



Paavo Kokkarinen

OPPIMISYMPÄRISTÖN TEKNINEN TOTEUTUS SELAIMELLE

OPPIMISYMPÄRISTÖN TEKNINEN TOTEUTUS SELAIMELLE

Paavo Kokkarinen
Opinnäytetyö
Kevät 2013
Tietotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, ohjelmistokehitys

Tekijä: Paavo Kokkarinen
Oppinnäytetyön nimi: Oppimisympäristön tekninen toteutus selaimelle
Työn ohjaaja: Terhi Holappa
Työn valmistumislukukausi ja -vuosi: Kevät 2013 Sivumäärä: 45

Oppinnäytetyön aiheeksi valittiin www-selaimelle tarkoitetun oppimisympäristön toteutus. Työn taustalta löytyy tilaajan tuntemat opettajat, jotka eivät ole olleet tyytyväisiä nykyisin käytössä oleviin ympäristöihin. Ne ovat opettajien mielestä liian hajautuneita ja eri sovellusten välillä liikkumiseen tuhlaantuu työaika. Lisäksi opettajat ovat kokeneet, etteivät he pysty tekemään muistiinpanoja oppilaiden kehityksestä riittävän helposti. Tilaaja on myös suorittanut markkinatutkimuksen, jonka mukaan uudelle ympäristölle olisi kysyntää, jos se korjaisi nykyisten ympäristöjen ongelmia.

Työn tavoitteena oli rakentaa www-selaimella toimiva käyttöliittymä, jonka avulla oppilaat, opettajat sekä oppilaiden huoltajat voivat kommunikoida keskenään. Lisäksi ympäristöön tehtiin toiminnot, joiden avulla oppilaiden kehitystä ja toimintaa koulussa voidaan seurata.

Oppimisympäristön kehitysprosessi koostui neljästä eri vaiheesta. Ensimmäisenä tehtiin tilaajan kanssa määrittely, jossa selvitettiin, mitä ominaisuuksia valmiilta ympäristöltä odotetaan. Sitä seurasi suunnittelu, jossa valittiin toteutusta varten tekniikat, sekä hahmoteltiin ohjelmiston rakenne. Toteutusvaiheessa kirjoitettiin itse ohjelmakoodi, joka testattiin yhdessä tilaajan kanssa testausvaiheessa.

Työ toteutettiin käyttäen vapaasti saatavilla olevia tekniikoita, jotta ympäristön käyttöönotosta ei koidu tilaajalle ylimääräisiä kustannuksia. Ympäristö on toteutettu PHP-ohjelmointikielellä ja käyttöliittymä muodostetaan HTML- ja CSS-tekniikoilla. Tietojen varastointiin käytetään MySQL-tietokantapalvelinta. Koko ympäristö sijaitsee tilaajan vaatimuksen mukaan Azure-pilvipalvelussa.

Työn tuloksena saatiin tuotettua tilaajalle käyttövalmis ohjelmisto, jota tilaaja voi esitellä ja hyödyntää tarpeensa mukaan. Kaikki tilaajan haluamat ominaisuudet saatiin toimimaan halutulla tavalla, eikä testausvaiheen aikana tullut negatiivista palautetta. Työn toteutus sujui siis hyvin, ja ohjelmisto on rakennettu niin, että sitä voi laajentaa tarvittaessa mahdollisimman helposti.

Asiasanat: PHP, MySQL, HTML, tietokanta, oppimisympäristö, tiedonkulku, koulut

ABSTRACT

Oulu University of Applied Sciences
Information technology, programming

Author: Paavo Kokkarinen

Title of thesis: Learning environment implementation for a web browser

Supervisor: Terhi Holappa

Term and year when the thesis was submitted: Spring 2013 Pages: 45

The subject of thesis was chosen to be implementation of a learning environment for a web browser. Contractor of thesis has had contacts with teachers who haven't been satisfied with the environments that they are currently using. Teachers think that they are too dispersed and time is wasted on moving between different applications. Teachers also feel that they cannot make notes about the development of students easily enough. The contractor has also performed a market research which says that there is a demand for a new kind of environment if it would fix the existing problems. The target of thesis was to build a user interface that works in a web browser, which students, teachers and guardians can use to communicate with each other. Additional features were built which can be used to monitor the development and activity of students.

The development of learning environment consisted of four phases. In first phase a software specification with the contractor was made, which tells what features were expected from the final result. After it was the design phase, in which the technologies that the environment will use were chosen. In implementation phase the software code was written, which was tested together with the contractor in testing phase.

Thesis was implemented using freely available technologies so that the installation of environment won't generate additional costs for contractor. Environment has been developed using PHP-programming language and user interface is generated using HTML and CSS technologies. Information is stored in a MySQL database server. Entire environment is located in Azure cloud service as required by the contractor.

As a result of thesis a ready for use software was developed, which can be used for presentation or actual use by the contractor according to its needs. Every feature that the contractor requested was made to work as intended and no negative feedback was received during the testing phase. The implementation of thesis succeeded well and the software is built so that it can be extended as easily as possible.

Keywords: PHP, MySQL, HTML, database, learning environment

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
KÄYTETYT LYHENTEET	7
1 JOHDANTO	8
2 MARKKINOILLA OLEVAT OPPIMISYMPÄRISTÖT	9
2.1 Wilma	9
2.2 Helmi	10
2.3 Kivahko	10
3 YLEINEN OHJELMISTOPROSESSI	12
4 SELAINPOHJAISTEN SOVELLUSTEN KEHITYSTYÖKALUT	14
4.1 PHP	14
4.2 MySQL	14
4.3 Windows Azure	15
4.4 HTML	15
4.5 CSS	15
5 PÄÄTARKOITUS JA TAVOITTEET	17
6 OPPIMISYMPÄRISTÖN KEHITYSPROSESSI	18
6.1 Määrittely	18
6.1.1 Tietokannan teko	18
6.1.2 Käyttäjien hallinta	18
6.1.3 Kurssien hallinta	19
6.1.4 Viestintäjärjestelmä	19
6.2 Suunnittelu	19
6.3 Ohjelmointi	20
6.4 Testaus	20
7 OPPIMISYMPÄRISTÖN TOTEUTUS	22
7.1 Käytetyt tekniikat	22
7.2 Tietokanta	23
7.2.1 Käyttäjätilit	23

7.2.2 Kurssit	24
7.2.3 Viestintä	25
7.3 Näkymät	25
7.4 Ohjelmointi	26
7.4.1 Istunto	26
7.4.2 Turvallisuus	27
7.4.3 UNIX-aika	27
8 TOTEUTETUN OHJELMISTON KUVAUS	29
8.1 Ohjelmiston rakenne	29
8.2 Kirjautuminen	30
8.3 Käyttäjien hallinta	31
8.4 Kurssien hallinta	34
8.5 Viestintäjärjestelmä	37
9 POHDINTA	40
LÄHTEET	42

KÄYTETYT LYHENTEET

Azure	Microsoftin ylläpitämä pilvipalvelu, jonka kautta sivuston saa näkymään Internetissä.
CSS	Cascading Style Sheets, WWW-dokumenteille kehitetty tyyliohjeiden laji.
Funktio	Aliohjelma, joka suorittaa tietyn toiminnon.
HTML	Hypertext Markup Language, kieli jolla tehdään WWW-sivuja.
MySQL	Relaatiotietokantaohjelmisto.
PHP	Hypertext Preprocessor, palvelinympäristöissä käytetty ohjelmointikieli.
SQL	Structured Query Language, relaatiotietokantojen kyselykieli.

1 JOHDANTO

Internet-selaimella käytettävät oppimisympäristöt ovat nykyään tärkeä osa koulumaailmaa (1). Ne toimivat eräänlaisena nykyajan reissuvihkona, jonka kautta koulu ja koti voivat olla yhteydessä keskenään. Ympäristöjen kautta voidaan reaaliaikaisesti tiedottaa koulun ajankohtaisista tapahtumista sekä niiden kautta koulun henkilökunta, oppilaat sekä huoltajat voivat kommunikoida keskenään. Lisäksi ympäristöistä löytyy yleensä työkalut kurssikirjanpitoon eli tiedot siitä, mille kursseille oppilaat ovat osallistuneet ja kuinka he ovat suoriutuneet niistä.

Opinnäytetyön taustalla on tilaajan kontaktit opettajiin, jotka ovat toivoneet ympäristöä, joka vastaisi paremmin heidän toiveitaan. Toiveet liittyvät ympäristön käytön helppouteen sekä vuorovaikutukseen oppilaiden ja huoltajien kanssa. Opettajilla on nykyisissä ympäristöissä ongelmia tehdä muistiinpanoja oppilailta. He ovat toivoneet, että voisivat helposti panna muistiin tietoja esimerkiksi oppilaan oppimiseen liittyvistä ongelmista tai tehtävien suorituksista. Myöskään vanhemmat eivät ole riittävän hyvin selvillä koulun tapahtumista tai lastensa kehitymisestä. Olemassa olevat ympäristöt ovat myös liian hajautuneita, eli asioiden hoitaminen ei onnistu yhdestä paikkaa, vaan aikaa tuhlaantuu eri ympäristöjen välillä liikkumiseen. Näistä seikoista johtuen tilaaja on päättänyt aloittaa uuden ympäristön kehittämisen, jota varten kuunnellaan opettajien toiveita. Ennen opinnäytetyön aloitusta tilaaja on tehnyt tuotteelle markkinatutkimuksen, jossa on selvennetty tuotteen potentiaali Suomen markkinoilla.

Opinnäytetyön tavoitteena on kehittää kouluissa käytettävä Internet-selaimella käytettävä ympäristö, jota voi käyttää kouluissa eri osapuolten väliseen yhteydenpitoon sekä oppilaiden kehityksen tarkkailuun. Työ on rajattu pelkästään selainpohjaisen ohjelmiston tekemiseen. Siihen ei kuulu ulkopuolisten sovellusten ohjelmointia.

2 MARKKINOILLA OLEVAT OPPIMISYMPÄRISTÖT

Maailmalla on olemassa lukemattomia ohjelmistoja koulujen käyttöön. Suomalaisissa kouluissa on kuitenkin pääosin käytössä tietyt samat ympäristöt, jotka ovat myös Suomessa kehitettyjä. Tällaisia ympäristöjä kutsutaan usein nimellä ”sähköinen reissuvihko”. Kaikista ympäristöistä löytyy samantyyppisiä viestintäominaisuuksia, jotka helpottavat yhteydenpitoa koulun sekä kodin välillä. Wilmaan ja Helmeen sisältyy lisäksi oppilashallintoon liittyviä toimintoja.

2.1 Wilma

Wilma™ on vuonna 1987 perustetun vaasalaisen StarSoftin kehittämä WWW-käyttöliittymä yrityksen muille ohjelmistoille. StarSoft on alansa markkinajohtaja. (2.) Sen sovelluksia käytti vuonna 2008 yli 1500 suomalaista koulua ja sen markkinaosuus vaihteli 70 %– 90 % välillä eri oppilaitoksissa (3). Wilma ei ole itsenäinen sovellus, vaan se käyttää hyödykseen yrityksen muita opiskelijahallintoon liittyviä tuotteita. Wilma toimii siis vain käyttöliittymä koulun henkilökunnalle sekä oppilaille ja huoltajille. Kuvassa 1 näkyy Wilman käyttöliittymä.



The screenshot shows the Wilma user interface for Marko Mallila. The top navigation bar is yellow and contains a star icon, the name 'Marko Mallila', and the text 'Oma etusivu'. On the left side, there is a vertical menu with the following items: Pikaviestit, Työjärjestys, Opinnot, Kokeet, Tuntimerkinnät, Tulosteet, Kyselyt, Kurssitarjotin, Tiedotteet, Opetussuunnitelma, and Opiskelijat. The main content area is titled 'Tervetuloa, Marko Mallila!'. It features two yellow warning boxes: the first says 'Sinulla on 3 tuntimerkintää, jotka on selvitettävä.' and the second says 'Ylläpito tiedottaa: Wilma-tunnukset'. Below these are sections for 'Pikaviestit' (showing 9 new messages, 9 from teachers) and 'Kurssit jaksossa' (listing courses like Englanti B1 ENB6, Kemia KE3, and Liikunta LD6).

KUVA 1. Oppilas on kirjautunut sisään Wilmaan

2.2 Helmi

Helmi™ on oululaisen vuonna 2000 perustetun Nextime Solutions Oy:n (4) kehittämä täysin Internet-pohjainen ratkaisu opetustoimen tarpeisiin (5). Se oli alun perin tunnettu Soneran sähköisenä reissuvihkona, mutta nykyään sitä markkinoidaan itsenäisenä tuotteena (6). Helmi ei tallenna tietojaan paikallisesti koulun palvelimelle, vaan tietokannat ovat täysin Nextimen hallinnassa (7). Tämä voi aiheuttaa ympäristöön ylimääräistä viivettä tai tietoliikenneyhteyksien pettäessä koko ympäristön kaatumisen. Toisaalta täysin keskitetyn ympäristön käyttöönotto on yksinkertaista eikä siihen vaadita koululta juurikaan ylimääräisiä resursseja. Kuvassa 2 on mainoskuva Helimestä, josta näkee, millainen sen käyttöliittymä on.

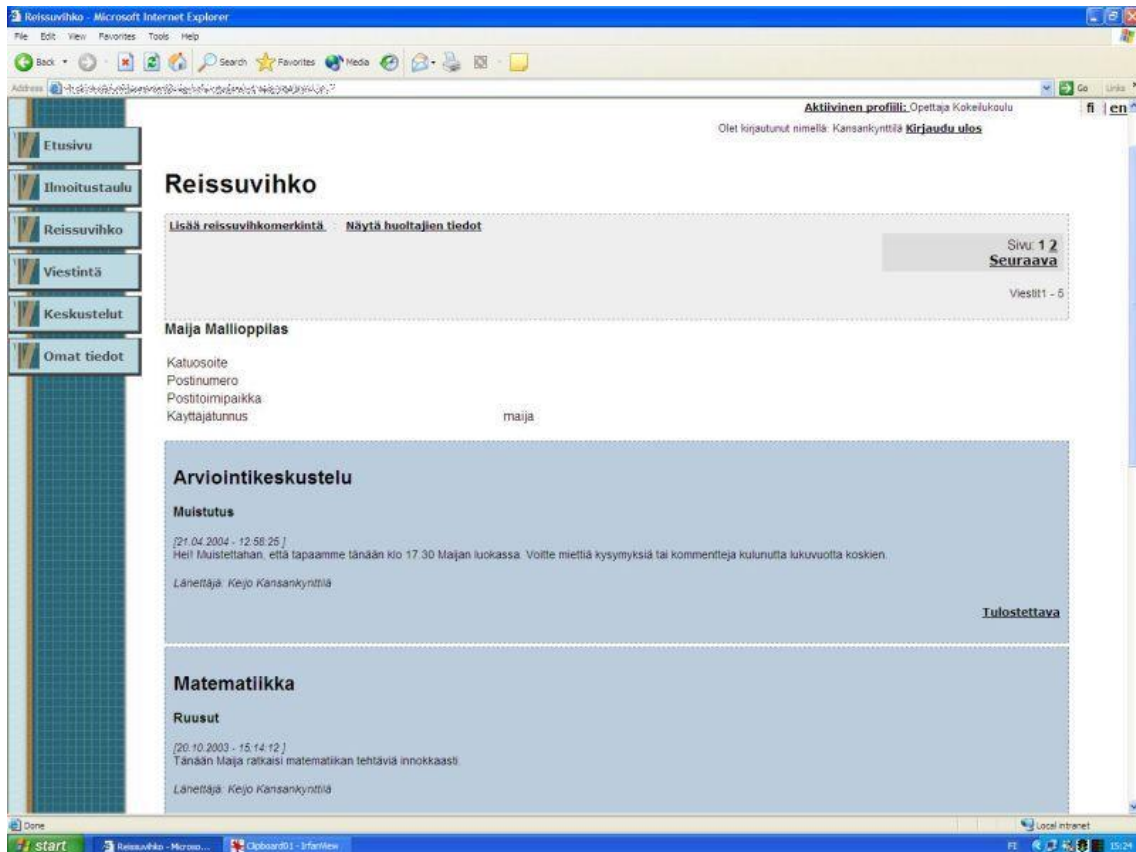


KUVA 2. Promootiokuva Helmistä (8)

2.3 Kivahko

Kivahko™-ohjelmisto on kehitetty osana MUKAVA-hanketta. Projektin tavoitteena on muistuttaa kasvatusvastuusta kaikkia niitä tahoja, jotka osallistuvat kouluikäisten lasten opetus- ja kasvatustyöhön. Projektissa on siis tavoitteena korostaa sosiaalisia taitoja nykyajan toimintaympäristössä. (9.) Kivahko tuli käyttöön vuonna 2002 ja sen saamasta palautteesta tuli ilmi, että tämän tyyppi-

sille sovelluksille on selvästi kysyntää. 73,4 % palautteesta antaneista henkilöistä piti välinettä tarpeellisena yhteydenpitokeinona (10). Kivahko on viestintäpöytäympäristö eikä siihen sisälly esimerkiksi kurssien hallintaa. Erikoisominaisuutena löytyy käyttömahdollisuus tekstiviesteillä. Kuvassa 3 näkyy opettaja kirjautuneena sisään Kivahkoon.



KUVA 3. Kivahkon reissuvihkonäkymä (11)

3 YLEINEN OHJELMISTOPROSESSI

Ohjelmistoprosessissa määritellään ohjelmiston elinkaari, jonka aikana se kehittyy tilaajan ideasta valmiiksi, käytettäväksi ohjelmistoksi (12). Prosessi eritellään viiteen eri vaiheeseen (kuva 4).



KUVA 4. Yleinen ohjelmistoprosessi (12)

Määrittelyvaiheessa päätetään, mitä toiminnollisuuksia valmiille tuotteelle halutaan. Siihen kuuluvat myös suorituskykyvaatimukset eli se, kuinka nopeasti ohjelmiston on suoritettava tietyt toiminnot, sekä laitteistovaatimukset eli se, millälaisessa ympäristössä ohjelmistoa halutaan käyttää. Määrittelyvaiheesta syntyy dokumentaatio, joka hyväksytetään tilaajalla. Näin varmistetaan se, että sekä tuottajalla että tilaajalla on yhteisymmärrys siitä, mitä valmiilta tuotteelta odotetaan. (13.)

Suunnittelussa päätetään ne menetelmät, joilla ohjelmistoa aletaan toteuttaa. Suunnitteluvaiheessa keskitytään teknisiin kysymyksiin, ja se on läheisessä yhteydessä siihen, kuinka itse ohjelmakoodia aletaan kirjoittamaan. (14.) Tässä vaiheessa päätetään, millä tekniikoilla edetään toteutusvaiheeseen. Tekniikoiden täytyy olla yhteensopivia tilaajan asettamien vaatimusten kanssa, eli

niiden tulee toimia halutuissa ympäristöissä luotettavasti sekä riittävän nopeasti. Tähän vaiheeseen kuuluu myös tarkempi käyttöliittymäsuunnittelu, eli pääte-tään, miltä ohjelmiston tuottamat näkymät näyttävät.

Kun määrittely- ja suunnitteluvaiheet ovat valmiita, pitäisi olla selkeä kuva siitä miten ohjelmistoa lähdetään toteuttamaan, kuinka sen tulee valmiina toimia ja miltä se tulee näyttämään. Seuraavaksi on siis toteutusvaihe, joka tarkoittaa ohjelmointityötä. Sitä suoritettaessa noudatetaan suunnitteluvaiheessa tehtyjä päätöksiä, joista selviää myös ohjelmiston rakenne. Vaiheen lopputuloksena on toimiva kokonaisuus, joka vastaa ennalta tehtyjä vaatimuksia sekä määrittelyjä. (15.)

Testausvaiheessa ohjelmistoa suoritetaan jonkin suunnitelman mukaan ja siitä yritetään etsiä virheitä. Käytännössä testausvaiheessa varmistetaan, että ohjelmistolle määrittelyvaiheessa asetetut vaatimukset toteutuvat. Näihin kuuluvat sekä halutut toiminnot että vaadittu suorituskyky. (16.) Testausvaiheessa löytyneet virheet tai puutteet on luonnollisesti korjattava.

Ylläpitovaiheessa toteutettu ja testattu ohjelmisto on tilaajan käytössä. Tässä vaiheessa ei enää pitäisi tulla ilmi vikoja, jos testausvaihe on suoritettu oikein. Ohjelmiston käyttäjä saattaa kuitenkin löytää käytön yhteydessä puutteita, joita ei ollut otettu huomioon ohjelmiston määrittelyvaiheessa. Tällöin ohjelmistoon voidaan tehdä lisäominaisuuksia kehittäjän toimesta erillisen sopimuksen mukaan. Uusia tarpeita tulee todennäköisesti ilmi joka tapauksessa tekniikoiden ja vaatimusten muuttuessa, ja sitä varten ohjelmiston kehittäjän kanssa solmi-taankin usein ylläpitosopimus. (17.)

4 SELAINPOHJAISTEN SOVELLUSTEN KEHITYSTYÖKALUT

Selainpohjaisten sovellusten kehittämiseen käytetään erikseen sitä varten luotuja tekniikoita. Tässä esitellyt tekniikat ovat laajasti käytössä olevia sekä Azurea lukuun ottamatta ilman korvausta kenen tahansa käytettävissä.

4.1 PHP

PHP on erityisesti WWW-sovellusten kehittämiseen luotu palvelinpohjainen skriptikieli (18). Lyhenne tulee alun perin sanoista Personal Home Page, mutta nykyään nimen kerrotaan olevan PHP: Hypertext Preprocessor. PHP ei ole alustariippuvainen, joten sillä tehtyä koodia voi ajaa lukuisissa eri käyttöjärjestelmissä.

PHP:n lähdekoodi on vapaasti saatavilla ja kielen käyttö on täysin ilmaista. Vapaan käytön ohella PHP:n etuna on myös sen avoimesta luonteesta johtuva aktiivinen kehittäjäyhteisö, josta saa ongelmien ilmaantuessa apua.

PHP-koodia voi kirjoittaa millä tahansa tekstieditorilla. Sen suorittamiseen vaaditaan palvelin, johon on asennettu PHP-tulkki (19). Koodin suorittamiseen ei siis tarvitse erillistä kääntäjää tai ohjelmointiympäristöä, sillä palvelin ajaa koodin aina käyttäjän sitä pyytäessä ja esittää tuloksena käyttäjälle näkymän. Hyvänä puolena tässä on se, ettei käyttäjä pääse näkemään koodia vaan se pysyy aina omistajan palvelimella turvassa.

4.2 MySQL

MySQL on nykyisin Orcalen omistama, maailman käytetyin relaatiotietokantaohjelmisto (20). Se on avoimen lähdekoodin ohjelmisto, josta on saatavilla monta eri versiota, joista jotkut maksavat. Community Server-versio on ilmainen ja sitä ylläpitää aktiivinen käyttäjäyhteisö (21).

Ensimmäisen julkisen version MySQL:stä julkaisi vuonna 1996 MySQL AB-yritys. Vuonna 2000 ohjelmisto vaihtoi vapaan koodin lisenssiin, mikä vaikutti alussa negatiivisesti yrityksen taloudelliseen tilanteeseen. Siitä huolimatta ohjelmiston suosio nousi vahvasti, ja rahoitusta sekä maksavia asiakkaita liittyi

runsaasti mukaan MySQL-leiriin. Java-kehityksestä parhaiten tunnettu Sun Microsystems osti MySQL:n vuonna 2008, ja hieman tämän jälkeen MySQL:n alkuperäiset perustajat jättivät Sunin, koska olivat tyytymättömiä siihen miten yritys toimii. Vuonna 2009 myös omasta tietokantaohjelmistostaan tuttu Oracle osti Sun Microsystemsin, ja näin MySQL päättyi lopulta Oraclen haltuun. (22.)

4.3 Windows Azure

Windows Azure on Microsoftin kehittämä pilvipalvelu, jonka kautta voi laittaa saataville itse rakennettuja sovelluksia pitkin maailmanlaajuista verkostoa, jota ylläpitää Microsoft (23). Pilvipalvelun käyttämisen etuna on se, ettei sovelluksen tekijän tarvitse erikseen huolehtia riittävän palvelintilan hankkimisesta, vaan Azure skaalautuu automaattisesti sovelluksen vaatimiin resursseihin.

Azure julkaistiin alun perin tammikuussa vuonna 2010 (24). Siitä lähtien se on saanut jatkuvasti uusia ominaisuuksia, ja nykyään PHP- ja MySQL-tuen ansiosta se on varteenotettava vaihtoehto myös avoimen koodin teknologioilla kehitetyille sovelluksille.

4.4 HTML

Selaimissa käyttäjille näkyvät asiat esitetään HTML- ja CSS tekniikoilla. HTML on standardoitu kuvauskieli, jota WWW-selain tulkitsee ja näyttää tuloksena käyttäjälle WWW-sivun. Sitä ylläpitää kansainvälinen organisaatio nimeltä World Wide Web Consortium (lyhennettynä W3C). W3C koostuu 387:stä eri jäsenorganisaatiosta (25) ja sitä johtaa Tim Berners-Lee. HTML-koodi koostuu erinäisistä kulmasulkein merkityistä tunnisteista. Tunnisteiden yhteyteen kirjoitetaan se tieto, minkä ohjelmoija haluaa esittää käyttäjälle. HTML:n kirjoittamiseen soveltuu mikä tahansa tekstieditori, koska sen koodia ei tarvitse ajaa kääntäjän läpi. WWW-selaimen ohjelmakoodista riippuu, miten HTML:nä kirjoitettu koodi käyttäjälle esitetään (26).

4.5 CSS

CSS on HTML:n kanssa käytettävä tyylikieli, jonka avulla voidaan määrittää erinäisiä tyylejä HTML-dokumenteille. CSS-standardi on HTML:n tapaan W3C:n

ylläpitämä. Sen idea on esittää dokumentin sisältö erillään siitä tiedosta, kuinka sisältö esitetään käyttäjälle. (27.) CSS:llä voidaan määrittää esimerkiksi painikkeiden käyttäytymistä, dokumentin värejä ja fontteja sekä tekstin asettelua. Käytännössä CSS:stä on runsaasti hyötyä ohjelmoijalle, koska hän voi viitata ennalta luotuihin tyylihin mistä tahansa HTML-dokumentista. Vaaditaan vain se, että dokumentin alussa on määritelty käytettävä CSS-tiedosto.

Itse CSS-tiedosto kirjoitetaan tietynlaisen syntaksin eli ohjelmointikielen lauseopin mukaan, joka muistuttaa hyvin paljon myös monissa muissa suosituissa ohjelmointikielissä käytettyä tyyliä. CSS-syntaksin alkuun kirjoitetaan valitsimen nimi, johon viitataan HTML-dokumenteissa. Nimen perusteella siis valitaan, mitä tyyliä halutaan kulloinkin käyttää. Valitsimen jälkeen kirjoitetaan aaltosulkujen sisään tyylin ominaisuudet ja arvot. Esimerkkinä voidaan käyttää kirjaisintyyppin määrittelyä. Tällöin ominaisuudeksi kirjoitetaan "font" ja sen jälkeen sille halutut arvot eli kirjaisintyyppin koko sekä nimi. (28.)

5 PÄÄTARKOITUS JA TAVOITTEET

Työn päätarkoituksena on kehittää tilaajalle ohjelmisto, josta voi tulevaisuudessa muodostua potentiaalinen kilpailija nykyisille kouluissa käytössä oleville ratkaisuille. Työhön toteutetaan oppimisympäristön ydinominaisuudet, joiden pohjalta tilaajan on mahdollista hankkia lisäpalautetta opettajilta ja tietoa siitä, kuinka ympäristöä kannattaa kehittää eteenpäin, jotta siitä tulee parempi kuin olemassa olevat ratkaisut. Kehitetty ympäristö tulee olemaan myös avoin muille päätelaitteille kuin PC, johon tämän työn kehitys on rajattu.

Työn tavoitteet jakaantuvat neljään eri ohjelmistoprosessin vaiheeseen. Ensimmäisenä tavoitteena on luoda yhteistyössä tilaajan kanssa määrittelydokumentaatio, josta käy selväksi valmiilta työltä odotettavat tulokset. Seuraavana toteutetaan itsenäisesti läpikäytävä suunnitteluvaihe, jossa tavoitteena on päättää millä menetelmillä ohjelmistoa aletaan toteuttaa. Kolmas vaihe on varsinainen toteutus eli tarvittavien ohjelmakoodien kirjoitus. Toteutusvaiheen tavoitteena on saada valmiiksi kaikki toiminnot, jotka määrittelyvaiheessa on päätetty toteutettavaksi. Viimeisenä vaiheena on valmiin kokonaisuuden testaus, jossa varmistetaan, ettei ohjelmistossa ole vikoja. Vaikka ympäristöä testataan samalla kun sitä kehitetään, loppuun on syytä eritellä vielä erikseen oma testausvaihe, jossa varmistetaan vielä lopullisesti tilaajan kanssa kokonaisuuden toimiminen halutulla tavalla.

6 OPPIMISYMPÄRISTÖN KEHITYSPROSESSI

Oppimisympäristön kehitysprosessin vaiheet kuvaavat kuinka kukin ohjelmistoprosessin vaiheista toteutui tämän työn kohdalla. Prosessiin kuului neljä vaihetta.

6.1 Määrittely

Työtä varten tehtiin ensimmäiseksi tilaajan kanssa alustava vaatimusmäärittely, joka on yhteenveto ominaisuuksia jotka tilaaja odottaa löytävänsä valmiista ohjelmasta. Vaadittavat ominaisuudet jaettiin neljään eri ryhmään.

6.1.1 Tietokannan teko

Tietokanta on SQL-kielinen ja toimii Azure-pilvipalvelussa. Käyttäjien salasana-tiedot täytyy varastoida salatussa muodossa tietoturvan takia. Tilaajalle tehdään valmis SQL-komentosarja, jotta ympäristön asentaminen ja tietokannan alustaminen onnistuvat helposti.

Tietokantaan on mahdollista päästä käsiksi myös mahdollisten tulevien sovellusten avulla. Esimerkiksi ympäristöön liitetty mobiilisovellus voi käyttää hyväkseen samoja käyttäjätilejä kuin nyt tehtävä sivusto.

6.1.2 Käyttäjien hallinta

Käyttäjiiä on neljää eri tyyppiä: järjestelmänvalvoja, opettaja, oppilas ja huoltaja. Käyttäjillä on eri oikeudet tehdä ympäristöön muutoksia tai hakea sieltä tietoja.

Järjestelmänvalvoja-tiliä on vain yksi kappale. Sen kautta voi hallinnoida muun tyyppisiä tilejä sekä muuttaa sivuston asetuksia. Järjestelmänvalvoja voi luoda uusia kursseja ympäristöön. Se voi myös liittää opettajia ja kursseja yhteen sekä lisätä oppilaita tietyille kurssille.

Opettajilla on omat tilit, joiden kautta he voivat arvostella kursseja. Oppilas voi tarkastella omia kurssejaan ja arvosanojaan. Opettajat voivat myös tehdä muis-tiinpanoja oppilaista. Huoltajille voidaan luoda tilejä, jotka on liitetty tiettyyn oppi-

laaseen. Sen kautta he voivat tutkia oppilaan arvosanoja ja kursseja. Kaikilla tileillä on viestintämahdollisuus.

Käyttäjän tason perusteella rajoitetaan hänen pääsyään ympäristöön. Aina kun käyttäjä tekee jotain, hänen arvonsa tarkastetaan ja hänelle näytetään sen mukainen näkymä. Jokaisen käyttäjän valikko näyttää erilaiselta. Toiminnot, joihin käyttäjällä ei ole pääsyä, piilotetaan.

6.1.3 Kurssien hallinta

Ympäristöön voi lisätä kursseja järjestelmänvalvojan tilin kautta. Kurseille määrätään tietokannasta löytyvä vastaava opettaja. Oppilaat lisätään kursseille ja sen jälkeen heitä voidaan arvioida opettaja-tilillä tarpeen mukaan. Yhden oppilaan voi lisätä samalla kertaa monelle kurssille valitsemalla halutut kurssit listalta.

SQL-lausekkeiden avulla tietoa voidaan hakea niin, että liitokset voidaan esittää käyttäjille heidän tarpeidensa mukaan. Esimerkiksi kurssin tiedot ja oppilaan arvosanat sijaitsevat eri taulukoissa, ja SQL-lauseiden avulla nämä tiedot voidaan esittää yhdessä samassa näkymässä.

6.1.4 Viestintäjärjestelmä

Sivustolle tehdään järjestelmä, jossa käyttäjät voivat lähettää viestejä toisilleen. Viestille valitaan vastaanottaja, sille kirjoitetaan otsikko ja sitten itse viesti.

Viestit ryhmitellään kolmeen eri kansioon. Saapuneille sekä lähetetyille viesteille on oma kansionsa ja lisäksi on erillinen roskakori-kansio. Poistetut viestit siirtyvät ensimmäiseksi roskakoriin, josta ne voidaan jälkeinpäin poistaa lopullisesti. Tietokantausekkeiden avulla eritellään kuka on viestin vastaanottaja, ja viestit näytetään aina vain oikeille henkilöille.

6.2 Suunnittelu

Kun määrittelyvaihe oli valmis, siirryttiin suunnitteluvaiheeseen ennen kuin ympäristön vaatimukset toteuttavaa ohjelmistoa alettiin kirjoittaa. Suunnitteluvaiheessa otettiin aluksi selvää eri tekniikoista ja menetelmistä, joilla vaatimuksien

mukainen ohjelmisto on mahdollista toteuttaa. Tämän oppimisympäristön vaatimusten toteuttamiseen vaadittiin tekniikoita, jotka toimivat yhdessä Azure-pilvipalvelun kanssa. Ennen ohjelmointivaiheeseen siirtymistä olikin syytä varmistaa sekä teoriassa että myös käytännössä, että tekniikat joilla ohjelmistoa aletaan toteuttamaan toimivat odotetulla tavalla. Kannatti siis luoda jo suunnitteluvaiheessa pieni kaikkia tekniikoita hyödyntävä ohjelma ja kokeilla sitä vaaditussa ympäristössä, jotta vältettiin mahdolliset ikävät yllätykset tulevissa vaiheissa.

Suunnitteluvaiheeseen kuului myös ohjelmiston rakenteen määrittelemineen. Tässä päätettiin kuinka ohjelmakoodi kirjoitetaan niin, että sitä on mahdollisimman helppo ylläpitää ja kehittää eteenpäin. Käytännössä tämä sisälsi päätökset ohjelmiston luokkarakenteesta sekä siitä, kuinka ohjelmakoodi ryhmitellään eri tiedostoihin.

6.3 Ohjelmointi

Kun suunnitteluvaihe oli tehty huolellisesti, itse ohjelmointi oli melko suoraviivaista toimintaa. Ohjelmointivaiheessa tarvitsi vain kirjoittaa ennalta määritetyn rakenteen sisälle koodia, jonka tuloksena syntyi ohjelmisto joka toteuttaa oppimisympäristölle asetetut vaatimukset. Ohjelmoinnin apuna toimivat valittuja tekniikoita varten luodut julkiset dokumentaatiot, joiden avulla tekniikoista löytyviä menetelmiä osattiin käyttää tarvittaessa ja näin tehdä ohjelmointityöstä mahdollisimman nopeaa ja sujuvaa. Toimintojen valmistuessa niitä myös samalla testattiin, mikä oli tärkeää varsinkin kun ohjelmistoa kehitettiin itsenäisesti ilman erillistä testausryhmää.

6.4 Testaus

Ohjelmistoa testasivat sekä tilaaja että kehittäjä. Kun kaikki vaatimusmäärittelyn mukaiset toiminnot saatiin valmiiksi, ohjelmisto siirrettiin Azure-pilvipalveluun ja sen toimintoja kokeiltiin vaatimusmäärittelyn mukaisesti. Samalla myös tilaajalle annettiin mahdollisuus tutustua itse ohjelmistoon ja testata sitä.

Toimintoja suorittaessa tutkittiin samalla tietokannassa tapahtuvia muutoksia ja näin varmistettiin, että ne toimivat halutulla tavalla. Ympäristöön tehtiin useita

eri käyttäjiä ja heidän välilleen luotiin yhteyksiä (esimerkiksi viestintään tai kurssiin liittyviä). Näin varmistettiin se, että ohjelmisto voi toimia myös oikeassa ympäristössä koulussa, jossa on paljon henkilökuntaa sekä oppilaita. Myös toimintojen suorittamiseen kuluvaan aikaan kiinnitettiin huomiota. Mikään testauksen aikana suoritettava toiminto ei vienyt huomattavan paljon aikaa. Tosin vaihtelua voi esiintyä tietoliikenneyhteyksistä johtuen.

Käyttöliittymän toimivuus varmistettiin käyttämällä useita selaimia. Ohjelmistoa on testattu selaimilla Internet Explorer 10, Firefox 19, Google Chrome 25 sekä Opera 12. Joidenkin painikkeiden ulkoasu saattoi olla eri selaimista riippuen hieman erilainen, mutta käytettävyyden kannalta sillä ei ollut merkitystä. IE:ssä havaittiin myös jonkun tekstikentän olevan erikokoinen kuin normaalisti, mutta toiminnollisuuteen liittyviä eriävyyksiä testauksen aikana ei löydetty.

7 OPPIMISYMPÄRISTÖN TOTEUTUS

Ympäristön toteuttamiseen käytettiin jatkuvaa SQL-tietokantayhteyttä tietojen tallentamiseen sekä olemassa olevien tietojen lukemiseen. Ohjelmiston ja näkymien toteuttamiseen käytettiin PHP:stä löytyviä ominaisuuksia sekä HTML-kieltä. HTML:n kanssa hyödynnettiin CSS-tyylitiedostoja, jotta sivustolle saatiin luotua jonkin verran visuaalista ilmettä.

7.1 Käytetyt tekniikat

Työ kehitettiin Windows-käyttöjärjestelmän alla, mutta itse koodi on alustariippumatonta. Kehitysvaiheessa työtä ohjelmoitiin ja testattiin Apache-palvelimen avulla. Tilaja halusi tuotteen toimivan Microsoftin Azure-pilvipalvelussa, mutta itse kehitystyö tehtiin paikallisesti omassa ympäristössä, koska tällöin työn uutta versiota ei aina tarvitse lähettää Azure-palvelimelle jotta sitä voi testata. Molemmat ympäristöt ovat kuitenkin yhteensopivia keskenään, sillä palvelinta tarvitaan pelkästään PHP-koodin ajamiseen eikä sillä ole olennaista merkitystä ohjelmiston toiminnan kannalta.

Työ on ohjelmoitu PHP-kielen 5.4 versiota käyttäen. PHP:lle on saatavilla runsaasti tukea eri verkkoyhteisöistä, mikä oli tärkeä seikka koska kokemusta PHP-ohjelmoinnista tämän työn edellyttämässä mittakaavassa ei juurikaan ollut. Tietokantaohjelmistona käytettiin MySQL:ää. Valinta oli helppo, koska kyseinen ohjelmisto oli ennestään tuttu ja se on helppo asentaa kehitysympäristöön. Myös Azure tukee MySQL-tietokantaa.

Työn toteuttamiseen MySQL valittiin ennen kaikkea siksi, että sen käyttö oli tullut tutuksi jo aiemmin. Tämä oli erityisen tärkeää jotta pystyttiin olemaan varmoja siitä, että jo opitut asiat SQL-kielestä pitävät paikkansa. Muita tietokantaohjelmistoja käytettäessä olisi saattanut joutua opiskelemaan pelkästään näille ohjelmistoille ominaisia toimintoja, mikä olisi vaikeuttanut työtä entisestään. Avoimen lähdekoodin ja pitkän historian ansiosta MySQL on myös hyvin tuettu käyttäjäyhteisöissä, joten esimerkkien ja avun saaminen olisi helppoa. Myös se, että Azure tukee MySQL-tietokantoja, oli projektin kulkua helpottava yksityiskohta.

7.2 Tietokanta

Ohjelmiston tietokanta on jaettu useaan eri taulukkoon. Taulukoiden tietoja yhdistelemällä ohjelma pystyy näyttämään käyttäjälle hänen tarvitsemansa tiedot yhdellä kertaa.

7.2.1 Käyttäjätilit

Kaikilla käyttäjillä on oma rivinsä käyttäjät-taulukossa. Taulukkoon sisältyy käyttäjien ID-numero, jonka avulla PHP-ohjelmisto yksilöi ja tunnistaa kunkin käyttäjän. ID-numeron avulla vältetään tilanteet, joissa samannimiset käyttäjät voisivat mennä sekaisin. ID-kenttä käyttää MySQL:n ”auto increment”-toimintoa, joka uuden käyttäjän luotaessa lisää automaattisesti aina yhden numeron edellisen käyttäjän ID-numeroon (29). Kyseisen ominaisuuden käyttäminen ei vaadi ollenkaan lisäohjelmointia, sillä pelkästään ID-kentän määrittäminen taulukon luontivaiheessa riittää.

Seuraavina kenttinä ovat käyttäjänimi ja salasana. Näitä tietoja käytetään itse sisäänkirjautumisprosessissa. Jokaiseen käyttäjänimeen on yhdistetty salasanakenttä. Salasana on säilötty tietokantaan salatussa muodossa, joka estää mahdollisia väärinkäytöksiä. Jos esimerkiksi jostain syystä ohjelmiston tietokanta vuotaa julkisuuteen, ympäristön kirjautumistietoja ei voi käyttää hyväkseen, koska salasanat eivät ole luettavassa muodossa. Kun käyttäjä yrittää kirjautua sisään ympäristöön, hänen kirjautumisruudussaan syöttämä salasana muunnetaan salattuun muotoon ja sitä verrataan tietokantaan. Jos molemmat täsmäyvät, käyttäjä päästetään sisään ympäristöön.

Tyyppi-kentästä käy ilmi käyttäjän rooli ympäristössä. Numero 1 tarkoittaa oppilasta, 2 opettajaa, 3 järjestelmänvalvojaa sekä 4 huoltajaa. Tyypinnumero ei näy missään vaiheessa ympäristön käyttäjälle, vaan se on laitettu tietokantaan pelkästään ohjelmiston toimivuutta ajatellen. PHP-ohjelmisto käyttää tyyppinumeroa tunnistukseen, mitkä oikeudet kullakin käyttäjälle on ympäristöön. Sen mukaan joko rajoitetaan tai sallitaan tietyt toiminnot, joita käyttäjä yrittää tehdä.

Lopuksi taulukosta löytyy perustietoja käyttäjästä. Niihin kuuluvat etu- ja sukunimi sekä sukupuoli. Järjestelmänvalvojalla kyseiset kentät eivät ole oleellisia. Käyttäjälle on myös voitu määrittää sähköpostiosoite. Tosin se ei ole pakollinen.

Tietokannasta löytyy oma taulukko huoltajille. Tämä taulukko on välttämätön, jotta oppilaat voidaan liittää oikeisiin huoltajiin. Taulukosta löytyy huoltajan käyttäjätunnuksen ID, jota vastaava numero löytyy käyttäjät-tilukosta. Tämän jälkeen taulukossa on huoltajaan liitetyn oppilaan ID, joka niin ikään löytyy myös käyttäjät -tilukosta. Kun huoltaja kirjautuu järjestelmään sisään, tietokannasta tarkistetaan minkä oppilaan huoltaja hän on ja huoltajalle esitetään tämän oppilaan tiedot.

Myös oppilaita varten on oma taulukkonsa. Taulukosta löytyy kaksi tietuetta. Toisessa on oppilaan ID-numero ja toisessa käyttäjästä tehdyt muistiinpanot. Muistiinpanot-kenttää voi tarkastella sekä muuttaa järjestelmään luodut opettajat.

7.2.2 Kurssit

Jokaiselle luodulle kurssille löytyy oma rivinsä kurssit-tilukosta. Kun uusi kurssi luodaan, sille määritellään ID edellä mainitun ”auto increment”-toiminnon avulla. Lisäksi kurssille täytyy määritellä jokin nimi. Kurssille täytyy myös määrittää jokin vastaava opettaja, jonka käyttäjä-ID löytyy tästä taulukosta. ID:tä vastaava opettaja löytyy luonnollisesta käyttäjät-tilukosta, jonka kautta opettajan nimen voi esittää kurssin yhteydessä.

Kurssien arvosanat sijaitsevat erillisessä taulukossa. Tämä on välttämätöntä, jotta arvosanat saadaan yhdistettyä oikeisiin oppilaisiin. Kurssit ja oppilaat yhdistävästä taulukosta löytyy kolme kenttää. ID-kentässä on sen oppilaan käyttäjä-ID, jolle arvosana kuuluu. Kurssi-kentästä löytyy kurssin ID, ja kurssin nimen saa haettua vertaamalla ID:tä kurssit-tilukkoon. Viimeisenä löytyy arvosana-kenttä, johon opettaja voi tallentaa joko kirjallisen tai numerollisen arvostelun.

7.2.3 Viestintä

Käyttäjien välisen viestinnän ylläpitämiseen riittää yksi taulukko. Aluksi taulukosta löytyy viestin ID, jonka kautta ympäristö saa eroteltua kunkin viestin ja tiettyä ID:tä pyydetessä tietokanta osaa toimittaa oikean viestin käyttäjälle. Seuraavassa kentässä on sen käyttäjän ID, jolta viesti on lähtöisin. Tätä seuraa sen käyttäjän ID, jolle viesti on tarkoitettu. Kun käyttäjä tarkistaa viestikansionsa, hänelle esitetään tietokannan avulla kaikki viestit, joissa hänen ID on merkitty vastaanottajaksi. Vastaavasti käyttäjän tarkistaessa hänen lähetettyjä viestejään hänelle esitetään ne viestit, joissa hänen ID on merkitty lähettäjä-kenttään.

Seuraavat kolme kenttää sisältävät itse viestin sekä sen lähetysajan. Ensimmäisenä on viestin otsikko ja sitä seuraa itse viesti. Lähetysaika merkitään UNIX-aikana. Se ilmoittaa ajan sekunteina vuoden 1970 ensimmäisestä hetkestä UTC-aikavyöhykkeen mukaan (30).

Lisäksi taulukosta löytyy kaksi kenttää, jotka vaikuttavat siihen kuinka viesti arkistoidaan. Viestit on merkitty joko luetuiksi tai lukemattomiksi. Alun perin viestin luku-arvo on 0. Kun käyttäjä avaa ensimmäistä kertaa viestin, arvo muuttuu 1-luvuksi. Tämän ansioista ohjelmisto voi näyttää lukemattomat ja luetut viestit käyttäjälle eri tavalla. Viesteillä on myös poistettu-asetus, jonka arvo on joko 0 tai 1. Jos arvo on 1, viesti esitetään käyttäjälle roskakori-kansiossa. Käyttäjän pitää siis merkata poistettavat viestit roskakori-kansioon, ennen kuin hän voi hävittää ne tietokannasta lopullisesti.

7.3 Näkymät

Toteutetussa ohjelmistossa on yksi CSS-tyylitiedosto ja HTML-dokumentit sisältyvät PHP-tiedostoihin. Yhteen CSS-tiedostoon voi sisällyttää loputtomasti eri tyytlejä, joihin voi viitata ohjelmiston tarpeiden mukaisesti aina tarvittaessa. Useampien tiedostojen käyttö olisi perusteltua jos erilaisia tyytlejä olisi niin paljon, että CSS-tiedoston koko kasvaisi suhteettoman suureksi. Tällöin ohjelmisto aiheuttaisi turhaa nettiliikennettä joka kuormittaisi sekä tilaajan että asiakkaan järjestelmiä. Tämän ohjelmiston valmis CSS-tiedosto on kuitenkin vain vajaan kahdeksan kilotavun kokoinen, joten tyylien erittely eri tiedostoihin ei ole tar-

peellista. Kyseiseen tyylitiedostoon on määritelty painikkeiden ja valikoiden käyttämät värit sekä se, kuinka ne käyttäytyvät kun hiiri vieään tietyn elementin päälle. Esimerkiksi painikkeiden sekä tekstikenttien reunojen pyöreys on määritetty CSS-tiedostosta löytyvien rounded-nimisten valitsimien sisään.

7.4 Ohjelmointi

Ohjelmiston toiminnollisuuksien luomisessa käytettiin hyväksi runsaasti PHP:n menetelmiä. PHP on suunniteltu tämän tyyppisiä järjestelmiä varten, joissa toimii useita eri käyttäjiä yhtä aikaa ja jotka haluavat jatkuvasti sekä lukea että syöttää tietoa järjestelmään.

7.4.1 Istunto

Kun käyttäjä kirjautuu tunnuksillaan järjestelmään, hänelle luodaan oma sessio eli istunto. Istunnon luominen PHP:ssä on erittäin helppoa. Sitä varten tarvitsee vain määrittää halutut arvot `$_SESSION`-nimiselle muuttujalle, jonka perusteella PHP luo automaattisesti käyttäjälle oman istunnon. Käytännössä palvelin luo jokaiselle istunnolle oma ID-tunnuksen. ID-tunnus välitetään käyttäjän selaimelle joko URL-osoitteen mukana tai evästeenä. Eväste on eräänlainen WWW-selaimen hallinnoima tiedosto, jonne voidaan automaattisesti säilöä muistiin käyttäjän tietoja tietyn WWW-sivuston osalta. Se, kumpaa tapa käytetään, riippuu palvelimen PHP-asetuksista. Tähän asiaan ei vaikuteta itse ohjelmakoodissa. Kun sisään kirjautunut käyttäjä toimii oppimisympäristössä, hänen selaimensa lähettää joka toiminnon yhteydessä palvelimelle istunnon ID-tunnuksen. Jos palvelin tunnistaa sen, käyttäjän toiminto voi jatkua. Jos ID-tunnusta ei löydy palvelimelta, tulkitaan se niin, ettei käyttäjä ole kirjautunut sisään ja hänelle esitetään kirjautumisruutu.

Ohjelmistossa istunnon tietoihin tallennetaan käyttäjästä neljä asiaa. Niihin kuuluvat käyttäjätunnuksen tyyppi eli mitkä oikeudet käyttäjällä on järjestelmään. Lisäksi tallennetaan käyttäjän ID-numero, mikä on siis eri asia kuin istunnon ID-tunnus jonka hallinta tapahtuu automaattisesti. Istuntoon tallennetaan myös käyttäjän etu-, suku-, sekä käyttäjänimi. Koska nämä tiedot on tallennettu istunnon tietoihin, niitä voidaan käyttää niin kauan kunnes käyttäjä kirjautuu ulos jär-

jestelmästä. Käytännössä tämä seikka mahdollistaa sen, että käyttäjä voi toimia ympäristössä tai tehdä muutoksia siihen. Useimmat toiminnot vaativat joko käyttäjän ID:n tunnistamista tai hänen oikeuksiensa tarkistamista.

7.4.2 Turvallisuus

Kun käyttäjä etenee kirjautumisprosessissa, ohjelma ensimmäiseksi muuntaa käyttäjän syöttämän salasanan MD5-muotoon. Muunnosta varten ohjelmasta löytyy erillinen funktio. MD5 on salausalgoritmi (31), jonka avulla estetään salasanan lukeminen suoraan tietokannasta. Esimerkiksi sana "selain" on MD5-muodossa merkkijono "2357228a8348e791ed278bb40b7d1983". Kun muunnos on valmis, kirjautumisprosessi jatkuu ohjelman ottamalla yhteyden tietokantaan. Jos tietokannan käyttäjät-taulukosta löytyy rivi, joka vastaa syötettyä käyttäjänimeä ja siihen liitettyä salasanaa MD5-muodossa, käyttäjä päästetään järjestelmään sisään. Jos syötetyt tiedot ovat väärinä, käyttäjälle näytetään vain virheviesti.

Ohjelmistossa on myös otettu huomioon SQL-injektoiden mahdollisuus. Kyseisessä hyökkäyksessä ohjelmistolle lähetettyjä tietokantapyyntöjä muokataan niin, että käyttäjä yrittää päästä käsiksi tietoihin joihin hänellä ei olisi oikeasti lupa päästä (32). Hyökkäykseltä suojautuminen edellyttää, että kaikki käyttäjän syöttämä tieto tarkistetaan ohjelmakoodissa. Käyttäjän syöttämiin tietoihin kuuluvat esimerkiksi käyttäjänimi, sähköpostiosoite tai kaikki, mille ohjelmistossa on olemassa erillinen tekstikenttä. Käytännössä tietokantakyselyt siis tarkistetaan ennen niiden suoritusta ja näin varmistetaan, että kysely on kirjautuneelle käyttäjälle sallittu eikä sitä ole yritetty muuttaa sellaiseksi, joka antaisi ylimääräisiä tietoja vastauksena.

7.4.3 UNIX-aika

Ohjelmisto käyttää aika- ja päiväystietojen varastointiin UNIX-aikajärjestelmää. Kyseinen tapa ajan tallentamiseen on hyvin yleinen eri käyttöjärjestelmissä, vaikkei sitä käyttäjälle semmoisenaan näytetäkään. Aika muunnetaan luonnollisesti ohjelmakoodissa käyttäjän helposti ymmärrettävään muotoon. Tätä merkintätapaa käyttäessä on otettava huomioon se, että vuoden 2038 tammi-

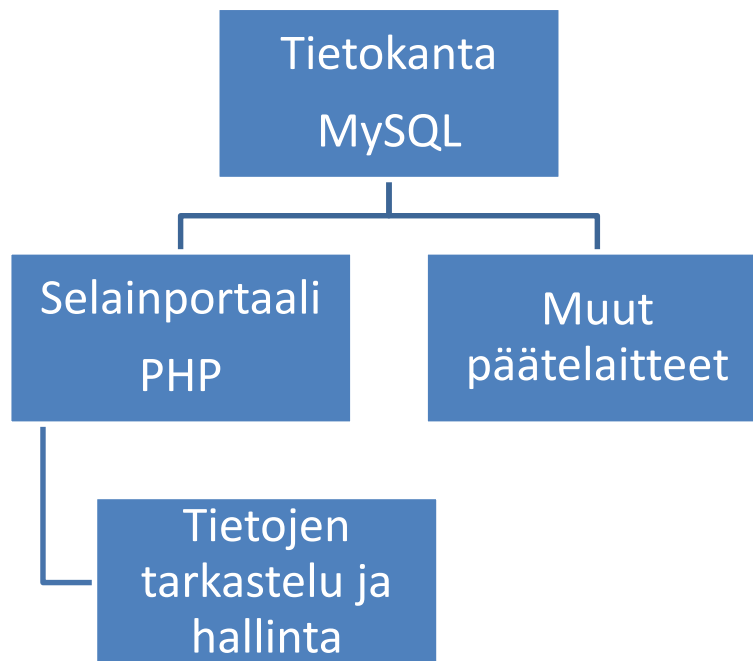
kuussa UNIX-aika ei toimi enää oikein 32-bittisissä järjestelmissä. Se johtuu siitä, että luvun varastointiin käytetystä tietotyypistä loppuu tila kesken, ja laskuri pyörähtää ympäri negatiiviseksi. (33.) Oletettavasti kuitenkin kyseisenä vuonna 32-bittisiä järjestelmiä ei enää ole käytössä vaan käyttöjärjestelmät on jo päivitetty 64-bittisiksi. 64-bittisissä järjestelmissä tietotyyppi on niin suuri, että se riittää sadoiksi miljardeiksi vuosiksi.

8 TOTEUTETUN OHJELMISTON KUVAUS

Ympäristön arkkitehtuuri on jätetty avoimeksi, jotta siihen voi liittää uusia sovelluksia mahdollisimman helposti. Valmis ohjelmisto toimii moniportaisen rakenteen avulla, joka varmistaa oikean toiminnollisuuden lisäksi sen, ettei ympäristöä yritetä käyttää väärin.

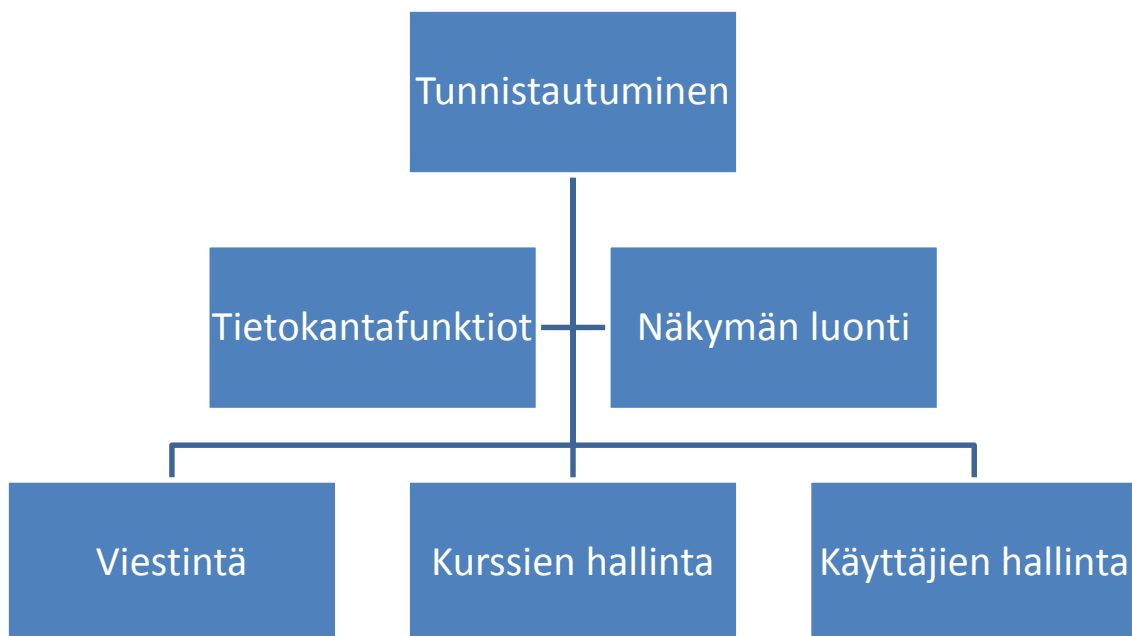
8.1 Ohjelmiston rakenne

Lopullinen PHP-ohjelmisto on jaettu useaan eri tiedostoon. Osa tiedostoista sisältää myös HTML-koodia, ja nämä tiedostot ovat vastuussa siitä mitä käyttäjälle näytetään. Ohjelmistoon kuuluu myös tiedostoja, joissa on pelkästään erinäisiä tarvittavia funktioita, joita muut tiedostot voivat kutsua tarvittaessa. Järjestelmään toteutettiin tietokanta, joka on koko järjestelmän ydin ja johon voi liittyä tulevaisuudessa muutenkin kuin nyt toteutetun selainportaalien avulla. Kuvassa 5 esitellään oppimisympäristön toteuttavan ohjelmiston arkkitehtuuri, josta selviää sen eri osat.



KUVA 5. Ympäristön arkkitehtuuri

Kuvassa 6 esitellään ohjelmiston rakenne siinä järjestyksessä, missä sen toiminnot suoritetaan. Ohjelman koodi on eritelty erinäisiin tiedostoihin, jotka käyvät pääosin ilmi kuvasta. Kun käyttäjä haluaa suorittaa jonkin toiminnon, on ylimpänä hierarkiassa tunnistautumisfunktioiden käsittely. Näiden avulla varmistetaan, että käyttäjällä on pääsy ympäristössä siihen toimintoon, mitä hän yrittää suorittaa. Seuraavana joko haetaan tietokannasta tarvittavat tiedot tai syötetään sinne uutta tietoa, joiden perusteella voidaan myös luoda näkymä. Kaikki toiminnot eivät välttämättä edellytä tietokantayhteyttä, joten näkymän luonti ei ole riippuvainen tietokantafunktioista. Hierarkian alimpana löytyvät järjestelmän ominaisuudet, jotka voidaan toteuttaa hierarkiassa ylempänä olevien yleisen tason funktioiden avulla. Koska yleisesti tarvittavat funktiot on eritelty itsenäisiksi, on järjestelmään helpompi kehittää lisäominaisuuksia, koska usein tarvittavat funktiot on helppo ottaa käyttöön aina kun niitä tarvitaan.

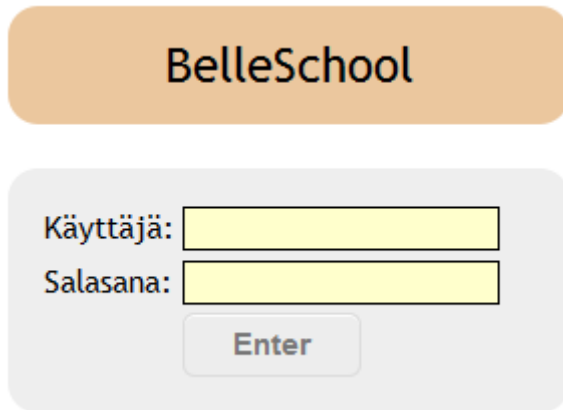


KUVA 6. Ohjelmiston rakenne

8.2 Kirjautuminen

Jotta käyttäjä voi kirjautua järjestelmään sisään, on edellytyksenä että järjestelmänvalvoja on luonut hänelle etukäteen tilin. Kun järjestelmä asennetaan ensimmäistä kertaa, sille luodaan saman tien järjestelmänvalvojan tili. Kun käyttäjä on syöttänyt kirjautumisikkunaan (kuva 7) nimen ja salasanan sekä painanut

Enter-painiketta, järjestelmä ottaa yhteyden tietokantaan ja tarkistaa syötetyt tiedot. Jos tiedot ovat väärät, käyttäjälle näytetään virheviesti. Jos ne ovat oikeat ja niitä vastaavat tiedot löytyvät tietokannasta, käyttäjä päästetään sisään järjestelmään ja hänelle luodaan istunto.



KUVA 7. Kirjautumisruutu

8.3 Käyttäjien hallinta

Käyttäjien hallinta on järjestelmänvalvoja-tilin vastuulla. Käyttäjiä on neljä eri tyyppiä, joista tosin järjestelmänvalvoja voi olla vain yksi. Kyseinen tili luodaan jo ohjelmiston asennusvaiheessa oletussalasanalla, mutta salasanaa voi muuttaa jälkeempään kirjautumalla tilillä sisään ja menemällä tilin asetuksiin. Onkin siis tärkeää, että tämä muutos tehdään ennen kuin järjestelmä otetaan julkiseen käyttöön.

Järjestelmänvalvojan valikossa esitetään sekä oppilas- että opettaja-käyttäjätilityypeille oma valikkonsa (kuva 8). Molemmista valikoista löytyvät LISÄÄ- ja MUUTA-valinnat. Käyttäjien lisääminen tapahtuu molemmille tyypeille samalla tavalla (kuva 9). Järjestelmänvalvojan täytyy syöttää pakollisina tietoina vain käyttäjän nimi ja sukupuoli. Vapaaehtoisina tietoina ovat sähköpostiosoite sekä salasana. Jos salasanaa ei syötetä erikseen, järjestelmä osaa generoida sen automaattisesti tätä varten kirjoitetun funktion avulla. Generoinnissa käytettyä funktiota ei kuvailta tarkemmin tietoturvaan liittyvien seikkojen vuoksi.



KUVA 8. Järjestelmänvalvojan valikko

Lisää opettaja

Etunimi:

Sukunimi:

Sukupuoli:

E-Mail:

Jos jätät alla olevat kentät tyhjiksi, niiden arvot generoidaan satunnaisesti ja kerrotaan tilin luomisen jälkeen.

Käyttäjänimi:

Salasana:

Salasana uudelleen:

KUVA 9. Uuden käyttäjän luontinäkö

Kun tarvittavat tiedot on syötetty lomakkeelle ja järjestelmänvalvoja on painanut Lisää-painiketta, järjestelmä tarkistaa, että tiedot ovat kelvollisia. Käytännössä varmistetaan, ettei samannimistä käyttäjää jo löydy ja mahdollisesti syötetyt salasanat täsmäyvät keskenään. Salasana täytyy syöttää kahdesti kirjoitusvirheiden välttämiseksi. Jos tiedot todetaan kelvollisiksi, ne syötetään tietokantaan ja järjestelmänvalvojalle näytetään sivu, jossa hän voi vielä tarkistaa uuden

käyttäjän nimen ja salasanan. Luotu käyttäjä voi muuttaa salasanaansa ja sähköpostiosoitettaan kirjaututtuaan sisään. Myös järjestelmänvalvoja voi muuttaa kunkin käyttäjän tietoja valitsemalla valikostaan MUUTA-vaihtoehdon ja avaamalla jo olemassa olevan käyttäjän tiedot.

Huoltajien käyttäjätilien hallinta tapahtuu oppilaiden kautta. Kun järjestelmänvalvoja menee muuttamaan oppilaan tietoja, hän voi mennä huoltajien liittämistä varten luodulle sivulle. Tällä sivulla tehdään kaikki huoltajiin liittyvät toimenpiteet (kuva 10). Jos järjestelmässä ei ole luotuna oppilaaseen liitettävää huoltajaa, sen tekeminen onnistuu tällä sivulla. Huoltajan luomista varten tarvittavat tiedot syötetään samaan tapaan kuin muidenkin käyttäjien. Kun tiedot ovat kunnossa, uusi huoltaja lisätään tietokantaan ja hänen sekä oppilaan välinen liitos luodaan samassa prosessissa. Huoltajien hallinnointisivulta näkee minkä nimiset huoltajat on liitetty kuhunkin oppilaaseen. Samalta sivulta voi myös poistaa jo liitettyjä huoltajia. Jos yhdellä huoltajalla on useita oppilaita, voidaan liitos tehdä helposti valitsemalla valikosta jo olemassa oleva huoltaja ja painamalla Lisää-painiketta. Huoltajan käyttäjätiedot tarvitsee siis syöttää vain kerran tilin luomista varten. Tämän jälkeen saman tilin voi liittää useaan oppilaaseen ilman ylimääräistä työtä.

Olemassa olevat huoltajat

Huoltaja:

huo lataja1

Lisää

Poista

Liitetyt huoltajat

huo lataja1

huo taja2

Lisää huoltaja

Etunimi:

Sukunimi:

Sukupuoli:

Nainen

E-mail:

Jos jätät alla olevat kentät tyhjiksi, niiden arvot generoidaan satunnaisesti ja ilmoitetaan tilin luomisen jälkeen.

Käyttäjänimi:

Salasana:

Salasana uudelleen:

Lisää huoltaja

KUVA 10. Huoltajien hallintanäkymä

8.4 Kurssien hallinta

Kurssien luontiin on käytettävä järjestelmänvalvoja-tiliä. Tilillä on mentävä valikosta KURSSIT-osioon ja valittava LISÄÄ. Jotta kurssija voidaan luoda, täytyy tietokannasta löytyä vähintään yksi opettaja-tili. Tämä on välttämätöntä, koska jokaiselle kurssille täytyy määrittää vastaava opettaja. Jos tietokannasta ei löydy yhtään opettajaa antaa ohjelmisto virheviestin, jossa kehoitetaan luomaan kyseisen tyyppinen tili. Kun vaatimus on täytetty voi järjestelmänvalvoja luoda kurssin antamalla sille nimen sekä määrittämällä vastaavan opettajan. Kurssin tiedot tallennetaan tietokantaan sitä varten varattuun taulukkoon.

Kun kurssi on luotu, siihen voidaan lisätä oppilaita. Tämäkin on järjestelmänvalvojan vastuulla. Lisääminen onnistuu valitsemalla valikosta oppilas-kohdasta vaihtoehto MUUTA. Tämän jälkeen järjestelmä esittää listan oppilaista, jotka on

tallennettu tietokantaan. Valvoja valitsee haluamansa oppilaan ja painaa Jatka-painiketta. Toiminto jatkuu hakemalla tietokannasta tiedot siitä, millä kursseilla oppilas jo on, sekä mille kursseille hänet voidaan lisätä. Ohjelmistossa on tarvittavat funktiot, jotta näiden tietojen esittämiseen voidaan luoda selkeät näkymät. Oppilaan kursseista esitetään taulukko, josta näkyvät kurssien nimet sekä mahdollinen arvosana (kuva 11). Käytännössä taulukon luomisfunktio lukee ensin tietokannasta kuinka monta riviä ja saraketta sen täytyy esittää. Näiden perusteella se luo silmukan, joka lisää tarvittavan määrän HTML-koodia, jotta taulukko on selkeä ja siitä löytyy tarvittava määrä tietoa. Kurssit joille oppilas voidaan lisätä esitetään ”rasti ruutuun”-tyyppisenä monivalintalomakkeena, jonka avulla yksi oppilas voidaan helposti lisätä monelle kurssille samalla kertaa. Lomake luodaan samalla periaatteella kuin taulukkokin eli analysoimalla tietokannan tiedot ja lisäämällä vaadittavat HTML-koodit. Kun muutokset tallennetaan, ohjelmissä käy läpi minkä kurssien kohdalla on tehty valinta, ja tekee sen perusteella tarvittavat muutokset tietokantaan kurssit ja oppilaat yhdistävään taulukoon.

Oppilaan kurssit

Käyttäjänimi	Etunimi	Sukunimi	Kurssin ID	Kurssin nimi	Arvosana	Käyttäjän ID
opi1	opi	las1	1	krus1		11
opi1	opi	las1	21	kurs3		11

Valittavat kurssit

- 1 krus1 21
- 11 kurs2 21
- 21 kurs3 51
- 31 kurssi kolme 81
- 41 matikka1 91
- 51 matikka2 91
- 61 fysiikka1 101
- 71 fysiikka2 101

KUVA 11. Oppilaan lisääminen kursseille

Kurssien arvosteluun on käytettävä opettaja-tyyppistä tiliä. Opettaja arvioi kurssin syöttämällä ensin sekä oppilaan että kurssin ID-numeron sekä lopuksi oppilaalle kuuluvan arvosanan (kuva 12). Arvosanan muoto on vapaa, eli sen voi antaa joko numeroin tai sanoin. Opettaja voi myös tarpeen vaatiessa muuttaa

sitä samalla tavalla kuin sitä antaessa. Kun opettaja hyväksyy arvosanan, oppilaat ja kurssit yhdistävään taulukkoon lisätään syötetty arvosana sille varattuun kenttään. Tämän jälkeen sekä oppilas että häneen mahdollisesti liitetty huoltaja voi tarkistaa arvosanan kirjautumalla omalle tililleen (kuva 13).

Oppilaan ID:

Kurssin ID:

Arvosana:

Lisää arvosana

Käyttäjänimi	Etunimi	Sukunimi	Kurssin ID	Kurssin nimi	Arvosana	Käyttäjän ID
opi2	opi	lass2	1	krus1	66666666	31
opi2	opi	lass2	11	kurs2	uuu	31
opi3	opi	las3	11	kurs2	4545	61
opi3	opi	las3	21	kurs3		61
opi1	opi	las1	1	krus1		11
opi1	opi	las1	21	kurs3		11
opi2	opi	lass2	31	kurssi kolme		31

KUVA 12. Kurssien arvostelu

opi las1

Arvosanat:

Käyttäjänimi	Etunimi	Sukunimi	Kurssin ID	Kurssin nimi	Arvosana	Käyttäjän ID
opi1	opi	las1	1	krus1		11
opi1	opi	las1	21	kurs3		11

opi lass2

Arvosanat:

Käyttäjänimi	Etunimi	Sukunimi	Kurssin ID	Kurssin nimi	Arvosana	Käyttäjän ID
opi2	opi	lass2	1	krus1	66666666	31
opi2	opi	lass2	11	kurs2	uuu	31
opi2	opi	lass2	31	kurssi kolme		31

KUVA 13. Huoltaja tarkastelee häneen liitettyjä oppilaitaan

8.5 Viestintäjärjestelmä

VIESTIT-valikosta löytyy kaikille käyttäjille neljä eri toimintoa. Viestin lähetysoiminto esittää käyttäjälle lomakkeen, jossa on kolme eri kenttää (kuva 14). Viestin vastaanottaja valitaan alavetovalikosta, johon on valmiiksi haettu tietokannasta järjestelmästä löytyvät käyttäjät joille sisään kirjautunut käyttäjä voi lähettää viestejä. Luonnollisesti viestille määritetään myös otsikko ja tilavimpaan kenttään kirjoitetaan itse viesti. Kun painetaan Lähetä-painiketta, ohjelmisto tekee muutoksia tietokannasta löytyvään viestejä varten luotuun taulukkoon. Sinne kirjataan itse viestin lisäksi sen lähettäjän sekä vastaanottajan ID:t joiden avulla viesti näkyy vain oikeille henkilöille.

Vastaanottaja:

Otsikko:

Teksti:

KUVA 14. Uuden viestin kirjoitus

Kun käyttäjä haluaa lukea saapuneita viestejä, hän voi valita VIESTIT-valikosta LUE-toiminnon. Siinä esitetään yksinkertaisena listana kaikki kyseiselle käyttäjälle lähetyt viestit (kuva 15). Listasta käy ilmi jokaisen viestin lähettäjän koko nimi, otsikko sekä lähetyksen ajankohta päiväyksen sekä kellonajan osalta mi-

nuuttien tarkkuudella. Viestin otsikko toimii linkkinä, jota painamalla käyttäjä pääsee käsiksi itse viestiin.

Lähettäjä	Otsikko	Päiväys
ad min	uusi viesti	06:49 - Friday, March 1, 2013
ad min	kurssin aloitus	06:49 - Friday, March 1, 2013
ad min	tietoja oppilaasta	06:49 - Friday, March 1, 2013
opi las3	sdfd	04:34 - Wednesday, February 20, 2013

KUVA 15. Näkymä saapuneista viesteistä

Viestin tarkastelunäkymästä löytyy kaikki viestin tiedot, eli otsikko, lähettäjä sekä lähetyksen ajankohta (kuva 16). Lopuksi esitetään tietenkin itse viesti. Viestin luettuaan käyttäjällä on kaksi eri valintaa. Hän voi joko painaa Vastaa-painiketta, jolloin hänet viedään suoraan viestin kirjoitusnäkömään josta on valmiiksi valittu juuri luetun viestin lähettäjä vastaanottajaksi. Toinen vaihtoehto on, että käyttäjä siirtää viestin roskakoriin painamalla Poista-painiketta.

tärkeä tiedotus

Lähetetty 06:52 - Friday, March 1, 2013 opettaja1

koulu suljetaan tänään kello 14.00

Vastaa : Poista

KUVA 16. Käyttäjä lukee viestiä

Käyttäjä voi myös tarkastella lähetettyjä viestejään samaan tapaan kuin vastaanotettuja valitsemalla VIESTIT-valikosta LÄHETETYT-valinnan. Kyseisen toiminnon on tarkoituksena toimia varmistuksena, josta käyttäjä voi tarkistaa että hänen lähettämänsä viesti on varmasti lähtenyt eteenpäin oikealle henkilölle. Lähetettyjen viestien näkymä ei eroa vastaanotettujen viestien näkymästä.

Myös ROSKAT-kansiossa voi tarkastella viestejä samaan tapaan kuin muissakin kansioissa. Erona on vain se, että kun käyttäjä valitsee viestistä Poista-

valinnan, ohjelmisto käyttää eri funktiota kuin vastaanotettujen viestien Poista-valinta, jonka kautta viesti vain siirretään näkyväksi ROSKAT-kansioon. Kun Poista-valintaa käytetään roskien yhteydessä, ohjelmisto antaa tietokannalle käskyn tuhota sieltä kokonaan viestin sisältävä rivi. Tällöin viesti häviää lopullisesti koko ympäristöstä eikä sitä voi enää tarkastella.

9 POHDINTA

Työn tarkoituksena oli tehdä tilaajalle toimintavalmis oppimisympäristö, jossa oppilaat, opettajat sekä oppilaiden huoltajat voivat toimia keskenään ja seurata oppilaiden edistymistä koulussa. Työnanto tuli tilaajalta, joka on saanut kontaktiensa kautta toiveita nykyisiä ratkaisuja paremmasta ympäristöstä. Vaatimuksena työlle oli haluttujen ominaisuuksien lisäksi se, että järjestelmää on pystytävä suorittamaan Azure-pilvipalvelussa.

Nykyisin käytössä olevia oppimisympäristöjä vertailemalla kävi ilmi, ettei täysin nyt kehitettyä ympäristöä vastaavaa ratkaisua ole Suomen markkinoilla. Wilma vaatii toimiakseen muitakin sovelluksia, mikä aiheuttaa ylimääräisiä kustannuksia. Helmi taas ei ole täysin koulun itsensä hallinnassa, sillä sen tarvitsemat palvelimet ovat täysin ohjelmistoa tarjoavan yrityksen vastuulla. Kivahko taas ei sisällä oppilashallintoon tarvittavia toimintoja.

Tavoitteiden toteuttamisessa onnistuttiin hyvin. Kaikki tilaajan kanssa sovitut ominaisuudet saatiin toteutettua ilman, että työn aikataulua olisi tarvinnut venyttää aiempaa sovittua pidemmäksi. Myös tilaaja oli työn tulokseen tyytyväinen, eikä testausvaiheen aikana tullut negatiivista palautetta.

Ohjelmointivaiheessa olisi kannattanut etukäteen varautua mahdollisiin ongelmiin ohjelmiston Azure-pilvipalveluun siirron yhteydessä. Koska ohjelmistoa kehitettiin paikallisesti omalta tietokoneelta, oli ohjelmiston kehitys- ja suoritusympäristöissä joitain eroja. Tiedyt PHP-funktiot eivät toimineet ollenkaan Azure-pilvipalvelussa, jonka takia ohjelmakoodia piti muuttaa erikseen Azurea varten. Hyvin suunniteltu ohjelmiston rakenne kuitenkin helpotti ongelman ratkaisua, sillä käytännössä koko ohjelmakoodista tarvitsi muuttaa vain muutamaa riviä. Jatkossa on viisasta kehittää ohjelmistoa joko pelkästään Azure-pilvipalvelua käyttäen tai vaihtoehtoisesti testata välittömästi muutoksien toimivuutta myös siinä.

Koska työssä keskityttiin ohjelmiston toiminnollisuuden toteuttamiseen, on lopputuloksen käyttöliittymä melko pelkistetty. Sitä onkin syytä kehittää jatkotoimenpiteenä opettajilta saadun palautteen perusteella. Toinen tärkeä ominai-

suus on kääntää ohjelmisto useammalle kielelle. Nyt toteutettu versio tukee pelkäästään suomen kieltä, mutta etenkin suuremmissa kouluissa voi tulla tarve käyttää ohjelmistoa useammalla kielellä.

LÄHTEET

1. Tahkokorpi, Mirjam 2012. Sähköinen yhteydenpito korvaa reissuvihkot. Saatavissa: http://yle.fi/uutiset/sahkoinen_yhteydenpito_korvaa_reissuvihkot/6329301. Hakupäivä 26.2.2013.
2. Suomalaista ohjelmisto-osaamista. Saatavissa: <http://www.starsoft.fi/public/?q=node/1099>. Hakupäivä 7.3.2013.
3. Lahti, Jarmo 2008. Kouluhallinnon ohjelmistojen webitys vauhdittuu. Saatavissa: <http://www.digitoday.fi/data/2008/08/19/kouluhallinnon-ohjelmistojen-webitys-vauhdittuu/200821472/66>. Hakupäivä 7.3.2013.
4. Nextime solutions oy. Saatavissa: <http://www.nextime.fi/>. Hakupäivä 7.3.2013.
5. Helmi. Saatavissa: <http://www.helmi.fi/>. Hakupäivä 7.3.2013.
6. Karvonen, Tuomas 2004. Soneralta sähköinen vihko koululaisille. Saatavissa: <http://www.digitoday.fi/data/2004/10/14/soneralta-sahkoinen-vihko-koululaisille/200414807/66>. Hakupäivä 7.3.2013.
7. Miksi Helmi?. Saatavissa: <http://www.helmireissuvihko.fi/index.php?page=5>. Hakupäivä 7.3.2013
8. Helmi Viestintä. Saatavissa: <http://www.helmi.fi/viestinta.html>. Hakupäivä 7.3.2013.
9. MUKAVA esittely. Saatavissa: <http://www.mukavahanke.com/index.html>. Hakupäivä 7.3.2013.
10. Kivahko - sähköinen viestintäsovellus kodin ja koulun yhteistyöhön. Saatavissa: <http://www.mukavahanke.com/projektit/kivahko.html>. Hakupäivä 25.3.2013.

11. Kivahko -digitaalinen viestintäjärjestelmä. Saatavissa:
<http://www.mukavahanke.com/kivahko/toiminnot.html>. Hakupäivä 7.3.2013.
12. Laine, Harri – Paakki, Jukka. Ohjelmistotuotanto. Saatavissa:
<http://www.cs.helsinki.fi/u/paakki/ohtuk03-luento2.pdf>. Hakupäivä 1.3.2013.
13. Software Requirements Specification. Saatavissa: http://www.sqa.org.uk/e-learning/AppsDev04CD/page_08.htm. Hakupäivä 5.3.2013.
14. Ohjelmistojen mallintaminen. Saatavissa:
<http://www.cs.helsinki.fi/u/mluukkai/ohmas10/luentokalvot/luento3.pdf>. Hakupäivä 5.3.2013.
15. Kehittämistyön vaiheet ja elinkaarimallit. Saatavissa:
http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittami-nen/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elikaarimallit.htm. Hakupäivä 5.3.2013.
16. Katara, Mika. Testaus osana ohjelmistoprosessia. Saatavissa:
http://www.cs.tut.fi/~testaus/s2011/luennot/OHJ-3060_2011_50-109.pdf. Hakupäivä 5.3.2013.
17. Software maintenance - an overview. Saatavissa:
<http://www.bcs.org/content/conWebDoc/3063>. Hakupäivä 7.3.2013.
18. PHP:n perusteet. Saatavissa: http://users.jyu.fi/~kolli/ITK215_05/php/. Hakupäivä 2.1.2013.
19. Apache 2.x on Microsoft Windows. Saatavissa:
<http://www.php.net/manual/en/install.windows.apache2.php>. Hakupäivä 4.4.2013.
20. MySQL Market Share. Saatavissa: <http://www.mysql.com/why-mysql/marketshare/>. Hakupäivä 4.1.2013.

21. MySQL Downloads. Saatavissa: <http://www.mysql.com/downloads/>. Hakupäivä 4.1.2013.
22. The history of MySQL. Saatavissa: <http://buytaert.net/the-history-of-mysql-ab>. Hakupäivä 4.1.2013.
23. What is Windows Azure. Saatavissa: <http://www.windowsazure.com/en-us/home/features/what-is-windows-azure/>. Hakupäivä 4.1.2013.
24. Windows Azure Platform Launch Update. Saatavissa: <http://blogs.msdn.com/b/windowsazure/archive/2009/10/29/windows-azure-platform-launch-update.aspx>. Hakupäivä 4.4.2013.
25. Current Members – W3C. Saatavissa: <http://www.w3.org/Consortium/Member/List>. Hakupäivä 15.1.2013.
26. Web Browsers. Saatavissa: <http://www.ntchosting.com/internet/web-browser.html>. Hakupäivä 4.4.2013.
27. HTML & CSS. Saatavissa: <http://www.w3.org/standards/webdesign/htmlcss>. Hakupäivä 4.4.2013.
28. CSS Tutorial. Saatavissa: <http://www.w3schools.com/css/>. Hakupäivä 4.4.2013.
29. SQL AUTO INCREMENT Field. Saatavissa: http://www.w3schools.com/sql/sql_autoincrement.asp. Hakupäivä 28.2.2013.
30. Unix time conversion. Saatavissa: http://www.onlineconversion.com/unix_time.htm. Hakupäivä 11.1.2013.
31. The MD5 Message-Digest Algorithm. Saatavissa: <http://www.ietf.org/rfc/rfc1321.txt>. Hakupäivä 28.2.2013.
32. SQL Injection. Saatavissa: <http://php.net/manual/en/security.database.sql-injection.php>. Hakupäivä 11.3.2013.

33. The End of Time. Saatavissa:

http://stablecross.com/files/End_Of_Time.html. Hakupäivä 28.2.2013.