

Juha Pulliainen

Mobiilioppimisen pelillistäminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

27.5.2013

Tekijä Otsikko	Juha Pulliainen Mobiilioppimisen pelillistäminen
Sivumäärä Aika	43 sivua 27.5.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	mediatekniikka
Suuntautumisvaihtoehto	digitaalinen media
Ohjaaja	lehtori Ilkka Kylmäniemi
<p>Mobiilioppiminen on yleistynyt nopeasti viime vuosien aikana. Syitä yleistymiseen ovat laitekannan, verkkoyhteyksien ja ohjelmistojen kehittyminen sekä käyttäjien tottuminen mobiiliverkkopalveluihin. Mobiilioppimispalveluiden haasteena on kuitenkin saada käyttäjät pysymään palvelussa uutuudenviehätyksen mentyä ohi. Jos käyttäjiä ei saada sitoutumaan palveluun, he lopettavat palvelun käytön.</p> <p>Pelillistäminen tähtää käyttäjämotivaation lisäämiseen ja käyttäjän sitouttamiseen erilaisilla pisteitys- ja palkitsemistavoilla. Insinööriyön tarkoituksena oli selvittää mahdollisuuksia mobiilioppimisen pelillistämiseen. Työssä toteutettiin pelimäinen mobiilikäyttöön tarkoitettu verkkosovellus, jossa käyttäjät pääsevät ratkaisemaan tehtäviä ja kilpailemaan keskenään. Työ tehtiin verkko-opetukseen erikoistuneen yrityksen tilauksesta, ja sitä on tarkoitus käyttää apuna yrityksen mLearning-ratkaisujen tutkimuksessa ja tuotekehityksessä. Työssä arvioitiin pelillistämisen eri osa-alueita, moninpelialustoja ja pisteiden ansainta- ja käyttömalleja. Arvioiden pohjalta suunniteltiin monipelikonsepti, jossa otettiin huomioon sisällön, käytettävyyden ja arkkitehtuurivalintojen vaikutuksia.</p> <p>Insinööriyössä toteutettu verkkosovellus tarjoaa mahdollisuuden harjoitella matemaattista tehtävänratkaisua kolmella eri tavalla. Perinteisin tavoista on vapaa harjoittelu, jossa käyttäjän tulee onnistua valitsemaan monivalintatehtävien vastausvaihtoehdoista oikea. Toinen tapa on ajastettu tehtävänratkaisu tietokonevastustajaa vastaan. Kolmas tapa on mahdollisuus haastaa toinen verkkosovelluksessa kirjautuneena oleva käyttäjä reaaliaikaiseen moninpeleihin, jossa käyttäjä voi saada pisteitä suhteellisesti enemmän itseään kokeneempien käyttäjien voittamisesta.</p> <p>Verkkosovellusta arvioitaessa löydettiin sekä konseptista että teknisestä ratkaisusta ominaisuuksia, joita voidaan kehittää eteenpäin ja hyödyntää tilaajayrityksen tuotannossa.</p>	
Avainsanat	verkko-oppiminen, mobiilioppiminen, e-learning, eLearning, m-learning, mLearning, MOOC, monipelit, pelillistäminen, pisteitysjärjestelmät

Author Title	Juha Pulliainen Gamification of mobile learning
Number of Pages Date	43 pages 27 May 2013
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructor	Ilkka Kylmäniemi, Lecturer
<p>Mobile learning has been growing rapidly during the past years. Reasons for this are improvements in the device base, network connections, software and users becoming familiar with networked mobile services. However, the challenge of mobile learning services is retaining the users once the novelty factor has passed. If the service fails to commit users to it, they will eventually stop using the service.</p> <p>Gamification aims to increase user motivation and retention with different point systems and rewarding methods. The aim of this thesis was to find out the possibilities for gamification of mobile learning. A game-like web application intended for mobile use, in which users get to solve exercises and compete with each other, was implemented during the creation of this thesis. This thesis was done for a company specialising in online learning, and the gathered experiences are planned to be used to aid the research and development of the client company's mLearning solutions. Different areas of gamification, multiplayer platforms, point earning and point utilization models were reviewed. Based on these, a multiplayer concept was designed, taking content, usability and architecture considerations into account.</p> <p>The implemented web application offers the possibility to practice mathematical exercises in three different ways. The most traditional way is free practice, where users must be able to find correct answers in multiple choice questions. Another way is a timed exercise against a computer opponent. A third way is the possibility to challenge another online user to a real-time multiplayer game, where users gain more points by defeating more experienced opponents.</p> <p>When evaluating the web application, some features which can be developed further and utilized in the client company's production were found.</p>	
Keywords	online learning, mobile learning, e-learning, eLearning, m-learning, mLearning, MOOC, multiplayer games, gamification, point systems

Sisällys

Lyhenteet ja määritelmät

1	Johdanto	1
2	Mobiilioppimisen pelillistäminen	2
2.1	Mobiilioppiminen	2
2.2	Pelillistäminen	3
2.3	Moninpeliympäristöt	6
3	Pelimäisen verkkosovelluksen suunnittelu	7
3.1	Vaatimukset	7
3.2	Sisältö ja tehtävät	9
3.3	Arkkitehtuuri ja rakenne	11
3.4	Tilatiedon synkronointi	14
3.5	Ajastettu tehtävänratkaisu	17
3.6	Pisteitys	18
4	Pelimäisen verkkosovelluksen toteutus	19
4.1	Työtavat ja komponentit	19
4.2	Kirjautuminen ja rekisteröityminen	20
4.3	Ominaisuudet	22
4.4	Testaus	35
4.5	Jatkokehitysmahdollisuudet	38
5	Yhteenveto	40
	Lähteet	41

Lyhenteet ja määritelmät

API	<i>Application Programming Interface</i> . Ohjelmointirajapinta, joka tarjoaa mahdollisuuden käyttää järjestelmää ennalta määritellyillä tavoilla.
Bundler	Gem-paketinhallintajärjestelmän hallintaohjelmisto.
CSS	<i>Cascading Style Sheets</i> . Www-sivustoilla yleisesti käytetty tyylimäärittäytapa ja tiedostotyyppi.
Gem	Ruby-ohjelmointikielen kanssa käytetty paketinhallintajärjestelmä ja lisäosan tyyppi.
HTML5	<i>Hypertext Markup Language 5</i> . Www-sivustoilla yleisesti käytetyn hypertekstimuodon versio 5.
.html.erb	Ruby on Rails -ohjelmistokehyksen käyttämä dynaamisten näkymien tiedostomuoto.
HTTP	<i>Hypertext Transfer Protocol</i> . Hypertekstin siirtoprotokolla, jota käytetään yleisesti www-sivustoilla.
mLearning	<i>Mobile learning</i> . Mobiilioppiminen, eli mobiililaitteilla käytävä verkkopohjainen oppimiskäytäntö.
MOOC	<i>Massive Open Online Course</i> . Suurelle osallistujamäärälle tarkoitettu verkkokurssi, johon kuka tahansa voi osallistua.
MVC	<i>Model-View-Controller</i> . Sovellusarkkitehtuuri, jossa mallit, näkymät ja käsittelijät toimivat toisistaan erillään.
OWASP	<i>Open Web Application Security Project</i> . Avoin projekti, joka seuraa ja listaa yleisimpiä tietoturva-avoittuvuuksia owasp.org -verkkosivustolla.
REST	<i>Representational State Transfer</i> . Malli ohjelmointirajapinnan tarjoamiseen HTTP-protokollalla.

1 Johdanto

Insinööriyön tarkoituksena on tutkia verkossa mobiililaitteilla tapahtuvan matematiikanopetuksen pelillistämistä. Työssä perehdytään pelimäisen lähestymistavan tapoihin, tekniikoihin ja mahdollisuuksiin vaikuttaa opiskelijoiden käyttökokemukseen ja motivaatioon.

Perinteiset verkko-opetuskonseptit on yleensä tarkoitettu itseopiskeluun, ja käyttäjän oletetaan saavan motivaation opiskeluun jostain muualta, kuten työ- tai opiskelupaikalta. Sosiaalinen kanssakäyminen opiskelijatovereiden ja opettajien kanssa voidaan toteuttaa kontaktiopetuksen tai työpajojen aikana. Verkko-opetus voi myös pohjautua verkkokeskustelun käyttöön, ja tällöin opettajan rooli keskustelun oikeaan suuntaan ohjaajana on tärkeä. (Ihanainen & Kalli 2009: 51–52; Ram ym. 2011.)

Yleissivistävän, vapaasti verkossa jaeltavan oppisisällön yhteyteen ei fyysistä opiskelumahdollisuutta, kontaktiopetusta tai ohjattua verkkokeskustelua kuitenkaan aina ole mahdollista toteuttaa järkevästi. Oppimiskokemus tai -palvelu voi houkuttaa opiskelijoita laadukkaalla asiasisällöllä ja selkeällä graafisella esitystavalla. Mielenkiinto voidaan saada pysymään paremmin yllä käyttämällä erilaisia aktivoivia tehtäviä. Sisällön laadukkuus ja miellyttävä ulkoasu eivät kuitenkaan välttämättä riitä. Sosiaalisen interaktion puuttuminen voi vähentää käyttäjien motivaatiota ja johtaa opiskelijoiden välisten suorituserojen kasvuun. Käyttäjät saadaan sitoutumaan palveluun paremmin sosiaalisen kanssakäymisen ja yhdessä tekemisen kautta. Mikäli opiskelutoveri on jo palvelussa, myös rima käyttäjäksi rekisteröitymiselle on matalammalla. (Järvelä & Häkkinen 2006: 231; Ram ym. 2011.)

Insinööriyössä suunnitellaan mobiililaitteilla käytettäväksi tarkoitettu ja moninpeliympäristönä toimiva verkkosovellus, jossa käyttäjät voivat nähdä toisia sisäänkirjautuneena olevia käyttäjiä ja tehdä harjoituksia yhdessä. Käytettävissä olevia teknologiamahdollisuuksia vertaillaan, ja niistä valitaan sopivimmat verkkosovellusta varten.

Työ tehdään Bitville Oy:n tilauksesta, ja sen tarkoituksena on kehittää yrityksen mLearning-konseptia. Suunniteltavaa verkkosovellusta on tarkoitus käyttää Bitville Oy:n tutkimus- ja tuotekehitystyökaluna ja tulevien pelillistämiskonseptien osa-alueiden ja jatkokehitysmahdollisuuksien kartoittamisen apuna.

2 Mobiilioppimisen pelillistäminen

2.1 Mobiilioppiminen

Mobiilioppiminen tarkoittaa yleisesti tietoverkon kautta jaeltavaa oppimissisältöä tai -aktiviteettia, joka mahdollistaa tehokkaan tiedonhankinnan ja oppimateriaalin käsittelemisen mobiililaitetta käyttämällä. Mobiilioppiminen sallii opiskelijan itse päättää, milloin ja missä hän haluaa käydä oppisisältöä läpi. Mobiilioppiminen on yleistynyt viime vuosina mobiililaitteiden ominaisuuksien ja verkkotekniikoiden parantumisen myötä. Mobiilioppimisen kasvu on ollut huomattavaa maailmanlaajuisesti, ja sen suosio on lisääntynyt erityisesti kehittyvissä maissa. (Wexler 2008: 7; Ally 2009.)

Mobiililaitteiden kehitys, niillä käytettävien palveluiden yleistyminen ja käyttäjien tapaa käyttää mobiililaitteita päivittäin ovat saaneet aikaan keskustelua muodollisen opetuksen uudistamisesta enemmän mobiiliteknologiaa hyödyntävään suuntaan (McHaney 2011: 61, 68). Mobiililaitteilla tapahtuva verkkopalveluiden käyttö on tärkeässä asemassa maissa ja alueilla, joissa tietoverkkojen infrastruktuuri on heikosti kehittynyttä. Tällaisissa tapauksissa kiinteät internetyhteydet ja tietokoneet eivät ole välttämättä koskaan ehtineet muodostua pääasialliseksi keinoksi käyttää verkkopalveluita. Mobiililaitteita käytetään suhteellisesti enemmän sekä viihde- että tietosisällön etsintään ja kulutukseen. (Jain 2006: 1.)

MOOC (Massive Open Online Course) on verkko-oppimisen muoto, joka tarkoittaa avointa, suurelle käyttäjäkunnalle tarkoitettua oppimispalvelua. MOOC-muodolle on tyypillistä, että kuka tahansa voi käyttää palvelua (Yuan & Powell 2013: 5). Esimerkkejä MOOC-palveluista on taulukossa 1.

Taulukko 1: Esimerkkejä MOOC-palveluista (Yuan & Powell 2013: 7–8).

Palvelun nimi	Verkko-osoite
edX	https://www.edX.org/
Coursera	https://www.coursera.org/
UDACITY	https://www.udacity.com/
Udemy	https://www.udemy.com/
P2Pu	https://p2pu.org/en/
Khan Academy	https://www.khanacademy.org/

Nimenomaan mobiililaitteille suunniteltuja MOOC-palveluita on vielä verrattain vähän. Tilanne on kuitenkin muuttumassa, ja sisällöntarjoajat ovat alkaneet tuoda oppimateriaaliaan saataville Apple iOS- ja Google Android -laitteille tarkoitettujen ohjelmistojen kautta. Esimerkiksi ilmaista oppimissisältöä tarjoava Khan Academy (khanacademy.org), joka on aiemmin ollut käytettävissä pääasiassa vain tietokoneilla, toi vuoden 2012 loppupuolella sisältönsä käytettäväksi Apple iOS -laitteille (Empson 2012; Thompson 2011).

2.2 Pelillistäminen

Pelillistäminen on erilaisten pelimekanismien, kuten tavoitteiden, mielikuvien ja haasteiden, käyttöä motivaation ja kiintyneisyyden lisäämiseksi (Zichermann & Cunningham 2011: 14; Huotari & Hamari 2012: 20). Pelillistämässä tärkeimmät aspektit ovat mielenkiinnon herättäminen, opiskelijan sitouttaminen, itseopiskelun kannustus ja ryhmäintegraatio. Oppimisen pelillistämisen erikoispiirre on, että vain vapaaehtoisuuteen perustuva pelillistäminen vaikuttaa positiivisesti (Bayliss & Schwartz 2009: 2). Suurin haaste pelillistämässä on, ettei kattavia tutkimuksia ole tehty ja pienimuotoisempien tutkimusten pohjalta ei vielä voida tehdä selkeitä johtopäätöksiä pelillistämisen keinoista (Bayliss & Schwartz 2009: 1).

Tutkimustulosten vähäisen määrän vuoksi käyttäjien houkuttelemista peleillä on myös kritisoitu. Pelien on sanottu vievän huomion itse pääasialta, ja niiden välittömyyden ja suoraviivaisuuden on väitetty tekevän asioista liian helppoja käsitellä. On kuitenkin liian varhaista tehdä johtopäätöksiä, koska rakenteellisesta pelimäisten keinojen hyödyntämisestä ei ole pitkäaikaisia kokemuksia. (LaPointe 2008: 229, 244.)

Tieto- ja viestintäteknisillä tavoilla tehostettu oppimiskokemus voi antaa opiskelijalle omistajuuden tunteen omaan oppimiseen, ja myös tällä on vaikutusta opiskelijan motivaatioon. Vaarana kuitenkin on liika lopputuloskeskeisyys, jolloin painopiste siirtyy pois itse oppimisprosessista. (Järvelä & Häkkinen 2006: 73.)

Pelimäinen lähestymistapa auttaa opiskelijaa pääsemään opintosisällössä eteenpäin ja tuo esille tärkeimmät opiskeltavat asiat ja niiden painopisteet palkitsemisen kautta. Palkinnot ja pisteet kertovat opiskelijan osaamistasosta, ja opiskelija voi itse kokea ansainneensa ne. Toisaalta huonosti toteutettu pelimäinen lähestymistapa voi hämmentää käyttäjiä. Jos käytetyt pelillistämiskeinot eivät tunnu sisällön luonteelta osalta, käyttäjälle voi jäädä kielteinen mielikuva palvelusta. (Raymer 2011; Siering 2012.)

Yksinpelinä tietokonevastustajaa vastaan pelattavat pelit vaativat ohjelmistokehittäjältä taitoa kehittää tietokonevastustajalle uskottavaa tekoälyä. Vastustaja ei saa olla liian vaikea, tai käyttäjä turhautuu. Myös liian helppo vastustaja saa käyttäjän menettämään mielenkiintonsa. Jos kehittäjä ei ole nähnyt vaivaa tietokonevastustajan tekemiseen, käyttäjä ei myöskään koe vastustajan voittamista arvokkaana onnistumisena. Vaikeustason valintaominaisuus helpottaa kehittäjän työtä, ja käyttäjä voi itse löytää itselleen sopivan haasteellisen tietokonevastustajan. Tietokonevastustajan tulee kuitenkin käyttäytyä ihmismäisesti, ja simuloitun vastustajan käyttäytymisen tulee vaikuttaa satunnaiselta. Tietokonevastustajan tulee myös pelata samoilla säännöillä kuin ihmispelaajat. (Adams 2010: 324–326; Habgood & Overmars 2006: 220.)

Yksi pelillistämiskeino on käyttää erilaisia peleistä tuttuja pisteitystapoja. Usein pisteityksen tavoitteena on lisätä käyttäjien halua edetä palvelussa, parantaa käyttäjien sitoutuneisuutta palveluun ja innostaa käyttäjiä käyttämään palvelua myös jatkossa. Pisteitysmalleja voi myös hyödyntää käyttäjien välisen sosiaalisen kanssakäymisen kehittämiseen. Pisteitystavat tulee kuitenkin suunnitella sopimaan kehitettävään konseptiin, jotta pisteiden keräämisestä ei tule ainoa käyttäjien tavoite. (Zichermann & Cunningham 2011: 38.)

Esimerkkejä erilaisista pisteitystavoista on lueteltu taulukossa 2.

Taulukko 2: Erilaisia pisteitystapoja (Zichermann & Cunningham 2011: 38).

Pisteitystapa	Pisteitystavan kuvaus
Kokemuspisteet	Käyttäjän kokemuspisteet karttavat kaikista palvelussa tehdyistä aktiviteeteista. Kokemuspisteillä kannustetaan käyttäjää etene- mään palvelussa ja kokeilemaan uusia asioita.
Lunastettavat pisteet	Lunastettavia pisteitä voi saada kuten kokemus- tai taitopisteitä, mutta niitä voi käyttää myös ”valuuttana” palvelussa.
Taitopisteet	Taitopisteitä karttuu silloin, kun jokin tehtävä tai asia on suoritettu laadukkaasti. Taitopisteillä kannustetaan käyttäjää huolellisuuteen ja taitojen kehittämiseen.
Karmapisteet	Karmapisteitä voi saada kuten kokemus-, taito- ja lunastettavia pisteitä. Käyttäjä voi itse jakaa karmapisteitä toisille käyttäjille tai sisällölle. Karmapisteillä kannustetaan käyttäjää ”äänestämään” tärkeinä pitämiänsä asioita.
Mainepisteet	Mainepisteitä voi saada luottamusta herättävästä käytöksestä. Mainepisteiden on tarkoitus kertoa käyttäjän luotettavuudesta muille käyttäjille.

Saavutukset ovat ennakkoon määriteltyjä ja tiedettyjä tavoitteita, joita käyttäjä voi suorittaa tekemällä asioita onnistuneesti tietyssä järjestyksessä tai tietynlaisina kokonaisuuksina. Saavutuksien suorituksista voi saada esimerkiksi virtuaalisia mitaleita, kunniamerkkejä tai pokaaleja. (Zichermann & Cunningham 2011: 88–89.)

Pelillistämisalustoilla tarkoitetaan järjestelmiä ja palveluita, jotka tarjoavat valmiita mekanismeja pelaajan moninpeliä varten luoman käyttäjätilin hallintaan, sosiaaliseen verkostoitumiseen, pelisaavutusten merkitsemiseen ja seuraamiseen sekä verkkoviestintään. (Herger 2011.)

Valmiita pelillistämiskeinoja tarjoavia alustoja ovat esimerkiksi Playstation Network, Xbox Live ja Valve Steam (Starkey 2013). Nämä palvelut on kehitetty niin sanotuiksi ekosysteemeiksi, joiden kautta käyttäjät voivat hankkia, hallita ja pelata pelejä yhdellä käyttäjätunnuksella. Pelisovelluksen rakentaminen ja pelillistämiskeinojen tehokas käyttö vaatii kuitenkin usein jonkin valmiin pelimoottorin, kuten Unity3D:n käyttämistä, sillä rajapintojen ja verkkoprotokollien uudelleenohjelmointi on työlästä ja vaatii asian-
tuntemusta.

Toisenlaista lähestymistapaa tarjoaa esimerkiksi Badgeville (<http://www.badgeville.com>). Se on pelillistämiseen erikoistunut yritys, joka markkinoi palveluitaan peli- ja mainejärjestelmien nopeaan käyttöönottoon ja hallintaan (Badgeville 2013). Pelillistämiseen Badgeville tarjoaa Behaviour Platform -nimistä alustaa, jonka kautta käyttäjien pisteitä, tasoja, saavutuksia, tehtäviä ja tuloksia voidaan jakaa ja seurata. Alusta myös mahdollistaa ilmoitusten jakelun käyttäjille, kun esimerkiksi uuden tehtävän avautuminen tai saavutuspalkinnon suoritus havaitaan. Behaviour Platform -alustan voi integroida verkkosovellukseen HTTP (Hypertext Transfer Protocol) -protokollalla toimivan REST (Representational State Transfer) -rajapinnan avulla. Alusta tarjoaa myös valmiin JavaScriptillä toimivan ohjelmistorajapinnan.

2.3 Moninpeliympäristöt

Verkossa tapahtuva moninpelaaminen vaatii ympäristön, joka tarjoaa osallistujille mahdollisuuden suhteellisen nopeaan vuorovaikutukseen. Nopeusvaatimukset riippuvat sovellusalueesta. Esimerkiksi hidastempoisessa strategiapelissä alle yhden minuutin vasteaika voi olla riittävä. Toisissa tapauksissa, kuten toimintamoninpeleissä, viiveen tulee olla alle sata millisekuntia, jotta se ei vaikuta pelaamiseen ja pelaajan motivaatioon kielteisesti (Claypool & Claypool 2010: 2, 7).

Vasteajaltaan nopean ja virheensietokyvyltään hyvän moninpeliympäristön kehittäminen vaatii paljon resursseja. Monilla pelien kehittämiseen keskittyvillä yrityksillä on kattava kokemus moninpeliympäristöjen kehittämisestä, ja ne lisensoivatkin kehittämäänsä moninpeliympäristöä muiden yritysten käytettäväksi (Griffiths 2013). Myös pelkäävät moninpeliympäristöjen kehittämiseen erikoistuneita palveluita on olemassa, kuten esimerkiksi Electrotank Electroserver (<http://www.electrotank.com/>) ja SmartFoxServer (<http://www.smartfoxserver.com/>). Niitä käytetään laajasti myös selainpelien moninpeliympäristöinä niiden suoraviivaisen käyttöönoton vuoksi.

3 Pelimäisen verkkosovelluksen suunnittelu

3.1 Vaatimukset

Mobiilioppimiseen tarkoitettujen verkkosovellusten tulee olla ulkoasultaan selkeä ja helposti omaksuttavissa. Verkkosovelluksen rakenteen tulee olla yksinkertainen, jotta navigointi on suoraviivaista mobiililaitteilla, joissa ruudun koko ja erottelukyky asettavat rajoituksia. Www-käyttöliittymän tulee toimia yleisimmissä verkkoselaimissa, ja vaikka pieniä eroja selainten välillä sallitaan, palvelun tulee toimia samalla tavoin eri alustoilla. Pelimäisessä verkkosovelluksessa pelin tulee olla helposti ja nopeasti käynnistettävissä, eikä käyttäjä saa joutua odottamaan pitkiä aikoja. Pisteiden ansaintamallin tulee olla helppotajuinen ja nopeasti ymmärrettävissä.

Valmiita pelillistämiskeinoja tarjoavat ekosysteemialustat (esimerkiksi PlayStation Network, Xbox Live ja Valve Steam) rajoittavat päätelaitekantaa toteuttajayrityksen tai strategisten yhteistyökumppaneiden valmistamiin malleihin tai tiettyihin käyttöjärjestelmiin. Esimerkiksi Xbox Live -alustaa voi käyttää vain natiivikoodilla toteutetuissa peleissä Windows-, Windows Phone- ja Xbox-alustoilla. Tämänkaltaisia päätelaite- ja käyttöjärjestelmäsidonnoisia alustoja ei voi käyttää järjestelmäriippumattoman selainpohjaisen moninpelin toteutukseen.

Käyttäjäkertomukset ovat lyhyitä kuvauksia siitä, mitä sovelluksen tai verkkopalvelun käyttäjän tulee olla mahdollista tehdä. Insinööriyössä toteutettavalle pelimäiselle verkkosovellukselle suunniteltiin seuraavat käyttäjäkertomukset:

- Opiskelija voi rekisteröityä palvelun käyttäjäksi antamalla sähköpostiosoitteensa ja valitsemalla salasanan.
- Opiskelija voi käyttäjäksi rekisteröidytyään kirjautua sisään sähköpostiosoitteellaan ja salasanallaan.
- Käyttäjä voi opiskella ja harjoitella oppimissisältöä itsenäisesti.
- Käyttäjä voi harjoitella moninpelin mekaniikkaa ilman toista ihmisvastustajaa.
- Käyttäjä voi valita, haluaako hän olla toisten käyttäjien haastettavissa.
- Käyttäjä voi saada pisteitä tekemällä harjoituksia.

- Käyttäjä voi nähdä listan toisista online-käyttäjistä ja heidän keräämistään pisteistä.
- Käyttäjä voi haastaa toisen online-käyttäjän moninpeliin.
- Haastettu käyttäjä voi valita, haluaako hän ottaa haasteen vastaan.
- Käyttäjä voi kilpailla toisen käyttäjän kanssa siitä, kumpi ratkaisee annetun tehtävän ensimmäisenä.

Insinööriyössä toteutettavan verkkosovelluksen kehitystyön tulee olla mahdollisimman tehokasta, jotta erilaisia ominaisuuksia voi kehittää ja testata nopealla aikataululla. Kehitystyöhön tulee löytyä helposti ohjelmointiresursseja, joten harvinaisempia tai erityisen nuoria ohjelmointikieliä tai sovellusarkkitehtuureja ei kannata käyttää. Verkkosovelluksen palvelimen tulee olla nopea ja tehokas pystyttä. Verkkopalvelimen asennuksen ja käyttöönoton on hyvä olla vapaita lisenssimaksuista ohjelmistojen osalta, jotta prototyyppien kehitystyöstä ei tule tarpeettomia kustannuksia. Mikäli järjestelmästä myöhemmin kehitetään kaupalliseen käyttöön tarkoitettu palvelu, palvelinarkkitehtuuri voidaan miettiä uudelleen liiketaloudelliselta näkökannalta. Kehitys- ja palvelinympäristön käyttöjärjestelmien ja muiden ohjelmistokomponenttien tulee olla mahdollisimman samankaltaisia, jotta versio- tai alustariippuvaisista eroista ei aiheudu tarpeettomia virhetilanteita.

Toteutettavan verkkosovelluksen sisällön aihealueena tulee olla matematiikka. Tehtävätyypit tulee suunnitella niin, että käyttäjä voi käyttää tehtävien ratkaisussa apunaan kynää, paperia ja laskinta, jos niin haluaa. Järjestelmän tulee näyttää tehtävänanto ja mahdollistaa vastauksen syöttäminen. Työtapa soveltuu matematiikan opetukseen parhaiten silloin, kun käyttäjillä on jo perustieto siitä, missä järjestyksessä tehtävät kannattaa ratkaista. Pelkkien lopullisten vastausten syöttäminen, varsinkin monivalintakysymyksiin vastaamalla, saattaa sisältää riskin tehtävien vääränlaisesta ratkaisutavasta, vaikka lopulliset vastaukset olisivat oikeita. Sisällöntekijän tulee ymmärtää nämä rajoitukset ja varmistaa, että virheellisten ratkaisumallien riski ja arvaamisen hyöty minimoidaan.

Tilatiedon synkronoinnin ja käyttöliittymän tulee toimia vakaasti ja samalla tavoin eri käyttäjillä, joten toteutettava verkkosovellus vaatii modernin verkkoselaimen. Käyttöliittymä suunniteltiin toimimaan parhaiten laitteissa, joiden ruudun erottelukyky on leveys-suunnassa vähintään 320 kuvapistettä ja korkeussuunnassa vähintään 480 kuvapistet-

tä. Päätös tehtiin sillä perusteella, että pienemmän erottelukyvyn mobiililaitteella moninpelahaasteissa tarpeellisia tietoja ei voi näyttää yhdellä kertaa. Tuetuiksi verkkoselaimiksi valittiin yleisimpien mobiilikäyttöjärjestelmien oletusselaimet. Kehitystyö ja kehitystyön aikainen testaus tehtiin Google Chrome -selaimen Windows 7 -versiolla, joten myös se lisättiin tuettujen selainten listaan. Tuetut verkkoselaimet ja niiden versiot on lueteltu taulukossa 3.

Taulukko 3: Verkkoselaimet, joita toteutettavan verkkosovelluksen tulee tukea.

Käyttöjärjestelmä	Selain	Selaimen versio
Apple iOS 6	Mobile Safari	6
Google Android 4	Android Browser	4
Microsoft Windows Phone 7.8	Internet Explorer Mobile	9
Microsoft Windows Phone 8	Internet Explorer Mobile	10
Microsoft Windows 7	Google Chrome	26

3.2 Sisältö ja tehtävät

Mobiilioppimisen sisällön pelillistämistä voidaan lähestyä monesta eri suunnasta. Sisällössä etenemistä voidaan tehdä houkuttelevammaksi pisteittämällä sen läpikäyntiä. Käyttäjien pistemääriä voidaan verrata keskenään erilaisilla mittareilla ja näin saada käyttäjiä pyrkimään parantamaan pistemääriään.

Kilpailullisuutta voidaan korostaa antamalla kahdelle tai useammalle käyttäjälle sama tehtävä. Käyttäjien keskinäinen vastakkainasettelu kannattaa kuitenkin tehdä varoen, sillä liian suorituskeskeinen ja tarpeetonta stressiä aiheuttava kilpailukonsepti voi karkottaa joitain käyttäjiä. (Vorderer ym. 2003: 2, 5.)

Toinen mahdollinen lähestymistapa on lisätä peliin elementtejä, jotka eivät liity oppimissisältöön, mutta joiden kautta pulmia ja niiden ratkaisuja voidaan kehittää. Aktiviteetit ja tehtävyytyypit riippuvat sisällön luonteesta ja aihealueesta.

Yksi yleisimpiä mobiilioppimisessa käytettyjä tehtävyytyyppejä ovat monivalintatehtävät (Jenkins 2012). Monivalintatehtävässä käyttäjälle näytetään kaksi tai useampia vastausvaihtoehtoja, joista käyttäjän tulee valita annettuun kysymykseen oikea vastaus. Jär-

jestelmä voi antaa palautetta vastauksen oikeellisuudesta ja suositellusta ratkaisutavasta. Esimerkki monivalintatehtävästä on esitelty kuvassa 1.

Kysymysteksti

Vastausvaihtoehto A

Vastausvaihtoehto B

Vastausvaihtoehto C

Vastausvaihtoehto D

Kuva 1: Esimerkki monivalintatehtävästä.

Käyttäjälle saattaa tulla houkutus arvata tehtävän vastaus eli valita satunnainen tai ensimmäisenä oikealta näyttävä vastausvaihtoehto, jolloin oppimissisältö ei ole enää tarkoituksenmukainen. Arvaamisen hyötyjä voidaan vähentää suunnittelemalla järjestelmä siten, että vääristä vastauksista rangaistaan. Rangaistuksien käyttämisessä tulee kuitenkin olla varovainen, sillä ne saattavat antaa käyttäjälle kielteisen kuvan palvelusta ja vähentää käyttömotivaatiota. Verkkosovelluksen pisteitysjärjestelmä suunniteltiin siten, että väärän vastauksen antaminen vähentää käyttäjän pisteitä, mutta tehtävään vastaamatta jättäminen ei vaikuta pisteisiin.

Tekstinsyöttötehtävässä opiskelija kirjoittaa mielestään oikean vastauksen sille varattuun kenttään. Vastaus tarkistetaan koneellisesti, ja sen oikeellisuus ilmoitetaan käyttäjälle. Esimerkki tekstinsyöttötehtävästä on esitelty kuvassa 2.

Kysymysteksti

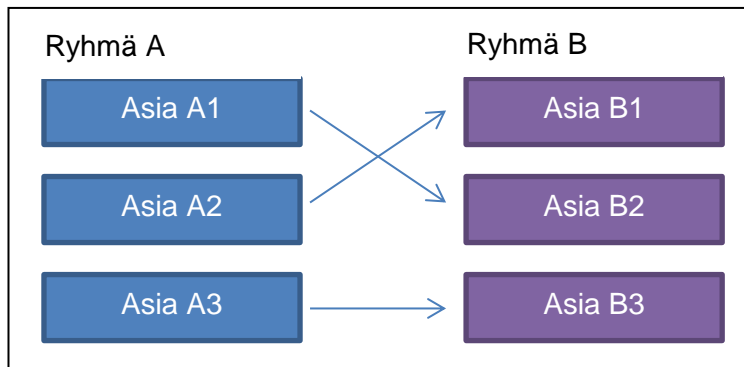
Käyttäjän vastauksen tekstinsyöttökenttä

Lähetä vastaus

Kuva 2: Esimerkki tekstinsyöttötehtävästä.

Tekstinsyöttötehtävän haasteena on matematiikan kirjoittaminen tietokoneella ja mobiililaitteella. Vastausten tulee joko olla yksinkertaisia tai käyttäjät tulee ensin kouluttaa monimutkaisempien kaavojen syöttämiseen niin, että järjestelmä voi tulkita vastauksen. Tekstinsyöttötehtävässä tulee myös huolehtia, että oikean vastauksen voi syöttää vain tietyllä tavalla. (Lindquist ym. 2007.)

Graafinen ryhmittelytehtävä näyttää käyttäjälle kaksi ryhmää, joissa molemmissa on vähintään kaksi asiaa. Käyttäjän on mahdollista yhdistää asia ryhmästä A vastaavaan asiaan ryhmästä B. Osioiden sijoittelu vaatii käyttäjän päätelaitteen näytöltä tarpeeksi suuren erottelukyvyn. Tehtävätyyppejä rajoittavat myös kohdepäätelaitteiden muut rajoitukset, esimerkiksi kosketusnäytön tarkkuus tai kursorinavigoinnin yhteensopivuusongelmat. Esimerkki tekstinsyöttötehtävästä on esitelty kuvassa 3.



Kuva 3: Esimerkki graafisesta yhdistelytehtävästä.

Sovelluksen suunnitteluvaiheessa kävi ilmi, että sekä tekstinsyöttö- että ryhmittelytehtävien tukeminen aiheuttaisi käytettävyyssongelmia. Lisäksi tekstinsyöttötehtävätyypin huomattiin olevan epäyhteensopiva ajastetun tehtävänratkaisun kanssa, sillä suuri osa ajasta kuluu vastauksen kirjoittamiseen. Pääasialliseksi tehtävätyypiksi valittiin monivalintatehtävät.

3.3 Arkkitehtuuri ja rakenne

Yksi nopeaan tuotekehitykseen käytettävistä verkkopalvelimen ohjelmistokehysvaihtoehtoista on Ruby on Rails. Siihen liittyy ohjelmistokirjastojen ja -työkalujen lisäksi myös tapa tehdä kehitystyötä. Ohjelmakoodin tyyli, ohjelmoinnin työnkulku, palvelunhallinta ja julkaisutavat käyttävät hyväksi havaittuja tapoja. Ohjelmistokehittäjä voi myös hyötyä laajasta ja asiantuntevasta kehittäjäyhteisöstä ja hyvästä dokumentaation saatavuudesta.

Insinööriyössä toteutettavan verkkosovelluksen luonne vaatii, että loppukäyttäjien selaimessa suoritetaan ohjelmakoodia tilapäivitysten hakua ja ilmoitusten näyttämistä varten. JavaScript-kielen käyttö on ainoa vaihtoehto silloin, kun verkkosovelluksen tulee toimia yleisimmissä mobiiliselaimissa. JavaScript-toimintojen ohjelmoinnin nopeuttamiseksi voi käyttää valmiita ohjelmistokirjastoja ja -kehyskiä. Esimerkiksi JQuery- ja Prototype-kirjastot ovat yleisesti käytettyjä ja helppoja ottaa käyttöön riippumatta tustajärjestelmän arkkitehtuurista. Myös kattavampia JavaScript-ohjelmistokehyskiä voi käyttää vaadittujen toimintojen toteuttamiseen. Niistä esimerkiksi AngularJS- ja Backbone.js-kehysten käyttö yleistyy nopeasti. Kokonaisten ohjelmistokehysten käyttöä puoltaa mahdollisuus, että prototyypituotantoa ja uusien ominaisuuksien jatkokehitystä voi nopeuttaa kehysten tehokkailla tiedon yhdistämistoiminnoilla palvelimien, selainkoodin ja käyttöliittymän välillä. Haittapuolena on kuitenkin verkkosovelluksen arkkitehtuurin lukittautuminen jo varhaisessa vaiheessa käyttämään tiettyä ohjelmistoarkkitehtuuria. Ohjelmistokehysten käyttöön voidaan siirtyä myös myöhemmin, ja kannattavinta lähestymistapaa voi pohtia tarkemmin jatkokehitystarpeita suunniteltaessa.

Mahdollisia käyttöjärjestelmävaihtoehtoja ovat Windows, Linux ja Apple Mac OS X. Näistä ainoastaan Linux on vapaa lisenssimaksuista. Linux-käyttöjärjestelmällä yleisimmät www-palvelinohjelmistot ovat Apache ja nginx. Molempia käytetään sekä staattisen että dynaamisen www-sisällön tarjoamiseen. Tämän lisäksi tarjolla on myös monia ohjelmointikieli- ja ohjelmistokehyskohtaisia www-palvelinvaihtoehtoja. Niitä käytetään usein kehitysympäristöissä, joissa palvelinkapasiteetin tai suorituskyvyn optimoinnille ei ole tarvetta.

Tietokanta

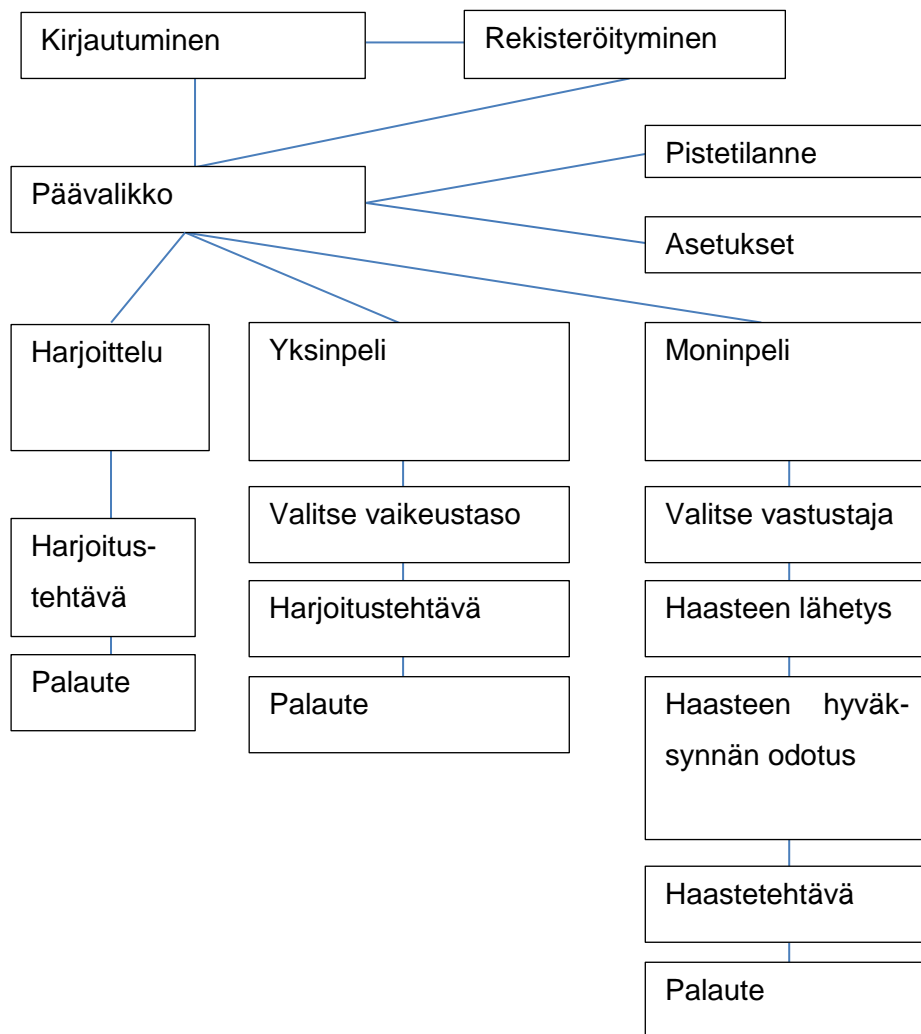
Tietokanta suunniteltiin pitämään yllä käyttäjä-, peli- ja pistetietoja. Lisäksi tietokannan tehtävänä on tarjota tehokas tapa tilapäivitysten ja haastekutsujen tilapäiselle tallennukselle. Myös itse oppimissisältönä käytettävät harjoitukset suunniteltiin tallennettavaksi tietokantaan.

Tietokanta ei ole välttämätön kaikkien tilapäivitysviestien välittämiseen. Kolmannen osapuolen suoraviestipalveluita käytettäessä osa reaaliaikatoiminnoista voidaan toteuttaa siten, että niistä ei jää pysyvää tallennusmerkintää tietokantaan. Kaikki toimintoon liittyvä tieto voidaan kuljettaa viestin mukana, jolloin vain käyttäjän lopullinen toiminta –

esimerkiksi uuden pelin haasteeseen vastaaminen – aiheuttaa sivulatauspyynnön verkkosovelluksen palvelimella.

Käyttöliittymän rakenne

Insinööriyössä toteutettavan verkkosovelluksen käyttöliittymän tulee mahdollistaa vaatimuksissa sivulla 7 luetellut käyttötapaukset, mutta olla silti tarpeeksi yksinkertainen mobiiliselaimilla tapahtuvaan käyttöön. Käyttöliittymän suunnitellun rakenteen voi nähdä kokonaisuutena sivukartassa kuvassa 4.



Kuva 4: Insinööriyössä toteutettavan verkkosovelluksen sivukartta.

Moninpeliin pääsee kahta mahdollista kautta: lähettämällä haasteen ja hyväksymällä vastaanotetun haasteen. Käyttäjä voi vastaanottaa haastepyyntöjä kaikissa muissa

paitsi haastetehtäväosiossa. Haasteen hyväksymällä käyttäjä siirtyy suoraan haaste-tehtävään.

3.4 Tilatiedon synkronointi

Kun kaksi tai useampi käyttäjää vaikuttaa toistensa tilaan, tilatiedon pitämisellä synkronoituna on tärkeä rooli. Mikäli synkronointi menetetään tai se on epätarkkaa, järjestelmä muuttuu epävakaaksi ja käyttävät kokevat viiveitä ja toiminnallisuuksien epäloogisuutta. Selainpohjaisessa järjestelmässä yksi tapa pitää tilatietoa synkronoituna on käyttää kiertokyselyitä, joissa käyttäjien selaimet kysyvät palvelimelta tilapäivityksiä säännöllisesti. Tästä seuraa kuitenkin tehottomuutta ja tarpeetonta räsitusta palvelimelle, sillä HTTP-protokollaa ei ole optimoitu tällaiseen käyttötarkoitukseen.

Vakaampi tapa on käyttää HTML5 (Hypertext Markup Language 5) -määrittelyn mukaista WebSocket API (Application Programming Interface) -tapaa. WebSocket API tarjoaa mahdollisuuden nopeampaan kaksisuuntaiseen tiedonsiirtoon, ja se käyttää vähemmän palvelin- ja verkkokapasiteettia (Shankar 2012: 293). WebSocket API:n käytön haittana on, että se vaatii selaimelta standardinmukaisen tuen. Selaimilla, joista ominaisuus puuttuu tai se on toteutettu standardista poikkeavalla tavalla, ei palvelua voi käyttää (Rettig 2012: 408).

WebSocket API -tapaa voi käyttää esimerkiksi Node.js-ohjelmistokomponentin avulla. Ohjelmistokomponentin käyttö vaatii ohjelmistokirjastojen tai -komponenttien asennusta sekä kehitys- että palvelinympäristöön (Makzan 2011: 236–238). Node.js-ohjelmistokomponentin tehokas käyttö vaatii toimintojen ohjelmoimisen asynkronisella tavalla (Burnham 2012: 12, 48–49). Asennettavien komponenttien lisääntyessä järjestelmä voi monimutkaistua jo kehitystyön alkuvaiheessa. Paremmasta suorituskyvystä voi hyötyä myöhemmin tuotantoympäristössä, mutta prototyyppisovellusta kehitettäessä voi viiveiden välttämiseksi olla järkevämpää pitäytyä arkkitehtuuriltaan mahdollisimman yksinkertaisessa ratkaisussa.

Verkkoselaimessa toimiva moninpelijärjestelmä tekee tilatiedon päivitysten vuoksi tietokantakyselyitä useammin kuin keskiverto verkkopalvelu. Kiertokyselyillä toteutetut tilapäivitykset vaativat lyhyen tietokantakyselyn jokaisen päivityspyynnön yhteydessä. Tilapäivitysten tarpeen määrä kasvaa yhtäaikaisten käyttäjien määrän mukana. Tilapäivitysten tietokantakyselyiden määrä saattaa ruuhkauttaa tietokantapalvelimen ja

hidastaa palvelun muuta käyttöä, myös tapauksissa, joissa vastauksen sisältö on muuten tyhjä. Tilapäivitysten aikavälin ja käyttäjämäärän vaikutus tietokantakyselyiden määrään on esitetty taulukossa 4.

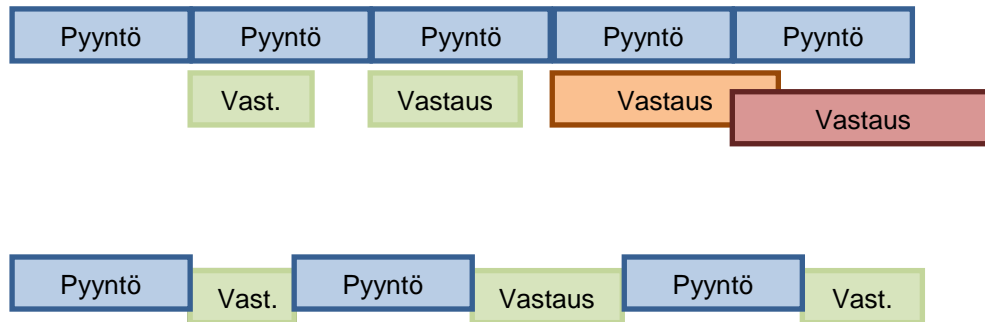
Taulukko 4: Online-käyttäjien määrän ja tilapäivityspyyntöjen aikavälin vaikutus tietokantakyselyjen vähimmäismäärään.

Online-käyttäjä	Tilapäivitysväli (s)	Tietokantakyselyitä vähintään (1/s)
1	20	0,05
1	10	0,10
1	2	0,50
10	20	0,50
10	10	1
10	2	5
100	20	5
100	10	10
100	2	50
1000	20	50
1000	10	100
1000	2	500

On huomioitavaa, että näissä esimerkissä on otettu huomioon vain tapaukset, joissa käyttäjät tekevät tilapäivityskyselyitä vain omien tilapäivitystensä hakuun, esimerkiksi vastaanotettaviin haasteilmoituksiin. Verkkopalvelun tietyissä osioissa järjestelmän tulee tehdä tilapäivityskyselyitä myös muiden käyttäjien tiedoista, esimerkiksi toisen käyttäjän online-tilan ja harjoitustehtävän tilan selvittämiseksi. Toisaalta verkkopalvelussa on osioita, joissa tietyn tyyppisten tilapäivityskyselyiden taajuutta voi pienentää käytettävyyden siitä kärsimättä.

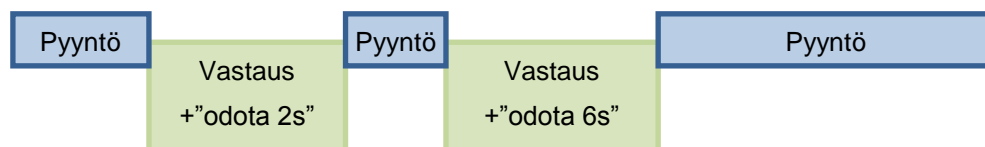
Tilapäivityskyselyitä ei kannata tehdä vakiotajuudella, koska kyselyn vastausaika vaihtelee. Mikäli vastausaika venyy pidemmäksi kuin kyselyn odotusaika, järjestelmä tekee seuraavan kyselyn, vaikka edellinen kysely on vielä käynnissä. Tästä seuraa päällekkäisiä kyselyitä ja vastausaikojen jatkuvaa pitenemistä, mikä johtaa loppujen

lopuksi palvelun kaatumiseen. Odotusajastin uuden kyselyn lähettämistä varten kannattaa käynnistää vasta, kun edelliseen kyselyyn on saatu vastaus. Todelliset kyselyiden aikavälit vaihtelevat riippuen kyselyn keston ja seuraavan kyselyn odottamisajan summasta. Kuvassa 5 on havainnollistettu vakiotaajuudella tehdyn tavan ja vastausajan huomioonottavan tavan eroja.



Kuva 5: Vakiotaajuudella tehtyjen ja vastausajat huomioonottavien päivityspyyntöjen vaikutus.

Tilapäivityskyselyissä voi käyttää myös palvelinkapasiteettiin ja online-käyttäjien hetkitäiseen määrään mukautuvaa tapaa. Tässä tapauksessa ensimmäinen tilapäivityskysely tehdään heti käyttäjän kirjaututtua sisään. Palvelin lähettää vastauksen tavalliseen tapaan, mutta liittää vastaukseen tiedon siitä, milloin käyttäjän selaimessa ajettava ohjelmakoodi saa tehdä seuraavan tilapäivityspyynnön. Mikäli online-käyttäjien hetkellinen määrä on vähäinen, palvelin voi määrätä tilapäivitykset hyvin usein tapahtuviksi. Käyttäjämäärän kasvaessa tilapäivitystaajuutta voi pienentää, jotta palvelinkapasiteetin riittävyys voidaan varmistaa. Käytettävyys heikkenee käyttäjämäärän kasvaessa, mutta vakavilta virhetilanteilta voidaan välttyä. Tapaa on havainnollistettu kuvassa 6.



Kuva 6: Mukautuva tilapäivitysten odotusaika.

Tämä mahdollistaa myös tilapäivityspyyntöjen priorisoinnin käyttäjätyyppin, käyttäjän senhetkisen toiminnan ja tilapäivityksen tyyppin perusteella. Kyselyt esimerkiksi käyttäjien online-tilasta voidaan tehdä eri tahdilla eri maantieteellisillä alueilla sijaitseville käyttäjille.

Verkkosovellus voi hyödyntää tilaviestien välittämiseen myös valmiita kolmannen osapuolen suoraviestipalveluita. Yksi tällainen palvelu on Pusher.com (<http://www.pusher.com>). Käyttäjien toisilleen lähettämiä haasteviestejä voi lähettää ja seurata Pusher.com-verkkopalvelun kautta, jolloin palvelinkapasiteettia tarvitse käyttää jatkuvaan uusien haasteviestien tarkistamiseen. Pusher.com tarjoaa valmiit rajapintamääritykset suoraan selaimella tehtäviä JavaScript-pohjaisia pyyntöjä varten, ja sillä voi välittää viestejä sekä palvelimen ja käyttäjän välillä että suoraan käyttäjien välillä (Leggetter 2012). Pusher.com-palvelu tarjoaa myös ohjelmakoodiesimerkkejä ja toteutusmalleja palvelinsovelluksia varten esimerkiksi PHP- ja Ruby-ohjelmointikielillä.

3.5 Ajastettu tehtävänratkaisu

Matematiikan kouluopetuksessa harjoitustehtävät ovat perinteisesti laskuja, jotka tulee ratkaista käyttäen paperia ja kynää. Yleensä matematiikan opetuksessa laskujen ratkaisemiselle ei anneta selkeätä aikarajaa, vaan opiskelijoiden annetaan löytää ratkaisut omaan tahtiin. Koetilanteissa on kuitenkin asetettu kokonaisaikaraja, ja esimerkiksi kuudelle tehtävän ratkaisemiselle voidaan antaa kaksi tuntia aikaa. Opiskelijoita kannustetaan tehtävien harjoitteluun siten, että niiden ratkaisu onnistuu tietyssä ajassa.

Verkko-oppimiskäytännöissä käyttäjiä voidaan kannustaa ratkaisemaan tehtäviä tietyssä ajassa käyttämällä ajastettua tehtävänratkaisua. Tehtävään käytetty aika voidaan näyttää käyttäjille, ja tehtävänratkaisulle voidaan antaa tarkka aikaraja. Mikäli tehtävänratkaisuun käytetty aika näytetään tehtävän aikana, käyttäjät saattavat ihmetellä, miksi aikaa mitataan. Mikäli aikaraja ei ole tehtävänratkaisussa kriittinen, asian tulee ilmetä aikalaskurin näyttötavassa tai alustavissa ohjeistuksissa. Toisaalta käyttäjät näkevät aikarajan vaikutuksen ensimmäisessä tehtävässä, jolloin he seuraavaa tehtävää aloittaessaan tietävät, miten aikalaskenta vaikuttaa lopputulokseen. Tarkan aikarajan asettamisella voidaan kertoa, että tehtävä tulisi ratkaista annetussa ajassa tai nopeammin. Kun aikaraja on asetettu oikein, käyttäjillä on tarpeeksi aikaa ratkaista tehtävä, mutta he eivät jää tehtävään ”jumiin” pidemmäksi aikaa.

Moninpelitehtävissä tarkan aikarajan käytöllä voidaan varmistaa, että pelit eivät kestä tätä aikarajaa kauempaa ja käyttäjä vapautuu seuraavaa peliä varten. Peli voi myös päättyä ennenaikaisesti, mikäli molemmat käyttäjät antavat vastauksensa ennen tehtävään varatun ajan loppumista. Aikarajan ja aikalaskurin käyttäminen tuo mukanaan

joitakin teknisiä haasteita. Esimerkiksi selaimella toteutetut aikalaskurit vaativat JavaScript-toimintojen käyttöä, eikä joitain yleisiä mobiiliselaimia voida täten tukea. Käyttäjien selainten ja palvelimien ajastimien tulee toimia synkronisesti. Myös verkon tiedonsiirtoviiveet tulee ottaa huomioon. Normaalitylanteessakin sivulataus saattaa kestää useita sekunteja, mikäli käyttäjällä on hidas tiedonsiirtoyhteys tai hän on huonolla verkon kuuluvuusalueella. Järjestelmän tulee ladata sivuresurssit valmiiksi aina, kun mahdollista, ennen ajastimien käynnistämistä.

3.6 Pisteitys

Insinööriyössä suunnitellussa pelillistämiskonseptissa pääasialliseksi pisteitysmalliksi valittiin haastepisteitys. Haastepisteitysmallissa taitopisteet ja lunastettavien pisteet yhdistettiin yhteen pistemäärään. Haastepisteet kertyvät kahdella eri tavoin: harjoitustehtävien oikeista vastauksista ja toisia käyttäjiä vastaan pelattujen haastepelien voitoista. Haastepisteitä voi myös menettää vastaamalla väärin haastepelien harjoitustehtäviin. Käyttäjä ei menetä pisteitä haastepelin häviöstä. Haastepisteille on ominaista, että vähemmän pisteitä hankkineet käyttäjät saavat suhteessa enemmän pisteitä haastepelien voitoista. Kun opiskelija rekisteröityy verkkopalvelun käyttäjäksi, hänellä ei ole yhtään haastepistettä. Uudet käyttäjät voivat hankkia haastepisteitä vain osallistumalla haasteisiin. Koska haastepisteitä voi menettää vain vastaamalla väärin, käyttäjät voivat kartuttaa pistesaldoaan alussa etenemällä haastetehtävissä varovaisesti.

Toinen pelillistämiskonseptiin valittu pisteitysmalli oli kokemuspisteitys. Se on kuitenkin käytössä vain tietyissä sovelluksen osioissa. Kerätyt kokemuspisteet säilyvät, eikä niitä voi menettää vääristä vastauksista, toisin kuin haastepisteitä.

4 Pelimäisen verkkosovelluksen toteutus

4.1 Työtavat ja komponentit

Insinööriyössä suunniteltu verkkosovellus toteutettiin käyttämällä Ruby on Rails -ohjelmistokehystä. Kehitystyö tehtiin Oracle VirtualBox -ympäristössä virtualisoidulla Ubuntu Linux 12.04 LTS -käyttöjärjestelmällä. Ruby on Rails -ympäristö asennettiin rvm-apuohjelmistolla ja Bundler-paketinhallintaohjelmistolla. Ohjelmointiin käytettiin Sublime Text -editoria Unity-työpöytäympäristössä. Git-versionhallintaa, vi-editoria ja Rails-hallintatyökaluja käytettiin komentoriviympäristössä Terminal-pääteohjelman kautta.

Ruby-ohjelmointiympäristön versioksi valittiin 2.0. Ruby on Rails -versioksi valittiin 3.2. Kehityksenaikaisena www-palvelimena käytettiin Unicorn-palvelinta. Kehitysympäristön tietokannoiksi valittiin MySQL5- ja SQLite3-tietokannat niiden yleisyyden ja hallinnan yksinkertaisuuden vuoksi. Tietokantayhteys toteutettiin Ruby on Rails -kehiksen ActiveRecord-mallilla. Mallit, ohjainkomponentit ja näkymät jaettiin erilleen MVC (Model-View-Controller) -arkkitehtuurin käytännön mukaan. Näkymät toteutettiin html.erb-tiedostoilla, ja ulkoasuun käytettiin CSS (Cascading Style Sheets) -tyylitiedostoja. Sisältönä käytettiin Bitville Oy:n itse tuottamia matemaattisia monivalintatehtäviä. Sisältö syötettiin tietokantaan Ruby on Rails -kehiksen rake-työkalulla käyttäen seeds.rb-tiedostoa ja komentoa "rake db:seed".

Sovellukseen asennettiin joitakin Ruby on Rails -kehiksen lisäosia. Lisäosat asennettiin Ruby-ohjelmointikielen mukana tulleella Gem-hallintaohjelmistolla ja Rails-sovelluskehiksen mukana tulleella Bundler-paketointiohjelmistolla. Asennetut lisäosat on lueteltu taulukossa 5.

Taulukko 5: Toteutetussa verkkosovelluksessa käytetyt Gem-lisäosat.

Lisäosan nimi	Lisäosan kuvaus
Better Errors	Mahdollistaa lähemmän ohjelmistovirheiden tarkastelun niiden sattuessa.
Airbrake	Mahdollistaa tuotantoympäristön ohjelmistovirheiden raportoinnin ja niiden tarkastelun ulkopuolisessa verkkopalvelussa verkkoselaimen kautta.
Devise	Sisältää valmiita komponentteja käyttäjien kirjautumisen hallintaan.
ExecJS, The Ruby Racer	Tarjoavat paikallisen JavaScript-ympäristön ilman tarvetta asentaa kokonaista palvelinohjelmistokomponenttia.
jQuery-rails	Tarjoaa valmiin JQuery-kirjaston, jota käyttäjien selaimet käyttävät ruudun päivitykseen ja taustatiedonsiirtoon.
Pusher	Tarjoaa mahdollisuuden lähettää tilapäivitysviestejä palvelimelta käyttäjille valmiin kolmannen osapuolen Pusher.com-palvelun kautta.

Sovellus vaatii kehitys- ja palvelinympäristön käyttöjärjestelmältä JavaScript-tuen. Tuki toteutettiin uuden sovelluksen luonnin jälkeen asentamalla ExecJS- ja The Ruby Racer -lisäosat. JavaScript-tuen toteuttamiseen voi käyttää myös node.js-ohjelmistokomponenttia.

4.2 Kirjautuminen ja rekisteröityminen

Insinööriyössä toteutettuun verkkosovellukseen pääsee navigoimalla verkkoselaimella sovelluksen verkko-osoitteeseen. Kävijälle näytetään sisäänkirjautumissivu (kuva 7), josta rekisteröityneet käyttäjät voivat kirjautua sisään käyttäjänimellään tai sähköpostiosoitteellaan.

Palvelun nimi

Kuva 7: Toteutetun verkkosovelluksen kirjautumissivu.

Mikäli kävijä ei ole vielä rekisteröitynyt, hän voi luoda uuden käyttäjätunnuksen valitsemalla rekisteröitymislinkin, antamalla sähköpostiosoitteensa ja valitsemalla salasanan (kuva 8). Käyttäjä voi myös valita itselleen käyttäjänimen. Käyttäjän antamaan sähköpostiosoitteeseen lähetetään sähköpostin varmistusviesti, jolla hänen tulee vahvistaa sähköpostiosoitteensa aitous kahden päivän kuluessa rekisteröitymisestä.

Rekisteröidy [Takaisin](#)

Kuva 8: Toteutetun verkkosovelluksen rekisteröitymissivu.

Sisäänkirjautumisen jälkeen käyttäjä näkee päävalikon. Päävalikosta käyttäjä pääsee harjoitteluun, yksinpeliin, moninpeliin, pistetietosivulle ja asetussivulle. Kun käyttäjä saapuu päävalikkoon ensimmäisen kerran, häneltä kysytään, haluaako hän osallistua haasteisiin. Kysymykseen voi valita vastauksen ”Ei, haluan harjoitella vain yksin” tai ”Kyllä, haluan osallista moninpelihaasteisiin”. Valinta toimii yksityisyysasetuksena, jolla käyttäjä voi itse hallita näkyvyyttään palvelussa. Asetusta voi muuttaa myöhemmin omien tietojen muokkaussivulta.

4.3 Ominaisuudet

Harjoittelu

Harjoittelussa (kuva 9) käyttäjä pääsee tekemään harjoituksia yksin ja tutustumaan palvelun sisältöön. Käyttäjälle esitetään matemaattinen monivalintatyypinen harjoitus-tehtävä. Tehtävän ratkaisuun ei ole aikarajaa. Kun käyttäjä on antanut vastauksen, oikea vastaus ja ratkaisumalli näytetään.

Vapaa harjoittelu	Takaisin
Mikä on x :n arvo, kun $x^2+4 = 29$	
<input type="radio"/> 3	
<input type="radio"/> 4	
<input checked="" type="radio"/> 5 ← <i>Vastauksesi oli oikein!</i>	
<input type="radio"/> 6	
Malliratkaisu:	
$x^2+4 = 29$	
$x^2 = 29-4$	
$x = \text{sqrt}(29-4)$	
$x = 5$	

Kuva 9: Harjoittelu toteutetussa verkkosovelluksessa.

Harjoitustehtävä on monivalintatyypinen kysymys. Mikäli käyttäjä vastaa kysymykseen oikein, hän pääsee tehtävän läpi ja ohjataan palautesivulle. Mikäli käyttäjä vastaa väärin, hän epäonnistuu tehtävässä. Palautesivu kertoo oikean vastauksen ja tehtävään käytetyn ajan ja näyttää malliratkaisun. Käyttäjä voi valita palautesivulta, haluaako hän yrittää uudestaan vai palata takaisin valikkoon.

Vapaassa harjoittelussa oikeat vastaukset eivät kerrytä käyttäjän pisteitä, mutta hän voi kerätä suoritusmerkkejä vapaan harjoittelun tehtävien eri aihealueista.

Moninpeli

Monipelissä käyttäjä voi haastaa toisia käyttäjiä tekemään aikarajoitettuja harjoituksia. Käyttäjälle näytetään lista muista online-käyttäjistä, jotka ovat sallineet näkyvyytensä palvelussa. Käyttäjä voi lähettää moninpelihaasteen kenelle tahansa listassa näkyvistä käyttäjistä.

Mikäli käyttäjä ei ole sallinut haasteiden vastaanottamista, hän näkee listan online-käyttäjistä, mutta ei voi lähettää haasteita. Mikäli käyttäjä kuitenkin haluaa lähettää haasteen, hänelle tarjotaan mahdollisuutta muuttaa asetustaan ilman tarvetta siirtyä asetussivulle.

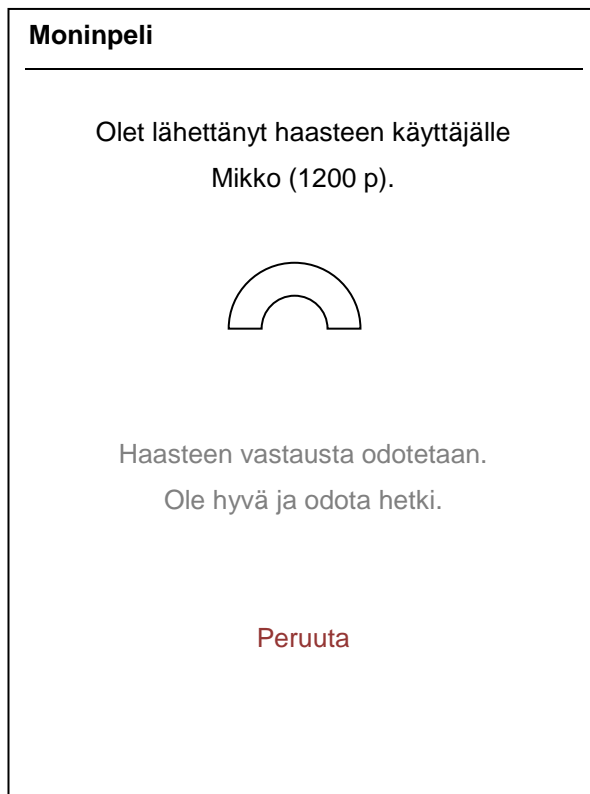
Haastettavien käyttäjien lista (kuva 10) näyttää kunkin käyttäjän nykyisen haastepisteiden määrän. Käyttäjä voi päätellä pistemäärästä, kuinka kokenut ja vaikea pelivastustaja toinen käyttäjä on. Lista näyttää käyttäjiä, joiden pistemäärät ovat hieman yli tai alle käyttäjän oman pistemäärän. Käyttäjälle tarjotaan kuitenkin mahdollisuus navigoida listassa ja valita myös käyttäjiä, joihin pistemäärän ero on suuri.

Valitse vastustaja	Takaisin
△	
Liisa	1650 p
Mikko	1200 p
Minä	900 p
Pete	650 p
Väiski	400 p
▽	

Kuva 10: Haastettavien käyttäjien lista.

Kokeneemman käyttäjän ei kannata haastaa vähemmän pisteitä hankkineita käyttäjiä, sillä haastepeleissä voittaja saa lisäpisteitä vain, mikäli hänen kokonaispistemääränsä oli ennen pelin aloitusta pienempi. Lisäpisteitä saa sitä enemmän, mitä suurempi pisteiden erotus on. Tämä tasoittaa käyttäjien välisiä eroja ja kannustaa käyttäjiä pyrkimään listassa ylöspäin.

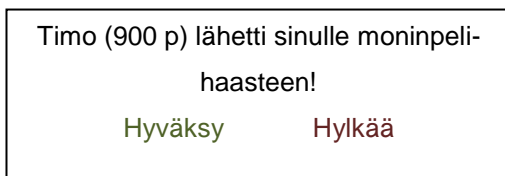
Kun käyttäjä valitsee haastettavan käyttäjän listasta, haaste lähetetään. Käyttäjälle näytetään sivu haasteen vastauksen odotusta varten (kuva 11). Tässä vaiheessa käyttäjä voi tulla toisiin ajatuksiin ja perua haasteen. Haasteen lähettämisen peruuttaminen palauttaa käyttäjän takaisin edellisen sivun käyttäjälistaan ja poistaa haasteilmoituksen haastetulta käyttäjältä. Sivun pitää yllä tietoa toisen käyttäjän tilasta ja ilmoittaa, mikäli toinen käyttäjä hylkää haasteen, ei vastaa siihen ajoissa tai poistuu palvelusta.



Kuva 11: Lähetetyn haasteen odotussivu.

Toinen käyttäjä voi olla parhaillaan pelaamassa moninpeliä jotakuta muuta pelaajaa vastaan. Palvelun mielekkyyden kannalta uusien haastepyyntöjen lähettämistä ei kannata sallia, kun toisella osapuolella on aiempi peli vielä kesken. Jos toinen peli on vielä käynnissä, haasteen vastauksen odotussivulla näytetään ilmoitus, että käyttäjän tulee odottaa hetki toisen pelin loppumista. Tämän vuoksi haastepelien maksimikesto tulee määritellä lyhyeksi, ettei odotusaika muodostu liian pitkäksi.

Kun käyttäjälle on lähetetty haaste, se näytetään ilmoituksena ruudun yläosassa (kuva 12). Ilmoitus pysyy ruudun yläosassa riippumatta sivun vierityksestä. Haasteilmoitus näytetään riippumatta siitä, missä palvelun osiossa hän tällä hetkellä on. Ainoa poikkeus tähän on se, että haasteilmoitusta ei näytetä, mikäli käyttäjällä on toinen haastepeli käynnissä.



Kuva 12: Ilmoitus vastaanotettavasta haasteesta.

Kutsu näytetään vain sen aikaa, kun haasteen lähettäjä on odotusruudullaan. Haasteilmoitus poistuu näkyvistä, jos haasteen lähettäjä poistuu odotusruudultaan. Ilmoitus poistuu näkyvistä myös automaattisesti yhden minuutin kuluttua, jos siihen ei vastata. Haasteilmoituksen näyttämiseen käytetään JavaScript-toimintoa, joka siirtää ilmoituksen ruudulla siten, että se on aina näkyvässä riippumatta sivun vierityksestä.

Jos haastekutsun vastaanottamisessa ja siihen vastaamisessa tulee paljon viivettä, haastajan odotusaika kasvaa epämieliseksi. Tämän vuoksi on tärkeää näyttää haastekutsu heti, kun se on mahdollista. Palvelu käyttää välittömän haastekutsun lähettämiseen Pusher-lisäosaa. Lisäksi käyttäjän selain tarkistaa tilapäivitykset palvelusta viiden sekunnin välein. Tekniikka on toimiva, kun käyttäjämäärä on suhteellisen pieni, alle 100 käyttäjää. Suuremmilla yhtäaikailla käyttäjämäärillä tilapäivitysten aikaväliä tulee pidentää. Palvelun järjestelmänvalvoja voi tehdä asetusmuutoksen palvelimella.

Tilapäivitysten tarkistus on toteutettu JavaScriptillä. Taustapäivitys ei ole arkkitehtuurin puolesta tuettuna kaikissa verkkoselaimissa (esimerkiksi Opera Mini -selaimessa ja Nokia Xpress browser -selaimessa), joten päätelaitekanta ja tuetut selaimet on rajoitettu sellaisiin, joissa taustapäivitys on mahdollista toteuttaa.

Kun toinen käyttäjä hyväksyy haasteen, sekä haastaja että haastettu käyttäjä näkevät ilmoitukset alkavasta pelistä. Ilmoitus valmistaa käyttäjät alkavaan peliin, ja sitä näytetään noin 10 sekunnin ajan. Ilmoituksen aikana palvelin valmistele tehtävän ja sen näkymä ladataan valmiiksi käyttäjien selaimiin. Mikäli sisällön lataus toiselle käyttäjistä kestää kauemmin, ilmoitus pidetään näkyvässä, kunnes kaikki näkymän resurssit on ladattu. Kun molemmat käyttäjät ovat saaneet tehtävän ladattua ja ilmoituksen aika umpeutuu, peli alkaa ja käyttäjille näytetään itse tehtävä.

Haastetehtävä

Molemmille pelissä oleville käyttäjille näytetään matemaattinen tehtävä yhtäaikaaisesti. Tehtävä koostuu kysymystekstistä (tehtävänanto) ja neljästä vastausvaihtoehdosta. Kun tehtävä näytetään, ruudun yläreunassa näytettävä aikalaskuri käynnistetään (kuva 13). Laskurin sekuntimäärä riippuu tehtävän tekijän valinnasta, mutta on oletusarvoisesti 60 sekuntia.

Haastetehtävä käyttäjää
Timo (900 p) vastaan

Aikaa jäljellä: 59 s

Mikä on x :n arvo, kun
 $x^2+4 = 29$

3
 4
 5
 6

Kuva 13: Haastetehtävän aloitus.

Jos käyttäjä antaa vastauksen ensimmäisenä, hänen tulee odottaa, että toinen käyttäjä on myös antanut vastauksensa. Mikäli toinen käyttäjä antaa vastauksen ensimmäisenä, käyttäjälle näytetään ilmoitus, jossa toisen käyttäjän kerrotaan vastanneen, mutta vastauksen oikeellisuutta ei vielä kerrota (kuva 14).

Haastetehtävä käyttäjää
Timo (900 p) vastaan

Aikaa jäljellä: 48 s
Timo vastasi ensimmäisenä (10 s)...

Mikä on x:n arvo, kun
 $x^2+4 = 29$

3
 4
 5
 6

Kuva 14: Haastetehtävä, kun toinen käyttäjä on vastannut.

Tehtävän suoritus päättyy, kun molemmat käyttäjät ovat antaneet vastauksensa tai kun laskuri pääsee nolnaan. Tehtävän päätyttyä käyttäjät ohjataan palauteruutuun.

Haastetehtävän palauteruutu (kuva 15) kertoo käyttäjille molempien käyttäjien valitsemat vastaukset ja niiden oikeellisuuden. Myös molempien käyttäjien pistemäärät näytetään.

Haastetehtävä käyttäjää
Timo (900 p) vastaan

Peli on päättynyt.
Menetit 5 haastepistettä. Timo vastasi ensimmäisenä ja sai 20 haastepistettä.

Mikä on x:n arvo, kun
 $x^2+4 = 29$

3
 4 <- Vastauksesi oli väärin!
 5 <- Timo vastasi oikein!
 6

Malliratkaisu:
 $x^2+4 = 29$
 $x^2 = 29-4$
 $x = \sqrt{29-4}$
 $x = 5$

Kuva 15: Haastetehtävä, kun peli on päättynyt.

Tehtävän suorituksen aikana käyttäjälle ei voi lähettää haastekutsuja. Mikäli haastekutsuja silti lähetetään esimerkiksi synkronointivirheen vuoksi, niitä ei näytetä käyttäjälle ennen haastepelin päättymistä. Kun haastetehtävä on päättynyt, käyttäjälle voi taas lähettää uusia haastekutsuja. Onkin mahdollista, että käyttäjä on niin suosittu vastapelelaaja, että hän näkee uuden haastekutsun heti edellisen pelin päätyttyä. Palauteruutu on mahdollisimman yksinkertainen, jotta sen ehtii käydä läpi ennen kuin uusi mahdollinen haastekutsu ehtii vanhentua.

Haastetehtävän voittaja on se, joka on antanut tehtävään oikean vastauksen. Mikäli molemmat käyttäjät ovat antaneet oikean vastauksen, voittaja on se, joka antoi sen ensimmäisenä.

Mikäli käyttäjä vastaa väärin, hän on haastetehtävän häviäjä. Mikäli molemmat käyttäjät vastaavat väärin, ensimmäiseksi vastannut käyttäjä on haastetehtävän häviäjä. Ta-

sapeli voi sattua vain tilanteessa, jossa kumpikaan käyttäjä ei anna vastausta. Tällöin pisteitä ei lasketa ollenkaan eikä suoritettu haastetehtävä vaikuta kummankaan käyttäjän kokonaispisteisiin lainkaan.

Pisteet lasketaan erikseen vastauksien oikeellisuudesta ja haastetehtävän voitosta. Oikeasta vastauksesta käyttäjä saa 10 pistettä. Väärästä vastauksesta käyttäjä menettää 5 pistettä.

Haastetehtävän voittamisesta saa 10 pistettä. Lisäksi haastepelin voitosta saa lisäpisteitä, jos toisen käyttäjän kokonaispistemäärä ennen pelin aloitusta oli suurempi. Saatujen lisäpisteiden määrä riippuu käyttäjillä ennen pelin aloitusta olleista kokonaispisteistä. Haastepelin voittaja saa 10 prosenttia käyttäjien kokonaispisteiden erotuksesta.

Mahdolliset haastetehtävän lopputulosten eri tapaukset ja käyttäjien saamat pisteet on listattu taulukossa 6.

Taulukko 6: Haastetehtävän lopputulosten tapaukset ja niiden pisteitys.

Tapaus	Ensimmäiseksi vastanneen saamat pisteet	Toiseksi vastanneen saamat pisteet
Ensimmäiseksi vastanneen vastaus on väärin. Toiseksi vastanneen vastaus on väärin.	Vastauksesta -5 Haasteesta 0	Vastauksesta -5 Haasteesta 0
Ensimmäiseksi vastanneen vastaus on väärin. Toiseksi vastanneen vastaus on oikein.	Vastauksesta -5 Haasteesta 0	Vastauksesta +10 Haasteesta +10 +10 % pistemäärien positiivisesta erotuksesta
Ensimmäiseksi vastanneen vastaus on oikein. Toiseksi vastanneen vastaus on väärin.	Vastauksesta +10 Haasteesta +10 +10 % pistemäärien positiivisesta erotuksesta	Vastauksesta -5 Haasteesta 0
Ensimmäiseksi vastanneen vastaus on oikein. Toiseksi vastanneen vastaus on oikein.	Vastauksesta +10 Haasteesta +10 +10 % pistemäärien positiivisesta erotuksesta	Vastauksesta +10 Haasteesta 0
Ensimmäiseksi vastanneen vastaus on väärin. Toinen pelaaja ei vastaa lainkaan.	Vastauksesta -5 Haasteesta 0	Vastauksesta 0 Haasteesta 0
Ensimmäiseksi vastanneen vastaus on oikein. Toinen pelaaja ei vastaa lainkaan.	Vastauksesta +10 Haasteesta +10 +10 % pistemäärien positiivisesta erotuksesta	Vastauksesta 0 Haasteesta 0
Kumpikaan pelaaja ei vastaa lainkaan.	Vastauksesta 0 Haasteesta 0	Vastauksesta 0 Haasteesta 0

Käyttäjien saamat pisteet lasketaan yhteen ja lisätään käyttäjien kokonaispistemääriin. Käyttäjät voivat tarkastella palauteruutua niin kauan kuin haluavat. Palauteruudulta käyttäjä voi palata takaisin käyttäjälistaan uuden haasteen lähettämistä varten.

Yksinpeli

Yksinpelissä käyttäjä voi tehdä aikarajoitettuja harjoituksia tietokonevastustajaa vastaan. Käyttäjän tulee ensin valita vaikeustaso helpon, keskivaikean ja vaikean välillä (kuva 16).

Valitse vaikeustaso	
Helppo	10p
Keskivaikea	20p
Vaikea	30p

Kuva 16: Yksinpelin vaikeustason valinta.

Tietokonevastustajat toteutettiin luomalla järjestelmään kolme näennäistä käyttäjää, jotka on asetettu tietokonevastustajiksi. Kun käyttäjä valitsee vaikeustason, valitulle tietokonevastustajalle lähetetään haaste, jonka tämä hyväksyy automaattisesti. Tietokonevastustajat voivat vastaanottaa ja hyväksyä haasteita kesken haastepelin, jotta useammat käyttäjät voivat haastaa saman tietokonevastustajan yhtäaikaisesti.

Kun käyttäjä on valinnut vaikeustason, hänelle esitetään samantyyppinen monivalinta-tehtävä kuin harjoittelussa ja moninpelissa. Tehtävä on aikarajoitettu samoin kuin moninpeli. Käyttäjä voittaa pelin, jos hän onnistuu antamaan oikean vastauksen ensimmäisenä (kuva 17). Jos käyttäjä antaa väärän vastauksen tai tietokonevastustaja ehtii vastata oikein ensin, käyttäjä häviää pelin.

Yksinpeli	Takaisin
<hr/>	
Aikaa kulunut: 19 s	
Vastasit ensimmäisenä (16 s)!	
Mikä on x:n arvo, kun	
$x^2+4 = 29$	
<input type="radio"/> 3	
<input type="radio"/> 4	
% 5 ← Molemmat vastasivat oikein!	
<input type="radio"/> 6	
Malliratkaisu:	
$x^2+4 = 29$	
$x^2 = 29-4$	
$x = \text{sqrt}(29-4)$	
$x = 5$	

Kuva 17: Käyttäjä vastaa ensimmäisenä.

Tietokoneen vastausaika riippuu valitusta vaikeustasosta ja tehtävän tekijän antamista asetuksista. Helppimmalla vaikeustasolla tietokonevastustaja vastaa, kun tehtävän alusta on kulunut 50 sekuntia. Helppoin tietokonevastustaja vastaa oikein 50 prosentin todennäköisyydellä. Keskivaikealla vaikeustasolla tietokonevastustaja vastaa 40 sekunnin kohdalla ja vastaus on 75 prosentin todennäköisyydellä oikein. Vaikeimmalla vaikeustasolla tietokonevastustaja antaa vastauksen 30 sekunnin kohdalla ja vastaus on 90 prosentin todennäköisyydellä oikein. Tietokonevastustajan vastaus toteutettiin palvelimella ajastettuna tehdystä HTTP-kutsusta. Todellisuudessa tietokonevastustaja tietää oikean vastauksen aina, mutta automaattisen vastauksen lähetyksen ohjelma-koodi (esimerkkikoodi 1) heikentää todennäköisyyttä vastata oikein.

```

require 'curb'
class Bot
  ...
  def submit_answer_for question
    answer_uri = question.correct_answer_uri
    if random(100) < self.error_probability
      answer_uri = question.incorrect_answer_uri
    end
    question.answer_response = Curl.get(answer_uri)
    question
  end
  ...

```

Esimerkkikoodi 1: Tietokonevastustajan automaattivastauksen satunnaisvirhe.

Peli päättyy, kun sekä käyttäjä että tietokone ovat vastanneet, tai viimeistään yhden minuutin kuluttua tehtävän aloituksesta (kuva 18).

Yksinpeli (vaikea)	Takaisin
<p>Aikaa jäljellä: 19 s</p> <p>Tietokone vastasi ensimmäisenä (30 s)...</p> <p>Mikä on x:n arvo, kun</p> $x^2+4 = 29$ <p> <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 </p> <p>Malliratkaisu:</p> $x^2+4 = 29$ $x^2 = 29-4$ $x = \sqrt{29-4}$ $x = 5$	<p>Peli on päättynyt.</p> <p>Tietokone voitti. Jäit ilman pisteitä.</p> <p>Mikä on x:n arvo, kun</p> $x^2+4 = 29$ <p> <input type="radio"/> 3 <input checked="" type="radio"/> 4 <- Vastauksesi oli väärin! <input checked="" type="radio"/> 5 <- Tietokone vastasi oikein! <input type="radio"/> 6 </p> <p>Malliratkaisu:</p> $x^2+4 = 29$ $x^2 = 29-4$ $x = \sqrt{29-4}$ $x = 5$

Kuva 18: Yksinpelin eteneminen ja pelin päättyminen.

Mikäli käyttäjä onnistuu voittamaan tietokonevastustajan, hän saa yksinpelipisteitä. Yksinpelipisteet ovat erillisiä monipelissä käytetyistä haastepisteistä. Yksinpelipisteet toimivat kuten taito- ja kokemuspisteet, eli niitä ei vähennetä missään vaiheessa.

On toivottavaa, että käyttäjä harjoittelee ajastettuja tehtäviä ainakin muutaman kerran tietokonetta vastaan, ennen kuin hän lähtee haastamaan muita käyttäjiä monipelissä.

4.4 Testaus

Toteutetun verkkosovelluksen testauksessa otettiin huomioon järjestelmävirheiden ilmitytulo ja korjattavan osan nopea paikallistaminen. Testauksessa käytettiin apuna Ruby on Rails -ohjelmistokehyksen ja Better Errors -lisäosan tarjoamia mahdollisuuksia käydä ohjelmistovirheitä syvemmin läpi. Lisäosa mahdollisti virheen paikallistamisen nopeammin kuin pelkän järjestelmän virhelokin perusteella on mahdollista. Pääasiallinen tehokkuushyöty saatiin mahdollisuudella käsitellä virheen aiheuttanutta oliota suorittamalla Ruby-komentoja suoraan virheilmoituksen sisältävässä selainikkunassa. Järjestelmään asennettiin myös Airbrake-lisäosa käytönaikasta virheseurantaa varten. Airbrake-lisäosaa voidaan käyttää raportoimaan virheistä selaimella käytettävään ulkopuoliseen verkkosovellukseen.

Järjestelmä myös testattiin tietoturvaavaoittuvuuksien varalta. Koska järjestelmää on tarkoitus käyttää vain sisäiseen tutkimukseen ja tuotekehitykseen, testaus suoritettiin kevyemmällä prosessilla kuin kaupallisessa järjestelmässä on tapana. Vaarana tässä on kuitenkin se, että esimerkiksi tietoturvaavaoittuvuudet jäävät huomaamatta ja saatavat hautautua syvälle sovelluksen arkkitehtuuriin, mikäli niitä ei huomata ja korjata ennen jatkokehityksen toteuttamista.

Tietoturvatestauksessa käytettiin apuna OWASP (Open Web Application Security Project) Top Ten 2013 -listaa, joka on OWASP Foundation -järjestön ylläpitämä, yleisimmät verkkosovellusten tietoturvaavaoittuvuudet listaava sivusto (Smithline & Vishwa 2013). Listan luettelemat tyypilliset tietoturvaavaoittuvuudet käytiin läpi ja havaitut tietomurtomahdollisuudet korjattiin. Löydetyt haavoittuvuudet ja niiden selitteet on lueteltu taulukossa 7.

Taulukko 7: Toteutetussa verkkosovelluksessa havaitut tietoturvaavaoittuvuudet ja niiden korjaukset.

Haavoittu- vuustyyppi	Komponentti ja kuvaus	Korjaus
A4 Insecure Direct Object References	Moninpelitoiminto sallii pelikutsun hyväksymisen käyttäjiltä, joille kutsua ei ole lähetetty.	Moninpelitoiminto tarkistaa pelikutsun vastaanottajan eikä salli muiden hyväksyä kutsua.
A4 Insecure Direct Object References	Harjoittelutoiminto sallii toisen käyttäjän harjoitushistorian selaamisen.	Harjoittelutoiminto listaa vain ne harjoitukset, joiden omistaja käyttäjä on.
A8 Failure to Restrict URL Access	Harjoittelutoiminto sallii käyttäjän vastata tehtävään uudestaan, vaikka vastaus on jo annettu.	Harjoittelutoiminto sallii tehtävän vastauksen tallentamisen vain, mikäli sitä ei vielä ole tietokannassa.

A5-, A6-, A7-, A9- ja A10-tyyppisten haavoittuvuuksien testauksen huomattiin vievän paljon testausaikaa, eikä niiden etsimistä kehitysympäristöstä nähty hyödylliseksi.

Verkkosovellus käyttää Devise-lisäosaa käyttäjien todennukseen ja istunnonhallintaan, joten A3-haavoittuvuuksien mahdollisuus riippuu lisäosan tietoturvan laadusta. Lisäosa on kuitenkin laajalti käytössä ja Ruby on Rails -yhteisö on aktiivinen etsimään ja korjaamaan tietoturvaavaoittuvuuksia, joten haavoittuvuuden riski on suhteellisen pieni. A3-tyyppisiä haavoittuvuuksia ei tämän vuoksi aktiivisesti etsitty.

Analyysi

Verkkosovelluksen harjoitus- ja yksinpeliosuuksien havaittiin toimivan luontevasti. Mobiiliyhteydelle ominaiset pitkät latausajat eivät häirinneet käyttöä, vaikka ajastetut tehtävät vaativatkin kylmähermoisuutta. Moninpeliosuudessa ilmeni hetkiä, jolloin toiminto oli valittu, mutta selain ei ladannut sivua niin nopeasti, kuin toiminto antoi ymmärtää. Selaimen latauspalkin eteneminen ei korreloinut todellista latausaikaa.

Verkkosovellusta käytettäessä havaittiin myös tapauksia, joissa mobiililaitteen tai selaimen JavaScript- tai näytönpäättösuorituskyky aiheutti viiveitä. Esimerkiksi pelin aloittaminen ja ensimmäisen kysymyksen näyttäminen toimivat toisella käyttäjällä välittömästi, mutta kestivät toisella käyttäjällä useamman sekunnin, vaikka resurssit on esiladattu palvelimelta. Tällöin käyttäjät oli asetettu keskenään eriarvoiseen asemaan, kos-

ka toinen käyttäjä sai enemmän vastausaikaa. Tällaisten tapausten välttämiseksi ainoa tapa on kattava testaus erilaisilla mobiililustoilla, erojen kartoitus ja optimoidun version tekeminen suorituskyvyltään heikoille laitteille. Toisaalta mobiililaittekohtaisissa toteutuksissa menetetään nopean kehitystyön edut ja testaus vie paljon aikaa.

Insinöörityönä tehdyssä järjestelmässä palvelun ulkoasuun ei ollut mahdollista käyttää yhtä paljon resursseja kuin vastaavan kaupallisen palvelun toteuttamisessa. Käyttökokemusta voi parantaa nykyisestä kehittämällä tehtävien ja käyttöliittymän graafista esitystapaa.

Uusia tehtävätyyppejä, kuten graafisia yhdistelytehtäviä, on mahdollista lisätä, mikäli se nähdään tarpeelliseksi. Tehtävätyyppien lisäämisessä tulee kuitenkin huolehtia siitä, että tehtävänratkaisuun käytetty aika ei ole liian pitkä, sillä haasteiden lähettäjä voi joutua odottamaan koko tämän ajan, ennen kuin hän voi haastaa toisen käyttäjän, jolla on jo haaste käynnissä. Verkkoselainten monimuotoisuuden ja mobiiliselainten rajoitusten vuoksi vedä- ja pudotatoimintoja kannattaa välttää uusissa tehtävätyypeissä.

Kiertokyselyihin perustuvat tilapäivitykset toimivat hyvin silloin, kun verkko tai palvelu olivat ruuhkattomia, mutta ruuhkatilanteessa tilapäivityksiin ei enää vastattu riittävän nopeasti ja moninpelin käytettävyys heikkeni. Käyttäjät joutuivat odottamaan haaste-tehtävän käynnistymistä kauemmin, ja jos peli oli käynnissä, he eivät saaneet tietoa vastustajan antamasta vastauksesta. Kun kapasiteetti loppui, tieto online-käyttäjien tilasta ei ollut enää tarkkaa ja käyttäjät pystyivät lähettämään haastepyyntöjä käyttäjille, jotka olivat jo kirjautuneet ulos palvelusta tai aloittaneet moninpelin toisen käyttäjän kanssa.

Järjestelmän todellinen kapasiteetti riippuu käytetystä palvelin- ja tietokanta-arkkitehtuurista, palvelimen resursseista ja verkon kapasiteetista. Järjestelmän suorituskyky vaihtelee näistä riippuen niin paljon, että kapasiteettiarviota ei kannata laskea yksittäisten komponenttien mukaan. Palvelimien ja verkon kokoonpanon suorituskyky kannattaa aina testata todellisessa ympäristössä. Mikäli pyyntöihin vastaamiseen kuluva aika on yli 5 sekuntia, tilapäivitysten arkkitehtuuria tulee optimoida tai käyttäjämääräennustetta kannattaa laskea. Tärkeintä on tietää, missä käyttäjämäärän raja sijaitsee ja mikä siihen vaikuttaa, jotta nämä asiat voidaan ottaa huomioon käyttöönottoa suunniteltaessa.

Käyttäjiltä vaadittiin sähköpostiosoitteen, salasanojen ja salasananvarmistuksen syöttöä rekisteröidyttäessä. Tämä aiheutti mobiililaitteilla testattaessa toisinaan käytettävyyssongelmia tapauksissa, joissa valittu salasana ei kelvannut ja salasاناتiedot tuli syöttää uudelleen. Nämä ovat kuitenkin tarpeellisia ominaisuuksia käyttäjätilien turvaamisen vuoksi, joten käytettävyyssongelmista ei voi päästä kokonaan eroon. Käyttöliittymä tehtiin kuitenkin mahdollisimman helppokäyttöiseksi niin, että prosessi on tuttu aikaisemmin muihin mobiilipalveluihin rekisteröityneille käyttäjille.

4.5 Jatkokehitysmahdollisuudet

Järjestelmä toteutettiin niin, että jatkokehitys on suoraviivaista. Ruby on Rails -ohjelmistokehityksen ja siihen liittyvän tuotantotavan antaa sen hyödyn, että uuden ohjelmistokehittäjän ei tarvitse perehtyä tekniseen dokumentaatioon jatkokehitystä varten.

Koska ulkoasu on tehty Ruby on Rails -ohjelmistokehityksen näkymäkomponenteilla ja CSS-tyylitiedoilla eikä erillistä tukea graafisten teemojen käytölle ole, ulkoasumuokkaukset kannattaa tehdä git-versionhallinnalla luotuun uuteen haaraan ("branch").

Jatkokehityksessä kannattaa keskittyä tilapäivitysarkkitehtuurin optimoimiseen, mikäli tavoitteena on tuoda palvelu suurempien käyttäjämäärien ulottuville.

Ruby on Rails -ohjelmistokehityksellä toteutettu REST-rajapinta antaa hyvät mahdollisuudet käyttää palvelua tai jotakin palvelun toiminnoista osana jotakin muuta ulkopuolista ratkaisua. Joissain tapauksissa rajapintaa voidaan käyttää suoraan, mutta mikäli ulkopuolinen ratkaisu niin vaatii, rajapintaan voidaan lisätä määrittelyitä vaikuttamatta www-käyttöliittymän käyttäjiin. Jatkokehityksessä voidaan esimerkiksi toteuttaa tietyille mobiililaitetyypille optimoitu natiivisovellus, joka kommunikoi järjestelmän kanssa REST-rajapinnan kautta.

Ammattimaisten Ruby on Rails- ja JavaScript-ohjelmistokehittäjien kannattaa tehdä järjestelmälle tietoturvakartoitus ennen jatkokehittämisen aloittamista. Näin voidaan varmistua siitä, että mahdolliset päälle rakennetut uudet ominaisuudet eivät altista tietoturvariskeille alkuperäisten ominaisuuksien kautta.

Sisältö, jota verkkosovellus voi näyttää, ei ole rajoitettu matemaattiseen sisältöön. Aihealueena voi olla mitä tahansa harjoitusperusteisesti opetettavaa sisältöä. Kannattaa

kuitenkin huomioida, että järjestelmä kannustaa ratkaisemaan haastetehtävät tietyssä ajassa. Parhaiten järjestelmän konseptiin mukautuu sellainen sisältö, jossa toistuvalla harjoittelulla voi parantaa ratkaisun löytämisen rutiinia ja virheiden välttämistä. Yksi esimerkki tällaisesta sisällöstä on monimutkaisen teknisen ympäristön tai järjestelmän, kuten verkkoarkkitehtuurin tai sähköteknisen laitteiston, asetusten määrittäminen tai suunnittelu.

Palveluun on suoraviivaista lisätä pikaviesti- tai keskustelutoiminto, esimerkiksi haaste-tehtävän palauteruudulle. Pikaviestitoiminnolla käyttäjät voivat keskustella pelin kulusta ja oikeasta ratkaisumallista ja näin oppia tehtävänratkaisusta toisiltaan. Järjestelmän arkkitehtuurissa on tuki tilapäiviyksille, ja uusien viestien tarkistus on mahdollista tehdä määrittelemällä uusi tilapäiviyksen ilmoitustyyppi. Haasteellisinta viestittelytoiminnossa on mobiiliselainten käyttöliittymä. Näytöllä on tilaa rajoitetusti, joten käyttöliittymän suunnitteluun kannattaa panostaa, jotta palvelu ei muutu monimutkaiseksi viestitoimintojen ja -ilmoitusten lisäämisen myötä.

Pelimäistä vaikutelmaa voidaan lisätä käyttämällä opiskelijalle annettavia palkintoja hänen suorittamiensa tehtävien mukaan. Oikein toteutettuna niiden käyttämisellä voi lisätä käyttäjien motivaatiota (Bayliss & Schwartz 2009: 12). Palkinnot voi toteuttaa niin, että käyttäjän voi esimerkiksi palkita virtuaalisella kulta-, hopea- tai pronssimitalilla riippuen siitä, kuinka monta peräkkäistä tehtävää hänen onnistuu ratkaista oikein. Mikäli sisältöä on jaoteltu eri aihealueisiin, palkinnot voi antaa riippuen valitusta aihealueesta. Myös saavutuspalkintojen lisääminen on mahdollista. Esimerkiksi trigonometrian osa-alueella ratkaistu tehtäväkokonaisuus voi merkitä ”trigonometriamestari”-saavutuksen suoritetuksi.

5 Yhteenveto

Insinööriytyössä perehdyttiin mobiilioppimiseen ja sen pelillistämiseen. Insinööriytyössä myös tutkittiin erilaisia keinoja mobiilioppimispalvelun käyttäjien motivoimiseksi ja sitouttamiseksi.

Mobiilioppimisen pelillistämistä esittelevän verkkosovelluksen suunnittelussa käytettiin pelillistämiskeinoina reaaliaikaista moninpeliä, käyttäjien välistä haastepisteitystä ja eritasoisia tietokonevastustajia. Haastepisteitys oli yksinkertainen toteuttaa, ja sen havaittiin olevan myös käyttäjille tarpeeksi helppotajuinen.

Mittavimmat tekniset haasteet tulivat käyttäjien välisestä tiedonvaihdosta, jossa kahden eri käyttäjän tilaa tuli pitää synkronoituna sekä toistensa että palvelimen kanssa. Näitä haasteita ilmeni kahdessa sovelluksen osassa: haastepelin aloituksen synkronoinnissa ja vastauksien antamisten ajastuksissa. Kun verkkoyhteydet toimivat nopeasti, suuria ongelmia ei esiintynyt. Verkkoa vaihdettaessa ja kuuluvuusalueen rajoilla oltaessa moninpeliosion käytöstä tuli katkonaista. Joitain ylimääräisiä viiveitä pystyttiin kiertämään resurssien esilatauksella.

Sovellusta ei optimoitu millekään tietylle päätelaitemalleille tai mobiiliselaimen versiolle. Testauksessa sovelluksen havaittiin toimivan hyvin myös alkuperäisestä kehityssuunnitelmasta pois jätetyissä selainversioissa. Yleisimmistä mobiiliselaimista vain välipalvelimen avulla tietoa tiivistävät Opera Mini- ja Nokia Xpress browser -selaimet aiheuttivat yhteensopivuusongelmia verkkosovelluksen kanssa.

Sovelluksen kehitystyö tehtiin Ruby on Rails -ohjelmistokehyksellä. Sen yksi havaittu hyöty oli, että kehykseen löytyi paljon valmiita Gem-lisäosia, joiden avulla ohjelmistokirjastojen käyttö ja kolmannen osapuolen palveluihin yhdistäminen oli suoraviivaista. Säästynyt aika käytettiin sovelluksen ydintoimintojen parantamiseen. Yhtenäisen arkkitehtuurin ja yleisesti käytössä olevien työtapojen seuraamisesta on hyötyä myös jatkokehityksessä.

Konsepti suunniteltiin sisällön puolesta yksinkertaiseksi. Olemassa olevaa valmista sisältöä voi käyttää esimerkkisovelluksen kanssa, kunhan sisältö ottaa mobiililaitteen näyttökoon rajoitukset huomioon. Konseptia ja sovellusta käytetään Bitville Oy:n mLearning-toteutusten kehitystyössä.

Lähteet

Adams, Ernest. 2010. Fundamentals of Game Design. Berkeley, CA, USA: New Riders, Pearson Education.

Ally, Mohamed. 2009. Mobile Learning: Transforming the Delivery of Education and Training. Edmonton, AB, CA: Athabasca.

About Badgeville. 2013. Verkkodokumentti. Badgeville. <<http://badgeville.com/about>>. Luettu 30.4.2013.

Bayliss, Jessica & Schwartz, David. 2009. Instructional design as game design. Proceedings of the 4th International Conference on Foundations of Digital Games FDG 09. New York, NY, USA: ACM Press.

Burnham, Trevor. 2012. Async JavaScript - Build More Responsive Apps with Less Code. USA: The Pragmatic Programmers.

Claypool, Mark & Claypool, Kajal. 2010. Latency Can Kill: Precision and Deadline in Online Games. Proceedings of the First ACM Multimedia Systems Conference. Scottsdale, AZ, USA.

Empson, Rip. 2012. Khan Academy Brings Its 3,500+ Educational Videos To The iPhone. Verkkodokumentti. TechCrunch / AOL Inc. <<http://techcrunch.com/2012/11/28/khan-academy-brings-its-3500-educational-videos-to-the-iphone/>>. Päivitetty 28.11.2012. Luettu 28.4.2013.

Griffiths, Daniel. 2013. Dragon Age 3, Next Mass Effect To Use Frostbite 3. Verkkodokumentti. Forbes. <<http://www.forbes.com/sites/danielnyegriffiths/2013/03/28/dragon-age-3-next-mass-effect-to-use-frostbite-3/>>. Päivitetty 28.3.2013. Luettu 14.4.2013.

Habgood, Jacob & Overmars, Mark. 2006. The Game Maker's Apprentice. New York, NY, USA: Apress.

Herger, Mario. 2011. A Checklist for Evaluating Gamification Platforms. Verkkodokumentti. Enterprise-Gamification.com. <<http://enterprise-gamification.com/index.php/blog/4-blog/42-a-checklist-for-evaluating-gamification-platforms>>. Päivitetty 24.10.2011. Luettu 2.5.2013.

Huotari, Kai & Hamari, Juho. 2012. Defining gamification: a service marketing perspective. MindTrek '12 Proceeding of the 16th International Academic MindTrek Conference, s.17-22. New York, NY, USA: ACM.

Ihanainen, Pekka & Kalli, Pekka. 2009. Verkon varassa - opetuksen pedagoginen kehittäminen verkkoympäristössä. Tampere: Jyväskylän ammattikorkeakoulu.

Jain, Ravi. 2006. The Mobile Web in Developing Countries. Verkkodokumentti. W3C. <<http://www.w3.org/2006/07/MWI-EC/PC/rj-mobileweb-developingcountries.pdf>>. 2006. Luettu 1.5.2013.

Jenkins, Tom. 2012. No more pilots, tests or trials: it's time for mLearning to grow up. Verkkodokumentti. CrossKnowledge Academy. <<http://www.crossknowledge-academy.com/blog/no-more-pilots-tests-or-trials-its-time-for-mlearning-to-grow-up/#.UX-u8LX-F8E>>. Päivitetty 13.12.2012. Luettu 30.4.2013.

Järvelä, Sanna, Häkkinen, Päivi & Lehtinen, Erno (toim). 2006. Oppimisen teoria ja teknologian opetuskäyttö. Helsinki: WSOY.

LaPointe, Deborah. 2008. Will Games and Emerging Technologies Influence the Learning Landscape? Learners in a Changing Learning Landscape, s. 227–249. Springer Netherlands.

Leggetter, Phil. 2012. How To Build A Real-Time Commenting System. Verkkodokumentti. Smashing Magazine. <<http://coding.smashingmagazine.com/2012/05/09/building-real-time-commenting-system/>>. Päivitetty 9.5.2012. Luettu 28.3.2013.

Lindquist, David, Denning, Tamara, Kelly, Michael & Malani, Roshni. 2007. Exploring the Potential of Mobile Phones. La Jolla, CA, USA: University of California.

Makzan. 2011. HTML5 Games Development by Example. Birmingham, UK: Packt Publishing.

McHaney, Roger. 2011. New Digital Shoreline : How Web 2.0 and Millennials Are Revolutionizing Higher Education. Sterling, VA, USA: Stylus Publishing.

Ram, Ashwin, Ai, Hua, Ram, Preetha & Sahay, Saurav. 2011. Open social learning communities. WIMS '11 Proceedings of the International Conference on Web Intelligence, Mining and Semantics. New York, NY, USA.

Raymer, Rick. 2011. Gamification: Using Game Mechanics to Enhance eLearning. Verkkodokumentti. eLearn Magazine, ACM. <<http://elearnmag.acm.org/featured.cfm?aid=2031772>>. 2011. Luettu 2.5.2013.

Rettig, Pascal. 2012. Professional HTML5 Mobile Game Development. Somerset, NJ, USA: Wiley.

Shankar, Aditya Ravi. 2012. Pro HTML5 Games. New York, NY, USA: Apress.

Siering, Greg. 2012. Gamification: Using Game-like Elements to Motivate and Engage Students. Verkkodokumentti. Indiana University. <<http://citl.indiana.edu/news/newsStories/dir-mar2012.php>>. 2012. Luettu 2.5.2013.

Smithline, Neil & Rajiv, Vishwa. 2013. OWASP Top 10 2013. Verkkodokumentti. OWASP Foundation. <https://www.owasp.org/index.php/Top_10_2013-T10>. Päivitetty 18.4.2013. Luettu 28.4.2013.

Starkey, Daniel. 2013. Trends of this Generation: Gamification. Verkkodokumentti. Destructoid. <<http://www.destructoid.com/trends-of-this-generation-gamification-243999.phtml>>. Päivitetty 19.2.2013. Luettu 2.5.2013.

Thompson, Clive. 2011. How Khan Academy Is Changing the Rules of Education. Verkkodokumentti. Wired Magazine. <http://www.wired.com/magazine/2011/07/ff_khan/all/1>. Päivitetty 15.7.2011. Luettu 28.4.2013.

Wexler, Steve. 2008. Guild Reseach 360 report on Mobile Learning. Santa Rosa, CA, USA: The eLearning Guild.

Vorderer, Peter, Hartmann, Tilo & Klimmt, Christoph. 2003. Explaining the enjoyment of playing video games: the role of competition. ICEC '03 Proceedings of the second international conference on Entertainment computing, s. 1–9. Pittsburgh, PA, USA.

Yuan, Li & Powell, Stephen. 2013. MOOCs and Open Education: Implications for Higher Education. Bolton, UK: JISC CETIS.

Zichermann, Gabe & Cunningham, Christopher. 2011. Gamification by Design. Sebastopol, CA, USA: O'Reilly Media.