 Flanagan, David. 2006. JavaScript - The Definitive Guide. 5th edition. Sebastopol : O'Reilly.
- 22 Designing for Web or Desktop?. 2002. Verkkodokumentti. Informant Communications Group and Microsoft Corporation. <<https://msdn.microsoft.com/en-us/library/ms973831.aspx>>. Luettu 17.4.2015.
- 23 Google's Android SDK Bypasses Java ME in Favor of Java Lite and Apache Harmony. 2007. Verkkodokumentti. InfoQ.com. <<http://www.infoq.com/news/2007/11/android-java>>. Luettu 18.4.2015.
- 24 About Objective-C. 2014. Verkkodokumentti. Apple Inc. <<https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>>. Luettu 18.4.2015.

- 25 What's a Universal Windows Platform (UWP) app? - Windows app development. 2015. Verkkodokumentti. Microsoft. <<https://msdn.microsoft.com/library/windows/apps/dn726767.aspx>>. Luettu 18.4.2015.
- 26 There's No Such Thing as a "Standards-Compliant" Browser | Impressive Webs. 2010. Verkkodokumentti. Impressivewebs. <<http://www.impressivewebs.com/no-standards-compliant-browser>>. Luettu 18.4.2015.
- 27 Configuring a Web Server (The Symfony CookBook). 2015. Verkkodokumentti. SensioLabs. <http://symfony.com/doc/2.7/cookbook/configuration/web_server_configuration.html>. Luettu 8.6.2015.
- 28 Designing Reusable Classes. 1988. Verkkodokumentti. Department of Computer Science University of Illinois at Urbana-Champaign. <<http://www.laputan.org/drc/drc.html>>. Päivitetty 5.11.2005. Luettu 29.4.2014.
- 29 James Shore: Dependency Injection Demystified. 2006. Verkkodokumentti. James Shore. <<http://www.jamesshore.com/Blog/Dependency-Injection-Demystified.html>>. Luettu 18.7.2015.
- 30 Getting Started with Doctrine — Doctrine 2 ORM 2 documentation. 2015. Verkkodokumentti. Doctrine. <<http://docs.doctrine-project.org/projects/doctrine-orm/en/stable/tutorials/getting-started.html>>. Luettu 14.4.2015.
- 31 Software Testing. 1999. Verkkodokumentti. Carnegie Mellon University. <http://www.ece.cmu.edu/~koopman/des_s99/sw_testing>. Luettu 9.11.2015.
- 32 Kolawa, Adam & Huizinga, Dorota. 2007. Automated Defect Prevention: Best Practices in Software Management. Wiley-IEEE Computer Society Press
- 33 The HipHop Virtual Machine. 2011. Verkkodokumentti. Facebook. <<https://www.facebook.com/notes/facebook-engineering/the-hiphop-virtual-machine/10150415177928920>>. Luettu 9.12.2011.
- 34 FasterCGI with HHVM « HHVM. 2013. Verkkodokumentti. Facebook. <<http://hhvm.com/blog/1817/fastercgi-with-hhvm>>. Luettu 22.4.2015.
- 35 Symfony 2.3 achieves 100% HHVM compatibility (Symfony Blog). 2015. Verkkodokumentti. Javier Eguiluz. <<http://symfony.com/blog/symfony-2-3-achieves-100-hhvm-compatibility>>. Luettu 26.11.2015.
- 36 PHP 5.6 vs HHVM 3.7 vs PHP 7 Benchmarks - Best Host News. 2015. Verkkodokumentti. Jonathan Griffin. <<https://www.besthostnews.com/php-5-6-vs-hhvm-3-7-vs-php-7-benchmarks>>. Luettu 26.11.2015.