

Opinnäytetyö (AMK)

Tietojenkäsittely

Tietojärjestelmät

2010

Sami Timonen

SELAINPOHJAISEN SOVELLUKSEN TOTEUTTAMINEN .NET- YMPÄRISTÖLLÄ



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Tietojärjestelmät

12.10.2010 | 38

Anne Jumppanen

Sami Timonen

Selainpohjaisen sovelluksen toteuttaminen .NET-ympäristössä

Microsoftin kehittämä ohjelmointimalli .NET mahdollistaa usean ohjelmointikielen väliset yhteydet. Se on useista muista Microsoftin tuotteista poiketen alustasta riippumaton julkaistujen ECMA-334- ja ECMA-335-standardien ansiosta. Ne mahdollistavat kolmansille osapuolille jakelujen luomisen. ASP.NET on osa .NET-arkkitehtuuria. Se on täysin oliopohjainen ja tarkoitettu erityisesti selainpohjaisten sovellusten luontiin.

Henkilön todentamistapoja on useita. Suosituin näistä tavoista on todentaminen salasanan perusteella, koska se on helppo ja halpa tapa toteuttaa. Salasanatodennuksen kahtena suurimpana ongelmana on salasanan unohtaminen ja salasanan joutuminen väärän henkilön tietoon. Salasanan unohtuessa joudutaan käyttämään hätätodennusta, jossa käyttäjän todentamiseen käytetään usein järjestelmän ylläpitoa, sähköpostia, sekä kysymyksiä, joihin käyttäjä ainoastaan tietää vastauksen. Salasanan väriin käsiin joutumista on pyritty estämään erilaisilla sovelluksen tietoturva parantavilla ratkaisuilla, kuten varmenesuojauksilla ja salasanan tietokantaan salatussa muodossa tallentamisella.

Opinnäytteenä luotiin roolipohjainen internetin yli toimiva sovellus. Sovelluksen ydinominaisuuksia ovat kalenteripohjainen läsnäoloilmoittautuminen, käyttäjienhallinta, sekä tarvittavien raporttien tulostaminen. Osa sovelluksen ominaisuuksista, kuten kirjautuminen voidaan halutessa suojata varmenteella. Tämän lisäksi tietoturva on sovelluksessa otettu huomioon evästeiden ominaisuuksissa, sekä salasanan hash-muotoon tallentamisella. Sovellus on luotu Visual Studio 2008 ohjelmankehitysympäristöllä, ASP.NET-tekniikan avulla. Ohjelmointikielenä on käytetty pääasiassa C#-kieltä.

Opinnäytteen teoreettisessa osuudessa käytettiin hyödyksi alan kirjallisuutta. Sovelluksen määrittelyn ja suunnittelun pohjana käytettiin Hanna Rämön vuonna 2008 luomaa opinnäytettä ASP.NET ja selainpohjaisen sovelluksen suunnittelu. Sovellusta luotaessa käytettiin hyväksi alan kirjallisuutta ja useita keskustelufoorumia.

ASIASANAT:

ASP .NET, verkko-ohjelmointi, verifiointi, varmenteet, C#, SSL, SSH

BACHELOR'S THESIS | ABSTRACT

UNIVERSITY OF APPLIED SCIENCES

Information Technology | Information Systems

12.10.2010 | 38

Anne Jumppanen

Sami Timonen

Browser-Based Application Implementation in NET Environment

Microsoft's programming model .NET enables relationships between several programming languages. Unlike most of Microsoft's products, .NET is platform-independent. That has made possible the published ECMA-334 and ECMA-335 specifications. The specifications enable a third party to create distributions. .NET is fully object-oriented and specially designed for browser-based applications.

There are several methods of person identification. The most popular of them is the password based authentication, because it is an easy and inexpensive way to carry out. The Password based authentication has two major problems: forgetting the password and the password being used by the wrong person. When a person forgets the password, emergency authentication is required. Help desk, e-mail or life questions are often used for emergency authentication. Life questions are personal questions, which only the authenticated person is expected to be able to answer. Different security solutions, e.g. certificates and storing password at encrypted format in to database, are used for preventing password being used by the wrong person.

The main goal for this thesis was to create over the Internet an operating role-based application. The application's main features are calendar-based attendance information, user management and the necessary report printing. Some of the application features can be protected with certificates. In addition, application security has been taken into account with the characteristics of cookies, as well as storing password trough hash-function. Visual Studio 2008 was used as the application's program development environment and it was created with ASP.NET-technique. The main programming language of the application was C#.

The theoretical part of the thesis is based on field literature. The application definitions and planning was based on the theses on ASP.NET and Designing Web-Based Application created by Hanna Rämö in 2008. Field literature and a number of discussion forums were used in creating the application.

KEYWORDS:

ASP .NET, web programming, verification, certificates, C#, SSL, SSH

SISÄLTÖ

1 JOHDANTO	6
2 KÄYTETYT TEKNIIKAT JA NIIDEN HYÖDYT	8
2.1 .NET-ympäristö	8
2.2 ASP.NET	9
2.2 ADO.NET	11
2.3 C#	11
2.4 Visual Studio 2008	12
3 TODENNUS	13
3.1 Todentaminen henkilön omistaman esineen perusteella	13
3.2 Todentaminen salasanan perusteella	13
3.3 Todentaminen fyysisen ominaisuuden perusteella	14
4 VARMENTAMINEN	15
4.1 Varmenteet	15
4.2 Julkisen avaimen salaus	15
4.3 SSL-salaus	19
4.4 SSH-salaus	20
4.4 Salaaminen hajautusfunktiolla	22
5 OPINNÄYTETYÖSKENTELYN SEURANTAA AVUSTAVAN JÄRJESTELMÄN TOTEUTUS	23
5.1 Järjestelmän vaatimukset	23
5.2 Toiminnot ja ominaisuudet	24
5.2.1 Kirjautuminen ja kirjautumistunnukset	24
5.2.2 Ilmoitukset	26
5.2.3 Kalenteri ja ilmoittautuminen	27
5.2.4 Etsintä ja raporttien tulostaminen	29
5.2.5 Käyttäjienhallinta	30
5.3 Ohjeistus käyttöönottoon	32
6 POHDINTA	36
LÄHTEET	38
LIITTEET	
1 Määrittelydokumentti	
2 Suunnitteludokumentti	

3 Testausdokumentti

KUVAT

Kuva 1 ASP.NET-sivuston rakenne.....	10
Kuva 2 Tunnusten vaihto –sivu.....	26
Kuva 3 Ilmoitukset –sivu	27
Kuva 4 Kalenteri -sivu	28
Kuva 5 Raporttien tulostaminen –sivu	30
Kuva 6 Käyttäjienhallinta –sivu	32

KUVIOT

Kuvio 1. Salaus julkisen avaimen menetelmällä (Stallings 2005, 262.).....	16
Kuvio 2. Julkisen avaimen jakajana kolmas osapuoli (Stallings 2005, 293).	17
Kuvio 3. Sertifikaatin noutaminen (Stallings 2005, 294).....	19
Kuvio 4. SSH-arkkitehtuuri (Barrett & Silverman 2001, 20).	21

1 Johdanto

Microsoftin .NET-ympäristö mahdollistaa erittäin monipuolisten työpöytä- ja Internet-sovellusten luonnin. Sen yksi suurimmista hyödyistä on usean ohjelmointikielen yhteen toimivuus. Internet-sovellusten luonnin mahdollistaa .NET-ympäristön oliopohjainen ASP.NET-tekniikka.

Sovelluksen käyttäjän todentamiseksi on useita tapoja. Useimmat yleisimmin käytetyistä ratkaisuihin ovat vähän resursseja vieviä, mutta tietoturvallisuuden kannalta erittäin riskialttiita. Tietoturvallisuutta lisäävät varmenteet, joiden avulla varmistetaan, että yhteys kohteeseen tulee varmasti oikeasta paikasta.

Opinnäytteenä toteutettiin opinnäytetyöskentelyn seuraamista avustava ja Internetin yli toimiva sovellus. Sovelluksen tarkoituksena on helpottaa opinnäytetyön ohjaajan ja tekijän välistä yhteistyötä, ylläpitämällä opinnäyteprosessissa mukana olevia tietoja ja mahdollistamalla erilaisten raporttien tulostus näistä tiedoista. Sovelluksen keskeisiä ominaisuuksia ovat käyttäjien ilmoittautumiset opinnäytteitä koskeviin tilaisuuksiin, tilaisuuksien läsnäolojen merkitseminen, opettajien mahdollisuus seurata opiskelijan opinnäytteen etenemistä ja näiden tietojen tulostaminen. Sovelluksen ylläpidon vaatimien resurssien vähentämiseksi toteutettiin sovellukseen opettajille ylläpitäjän kaltaiset roolit. Työ tehtiin toimeksiantona Turun ammattikorkeakoulun tietojenkäsittelyn koulutusohjelmalle. Työn määrittelyn ja suunnittelun pohjana käytettiin Hanna Rämön opinnäytetyötä ASP.NET ja selainpohjaisen sovelluksen suunnittelu. Siinä on tutkittu järjestelmän vaatimuksia ja järjestelmän toteutuksessa käytettäviä tekniikoita. Lisäksi se sisältää alustavat määrittely- ja suunnitteludokumentit.

Opinnäytteen teoreettinen viitekehys koostuu sovelluksen luomisessa käytettyjen tekniikoiden, käyttäjien todentamisen sekä varmenneratkaisujen tutkimisesta. Käytettyjä tekniikoita ja niiden hyötyjä on tutkittu sovelluksen toteuttajan näkökulmasta. Käyttäjien todentamisen eri mahdollisuuksissa on otettu huomioon eri todentamistapojen tietoturvallisuus sekä niiden vaatimat

resurssit. Varmenneosiossa on keskitytty pääasiassa julkisen avaimen salausmenetelmiin.

Empiirisessä osiossa on kuvailtu toteutetun järjestelmän toiminnot ja ominaisuudet. Koska opinnäyte ei sisällä käyttöönottovaihetta, on opinnäytteeseen sisällytetty ohjeistus sovelluksen käyttöönottoa varten. Ohjeistuksessa on selvitetty ne kohdat toteutetusta järjestelmästä, jotka poikkeavat useimpien ASP.NET-sovellusten käyttöönotosta.

Itse sovelluksen luomisen ohjelmankehitysympäristönä oli Microsoft Visual Studio 2008. Sovellus luotiin ASP.NET-tekniikalla ja ohjelmointikielenä käytettiin C#:ia. Sovelluksen tarvitsema tietokanta luotiin Microsoft SQL Server 2005:llä. Sen lähtökohtana oli luoda kriteerit täyttävä sovellus, jonka käyttö on helppo omaksua. Ohjelmankehittämisen yhteydessä nousi esiin useita resurssien, käytettävyyden ja tietoturvallisuuden kannalta tärkeitä huomioita ja vaatimuksia, jotka vaativat poikkeamista alustavista suunnitelmista. Tämän vuoksi lopullinen sovellus poikkeaa osittain alustavista määrityksistä ja suunnitelmista, jotka olivat Hanna Rämön opinnäytetyössä ASP.NET ja selainpohjaisen sovelluksen suunnittelu.

2 Käytetyt tekniikat ja niiden hyödyt

2.1 .NET-ympäristö

Microsoftin kehittämän .NET-ohjelmointimallin yhtenä suurimpana hyötynä ohjelmoijan näkökulmasta on usean ohjelmointikielen yhteen toimivuus. (Troelsen 2007, 6.) Se on nimensä mukaisesti suunniteltu erityisesti hajautettuun ympäristöön, jossa ollaan yhteydessä toisiinsa Internetin välityksellä. Eri ohjelmointikielten toimivuuden mahdollistaa .NET-ympäristön virtuaalikone. Se huolehtii ohjelman ajamisesta. (Moghadampour 2009, 12.) Microsoft Framework 3.5 SDK:n tukemat kielet ovat C#, Visual Basic.NET, J#, C++/CLI ja Jscript .NET. Lisäksi usea muu ohjelmointikieli toimii niille kehitettyjen .NET -kääntäjien avulla. (Troelsen 2007, 9.) Yhteen toimivuutta käytetään ohjelman suorituksen lisäksi perinnässä, poikkeusten käsittelyssä sekä testausvaiheen virheiden etsinnässä ja korjauksessa (Troelsen 2007, 6).

Sovellusten luominen ja suorittaminen .NET-ympäristöllä on mahdollista Windows-käyttöjärjestelmien lisäksi myös muissa käyttöjärjestelmissä. Esimerkkinä Mac OS X ja useat Unix/Linux-jakelut. (Troelsen 2007, 6.) Tämän mahdollistavat Microsoftin julkaisemat ECMA-334 ja ECMA-335 dokumentit, joissa kuvaillaan `c#`:n ja common language infrastructure (CLI) syntaksit ja semantiikka, .NET kokoonpanon muodot, .NET nimiavaruudet, sekä virtual execution system (VES) mekaniikka. Dokumenttien avulla kolmannet osapuolet voivat luoda .NET -alustan jakeluita mille tahansa käyttöjärjestelmälle ja prosessorille. (Troelsen 2007, 32.)

Kolme tekijää mahdollistavat .NET-ympäristön toimimisen: common language runtime (CLR), common type system (CTS) ja common language specification (CLS). CLR etsii, lataa ja hallitsee .NET-syötteitä, hoitaa muistinhallinnan ja suorittaa useita turvatarkistuksia. Lisäksi se luo sovellukselle rajat ja rakenteen. CTS kuvailee kaikki tietotyypit ja ohjelma rakenteet joita suoritus tukee. Lisäksi se määrittelee niiden keskinäisen toimivuuden ja näkyvyyden .NET-

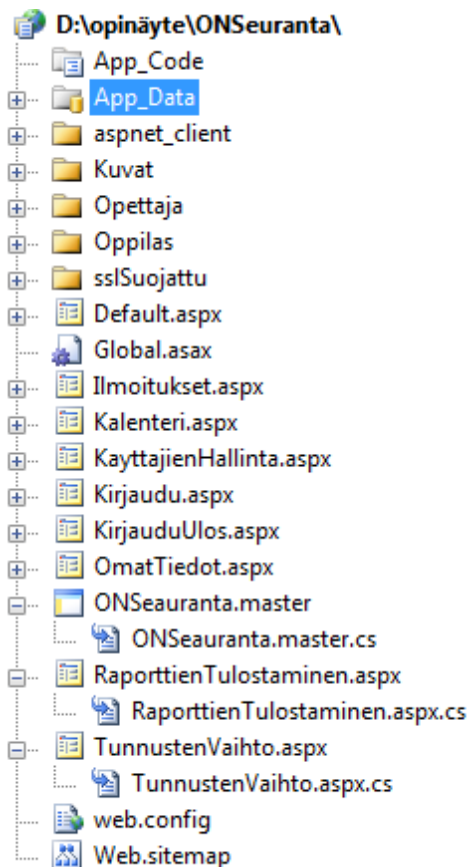
metadatatassa. CLS on CTS:n osa, joka määrittelee yleiset tyypit ja ohjelmointirakenteet, joita tuetut ohjelmointikielet käyttävät. (Troelsen 2007, 6, 7.) Toisin sanoen CLS sisältää säännöt, joita kääntäjän on tuettava tuottaakseen CLR:ssä toimivaa koodia (Troelsen 2007, 21).

2.2 ASP.NET

ASP.NET-tekniikan avulla .NET-alustaa voidaan käyttää luomaan työpöytäsovellusten lisäksi selainpohjaisia sovelluksia (Troelsen 2007, 1167). ASP.NET-tekniikalla voidaan luoda täysin oliopohjaisia sivustoja (Troelsen 2007, 1177).

ASP.NET on kehitetty ASP-tekniikan pohjalta. Sen suurimpia hyötyjä ohjelmoijan näkökulmasta perinteiseen ASP-tekniikkaan nähden on sivujen luominen .NET-ympäristön tukemilla ohjelmointikielillä. Lisäksi code-behind mallin avulla ASP.NET-sivujen logiikka voidaan eritellä erilliseen tiedostoon. (Troelsen 2007, 1177.)

ASP.NET webkontrollien avulla käyttöliittymä voidaan rakentaa kuten luotaessa työpöytäsovelluksia. Lisäksi ominaisuudet kuten master page ja tuki teemoille helpottavat sovellusten ulkoasujen luomista ja muokkaamista. (Troelsen 2007, 1177-1178.) Yleisimmin ASP.NET-sivusto rakentuu .aspx, code-behind, Global.asax, web.config ja .master sivuista.



Kuva 1 ASP.NET-sivuston rakenne.

Kuvassa 1 on esitettyä ASP.NET-sivuston sisältämiä tiedostoja. Aspx-päätteiset tiedostot sisältävät yleensä staattiset HTML-merkinnät, palvelinpuolen webkomponenttien määrykset ja kaikki tarvittavat staattiset ja dynaamiset websivujen sisällön (Troelsen 2007, 1179). Cs-päätteiset tiedostot ovat C# code-behind-tiedostoja. Näiden tiedostojen päätte riippuu käytettävästä ohjelmointikielestä. Niiden sisältö voidaan myös sisällyttää .dll-päätteisiin tiedostoihin, jolloin code-behind-tiedostoja ei tarvita. Master-päätteinen tiedosto on master page-tiedosto, jossa määritellään sivuston yhteinen runko, jota useimmat sivuston sivut käyttävät (Troelsen 2007, 1207). Global.asax-tiedostoon kuvataan koko sivuston käyttäytymiseen vaikuttavat tapahtumankäsittelijät (Troelsen 2007, 1252). Web.config-tiedostoon määritetään asetukset, jotka määrittävät kuinka sivusto toimii ajonaikana

(Troelsen 2007, 1203). Sinne voidaan esimerkiksi määrittää tiettyjen käyttäjien, tai käyttäjäroolien pääsy sivustolle.

ASP.NET pohjaisia sivuja voidaan luoda ja ylläpitää Windows-käyttöjärjestelmien lisäksi myös muissa käyttöjärjestelmissä ASP.NET-liitäntäisten avulla (Troelsen 2007, 1167).

2.2 ADO.NET

ADO.NET on joukko nimiavaruuksia, joiden avulla .NET-alusta keskustelee relaatiotietokantojen kanssa. Data provider –luokkien avulla .NET keskustele sille määritetyn tietokannan kanssa. (Troelsen 2007, 731.) Microsoftin .NET-jakelun mukana tulevat data provider -luokat tukevat OLE DB-, Microsoft SQL Server-, Microsoft SQL Server Mobile-, ODBC- ja Oracle-tietokantoja (Troelsen 2007, 735). Lisäksi useilta kolmansilta osapuolilta on saatavissa data provider – luokkia kaupallisiin ja avoimen lähdekoodin tietokantoihin (Troelsen 2007, 736).

ADO.NET voi olla yhteydettömässä tilassa tietokantaan (Troelsen 2007, 733). Yhteydettömän tilan mahdollistaa DataSet -objekti. DataSet luo tietokannan tauluista paikallisen kopion. Luotu kopio on oletuksena XML-muodossa, jonka vuoksi ADO.NET sisältää vahvan tuen XML-muotoisen datan käsittelylle. (Troelsen 2007, 731.) Data adapterin avulla DataSet muodostaa ja sulkee automaattisesti yhteyden tietokantaan. Tämä vapauttaa yhteyksiä muita käyttäjiä varten ja vähentää verkkoliikenteen käyttöä. (Troelsen 2007, 733)

2.3 C#

C# on Microsoftin kehittämä oliopohjainen ohjelmointikieli. Kieli on suunniteltu erityisesti .NET-alustaa varten ja se vaatii toimiakseen .NET-ajoympäristön. Se on luotu usean ohjelmointikielen pohjalta, joten useat sen syntaksit muistuttavat Java-, C++- ja Visual Basic 6.0-syntakseja. Siinä on pyritty yhdistämään Javan

selkeät syntaksit, Visual Basic 6.0:n yksinkertaisuus sekä C++:n tehokkuus ja joustavuus. (Troelsen 2007, 7-9.)

C# on kirjainherkkä kieli, joten koodissa isoilla ja pienillä kirjaimilla on merkitystä. (Moghadampour 2009, 70). Toisin kuin useimmissa ohjelmointikielissä, siinä ei ole mahdollista luoda globaaleja funktioita tai vakioita. Lisäksi C# vaatii, että kaikki ohjelmalogiikka sisältää tyyppimäärittelyn. (Troelsen 2007, 69.)

2.4 Visual Studio 2008

Visual Studio 2008 on Microsoftin erittäin monipuolinen ohjelmankehitysympäristö. Se sisältää työkaluja graafisen käyttöliittymän suunnitteluun, tietokantojenhallintaan, objektien ja projektien selaamiseen, sekä integroidun ohje- ja avustusjärjestelmän. (Troelsen 2007, 52)

Visual Studio 2008 sisältää vähintään kirjan verran ominaisuuksia (Troelsen 2007, 53). Kuvailen vain oppinäytetyöni kannalta tärkeimmät ominaisuudet, jotka puuttuvat useista muista ohjelmankehitysympäristöstä.

Visual Studio 2008 tukee koodin refaktorointia automatisoimalla prosessia. Refaktoroinnissa parannetaan koodin perustaa muuttamatta sen ulkoista käyttäytymistä. (Troelsen 2007, 56.) Yksi esimerkki automaattisesta refaktoroinnista on objektin uudelleen nimeäminen. Muuttamalla objektin nimeä se osaa nimetä objektiin viittaavat kohdat koodissa.

Valikoiden ja pikanäppäimien avulla voidaan lisätä valmiita runkoja C#-koodiin (Troelsen 2007, 58).

Class Designerin avulla voidaan suunnitella, lisätä, poistaa ja muokata luokkia, sekä niiden relaatioita visuaalisesti. Luokan muokkaaminen Class Designerissa, muuttaa luokkaa C# tiedostossa ja päinvastoin. (Troelsen 2007, 59)

Server Explorerin avulla voidaan muodostaa yhteyksiä tietokantoihin, luoda uusia tietokantoja, luoda tauluja tietokantoihin ja muokata taulujen

ominaisuuksia ja sisältöä sekä luoda tietokantakaavioita. Tietokantakaavioiden avulla voidaan visuaalisesti luoda ja muokata taulujen välisiä relaatioita. (Troelsen 2007, 744 - 749.)

Object Test Benchillä voidaan nopeasti luoda ja kutsua luokan esiintymää ilman koko sovelluksen suoritusta (Troelsen 2007, 62, 63).

3 Todennus

Tunnistautumisen tarkoituksena on selvittää kuka palvelun käyttäjistä on kyseessä (Järvinen 2003, 204). Todennuksella taas tarkoitetaan ihmisten henkilöllisyyden varmistamista tai tietokoneiden ja ohjelmistojen osalta niiden aitouden varmistamista (Järvinen 2003, 33.)

Ihmisen henkilöllisyyden varmistaminen perustuu joko henkilön omistamaan asiaan, hänen tietämäänsä asiaan tai hänen fyysisiin ominaisuuksiinsa (Järvinen 2003, 35, 36).

3.1 Todentaminen henkilön omistaman esineen perusteella

Henkilön omistamalla esineellä tarkoitetaan yleensä avainta tai henkilötodistusta. Tämän todentamistavan riskinä on todennusvälineen joutuminen jonkun muun kuin omistajan käsiin. Tämän riskin välttämiseksi esineen täytyisi olla vaikeasti kopioitavissa ja jollain tavalla sidoksissa käyttäjään. (Järvinen 2003, 35)

3.2 Todentaminen salasanan perusteella

Salasana on helpoin ja halvin tapa toteuttaa todennus. Tämän vuoksi salasanaa käytetään tietotekniikassa yleisimmin todentamaan ihmisen henkilöllisyys. Myös sen riskit ovat suuret. Salasanan pitäisi olla vain järjestelmän ja käyttäjän tiedossa, mutta se saattaa paljastua tai se voidaan kaapata käyttäjän

tietämättä. Koska salasana ei ole fyysinen esine, sen kaappaamista ei välttämättä huomaa, toisin kuin esimerkiksi kulkukortin katoamista. (Järvinen 2003, 35.) Käyttäjät saattavat usein myös unohtaa salasanansa. Tämän vuoksi käytetään normaalin todennuksen rinnalla niin kutsuttua hätätodennusta, joka korvaa aikaisemman salasanan. Siinä käytetään todentamiseen joko sähköpostia, kysymyksiä tai järjestelmän ylläpitoa. (Brainard ym. 2006, 1.)

Käytetyin salasanan palautustapa on sähköpostin kautta todentaminen. Siinä käyttäjän unohtaessa salasanansa, lähetetään uusi salasana sähköpostiin. Toisessa tavassa on salasanan unohtuessa käyttäjältä kysyttävät kysymykset, joihin vain käyttäjän oletetaan tietävän vastaukset. Tämän jälkeen käyttäjälle annetaan uusi salasana palveluun. (Brainard ym. 2006, 1.) Tämä luo lisää riskejä. Kysymyksiin saatetaan arvata vastaukset, vastaukset voivat olla yleisessä tiedossa esimerkiksi julkisissa rekistereissä, tai käyttäjä on voinut paljastaa vastauksen esimerkiksi jossain kyselyssä. Tällöin vastaus saattaa olla ei-luotettavan henkilön tiedossa. (Brainard ym. 2006, 1, 3.) Järjestelmän ylläpidon kautta todentamisessa ylläpito antaa käyttäjälle uuden salasanan. Tässä tavassa voidaan käyttää hyväksi useita eri todennuksia. Ylläpito voi esimerkiksi tunnistaa käyttäjän henkilötodistuksen perusteella. Vaikka tämä onkin suosittu todennustapa, tämä vie ylläpidolta resursseja ja saattaa siksi tulla kalliiksi. (Brainard ym. 2006, 1, 2.)

3.3 Todentaminen fyysisen ominaisuuden perusteella

Henkilön fyysisiin ominaisuuksiin perustuva todennus on hankalin ja kallein toteuttaa, koska se vaatii erikoislaitteita. Tällaisia todennuksia ovat esimerkiksi kasvonpiirteisiin, ääneen, sormenjälkiin tai verkkokalvoon perustuvat todennukset. Erikoislaitteet ja se, ettei esimerkiksi palvelimella ole yksilöllisiä fyysisiä ominaisuuksia, joilla aitous voitaisiin varmistaa, rajoittavat käyttötilanteiden määrää. (Järvinen 2003, 36.)

4 Varmentaminen

4.1 Varmenteet

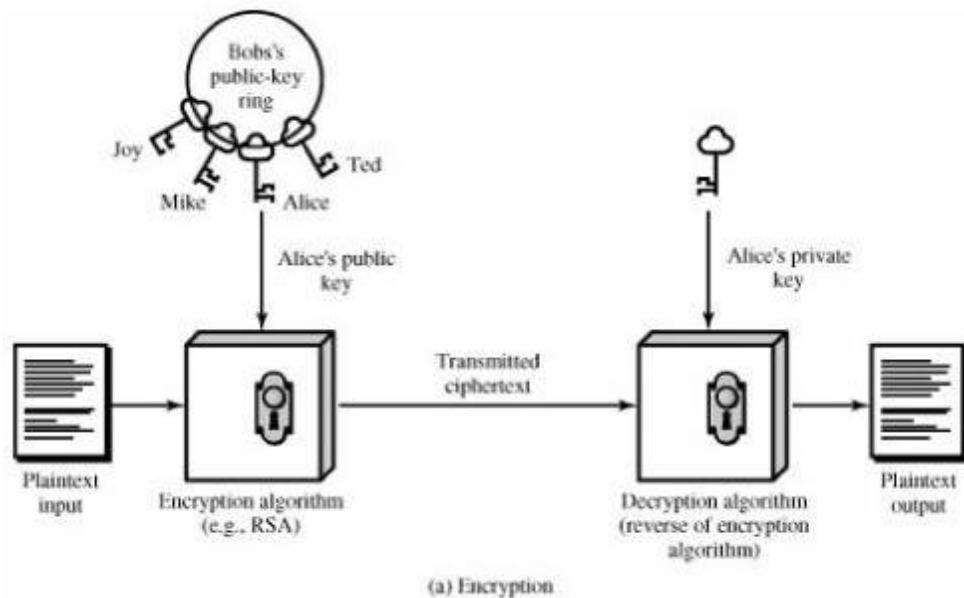
Jotta voitaisiin olla varmoja, että yhteys kohteeseen tulee oikeasta paikasta, eikä esimerkiksi yhteyttä ole kaapattu, joudutaan käyttämään varmenteita todentamisessa (Järvinen 2003, 121). Varmenne on sähköinen todistus, jonka avaamiseen tarvitaan varmenteen myöntäjän julkinen avain. Varmenne koostuu joukosta tietoa, jonka varmenteen myöntäjä on pakannut tiivisteeksi ja digitaalisesti allekirjoittanut. (Järvinen 2003, 159.)

4.2 Julkisen avaimen salaus

Julkisen avaimen salausmenetelmä toimii epäsymmetrisellä salausmenetelmällä, joten siinä käytetään kahta avainta: yhtä avainta salaamiseen ja toista salauksen purkuun. (Stallings 2005, 258). Salaamiseen käytettävää avainta kutsutaan julkiseksi avaimeksi. Salauksen purkuun käytettävää julkisesta avaimesta johdettua avainta kutsutaan salaiseksi tai yksityiseksi avaimeksi. (Hakala ym. 2006, 375.) Salauksen purkamiseen käytettävää avainta ei pystytä selvittämään salaamiseen käytettävän avaimen ja siinä käytettävän algoritmin perusteella. (Stallings 2005, 258.) Salainen avain saadaan selville julkisesta avaimesta ainoastaan jos tiedetään kuinka avainparit luotiin. (Hakala ym. 2006, 375). Se voidaan kuitenkin purkaa käyttämällä brute-force-hyökkäystä. Siinä kokeillaan kaikkia mahdollisia avaimia, kunnes oikea löytyy. Tietokoneiden tehojen kasvaessa mahdollistuu aina vain nopeampi avainten kokeilu. Tähän on pyritty vastaamaan kasvattamalla avainten kokoa. Kokojen kasvattaminen kuitenkin hidastaa salaamista ja salauksen purkua. (Stallings 2005, 266.)

Kuviossa 1 on esitettyä julkisella avaimella salaamisen toiminta. Bobilla oleva Alicen julkinen avain ja Alicen yksityinen avain ovat luotuja avainpareja. Bob valitsee julkisista avaimistaan Alicelta saamansa Alicen julkisen avaimen. Alicen

julkisen avaimen ja salausalgoritmin avulla tiedosto salataan. Tämän jälkeen tiedosto lähetetään salattuna Alicelle. Koska tiedosto on salattu Alicen julkisella avaimella, pystyy ainoastaan hän avaamaan sen oman yksityisen avaimen ja salaamisen purkuun tarkoitetun algoritmin avulla.



Kuvio 1. Salaus julkisen avaimen menetelmällä (Stallings 2005, 262.)

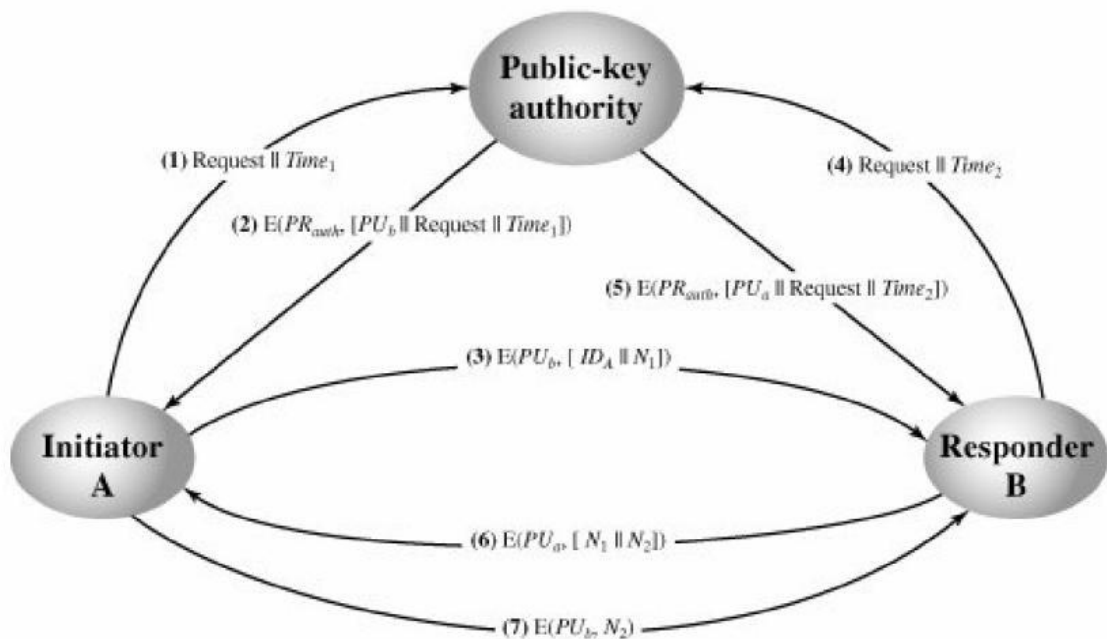
Esiintymistä jonain muuna käyttäjänä, ja saamalla näin julkisen avaimen käyttöön, on pyritty estämään antamalla julkiset avaimet luotetun kolmannen osapuolen jaettaviksi.

Kuviossa 2 on esitettyä kuinka A lähettää luotetulle kolmannelle osapuolelle lähetyksellä (Time₁) merkityn pyynnön avaimen saamiseksi. Kolmasosapuoli lähettää A:lle paketin joka sisältää B:n avaimen (PU_b) ja alkuperäisen A:n lähettämän aikaleimatun viestin. Paketti lähetetään salattuna kolmannen osapuolen yksityisellä salausavaimella (PR_{auth}). A:lla on hallussaan kolmannen osapuolen julkinen avain, joten hän pystyy purkamaan viestin. Näin A voi varmistaa että avain tuli kolmannelta osapuolelta ja verrata lähettämäänsä viestiä paketin sisältämään viestiin ja aikaleimaan.

Tämän jälkeen A salaa haluamansa viestin B:n julkisella avaimella ja lähettää sen B:lle. Salattu paketti sisältää A:n tunnisteiden (ID_A) ja A:n luoman sekalaisen merkkijonon (N_1).

Kun B on saanut A:lta salatun viestin, hakee hän A:n julkisen avaimen kolmannelta osapuolelta samoin kuin A haki B:n avaimen. Nyt B pystyy purkamaan A:n lähettämät viestit sekä salaamaan ja lähettämään viestejä A:lle. Tämä taas pystyy omalla yksityisellä avaimellaan purkamaan B:n lähettämät viestit ja B:n julkisella avaimella salaamaan viestejä B:tä varten.

Jotta A olisi varma että viesti meni B:lle perille, lähettää B lopuksi vielä A:n viestin sisältämän sekalaisen merkkijonon (N_1) ja itse luomansa sekalaisen merkkijonon (N_2) A:lle salattuna A:n julkisella avaimella. A lähettää tämän jälkeen vastauksena B:n luoman merkkijonon (N_2) salattuna B:n julkisella avaimella. (Stallings 2005, 291-293.)

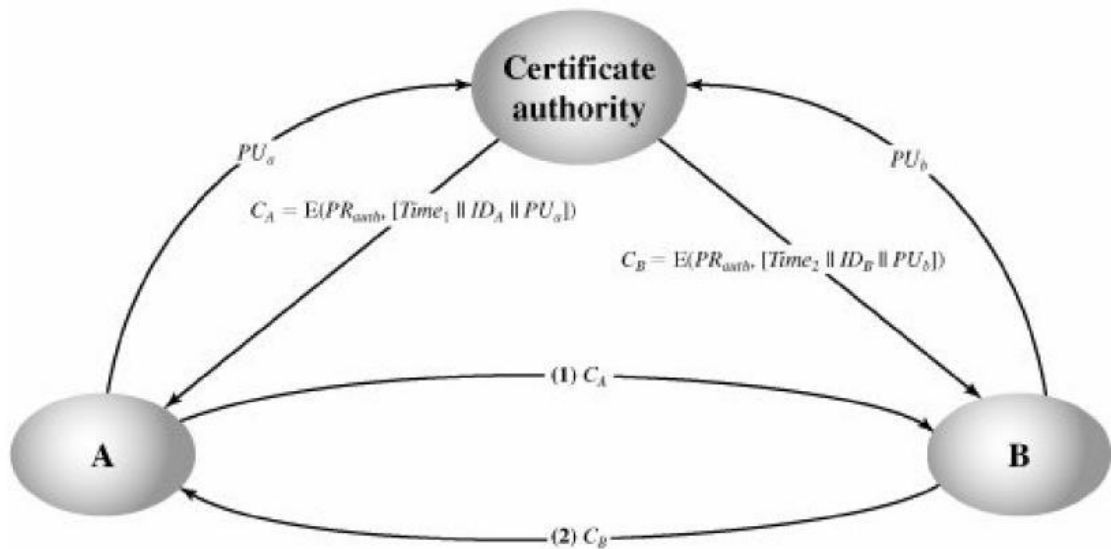


Kuvio 2. Julkisen avaimen jakajana kolmas osapuoli (Stallings 2005, 293).

Kolmannen osapuolen jakamilla avaimilla on heikkoutena kolmannen osapuolen luotettavuus. Julkisiin avaimiin on saatettu päästä käsiksi, esimerkiksi tietomurron avulla. (Stallings 2005, 293.)

Kolmannen osapuolen jakamien avaimien sijaan, voidaan myös käyttää sertifikaatti-varmennusta. (Stallings 2005, 294). Lähes kaikissa laajamittaisissa julkisen avaimen järjestelmissä käytetään varmenneita. (Hakala ym. 2006, 376). Siinä avaimeen liitetään avaimen omistajan tunniste. Kolmas osapuoli luo näiden pohjalta avaimen omistajalle sertifikaatin, johon on liitetty kolmannen osapuolen allekirjoitus. Kolmantena osapuolena toimii yleensä jokin virallinen taho. (Stallings 2005, 294.) Tunnisteen, avaimen ja allekirjoituksen lisäksi sertifikaatti sisältää usein myöntäjän nimen, voimassaoloajan ja sarjanumeron. Kysymällä sertifikaatin myöntäjältä aitoutta voidaan olla varmoja, että sertifikaatin omaava on oikea henkilö ja sertifikaatti voimassaoleva. (Ruohonen 2002, 266.) Sertifikaatin sisältämistä tiedoista saadaan selville sertifikaatin omistajan tunniste, jota voidaan verrata avaimeen liitettyyn tunnisteeseen.

Kuviossa on 3 esitettyä kuinka A lähettää sertifikaatin myöntäjälle julkisen avaimensa (PU_a). Sertifikaatin myöntäjä lähettää A:lle sertifikaatin (C_A), joka sisältää A:n tunnisteen (ID_A), voimassaoloajan (T) ja A:n julkisen avaimen (PU_a). Sertifikaatti on salattu sertifikaatin myöntäjän yksityisellä avaimella (PR_{auth}). Koska sertifikaatti on salattu yksityisellä avaimella, pystytään se purkamaan ainoastaan sertifikaatin myöntäjän julkisella avaimella. Näin varmistutaan siitä että sertifikaatti on sertifikaatin myöntäjältä. Tämän jälkeen A lähettää sertifikaatin (C_A) B:lle. B avaa sertifikaatin sen myöntäjän julkisella avaimella. Nyt B pystyy lähettämään A:lle viestejä salattuna A:n julkisella avaimella. B pystyy myös halutessaan varmistamaan sertifikaatin aitous ja voimassaolo sertifikaatin myöntäjältä. B hakee sertifikaatin samoin kuin A, lähettämällä oman julkisen avaimensa (PU_b) sertifikaatin myöntäjälle. (Stallings 2005, 295.)



Kuvio 3. Sertifikaatin noutaminen (Stallings 2005, 294).

4.3 SSL-salaus

Vuonna 1994 Netscape Communications Corporation kehitti SSL-salausprotokollan. Sitä käytetään mm. HTTPS, SSMTP, NNTP, FTPS ja Telnets-protokollissa. (Ruohonen 2002, 289.) Vaikka SSL-salausta käytetäänkin usein vain palvelimen aitouden varmentamiseen, voidaan sitä käyttää varmistamaan että yhteys on luottamuksellinen, viestit ehyitä ja osapuolten aitouden varmentamiseen. (Hakala ym. 2006, 390). Jos yhteyden aikana on tapahtunut jotain poikkeavaa, lähetetään hälytysviesti ja virhe joko korjataan tai yhteys katkaistaan. Jos viesti muuttuu matkalla, johtaa se aina yhteyden katkaisemiseen. (Ruohonen 2002, 290.)

SSL:n toiminta jakaantuu kahteen osaan: Handshake-protokolla ja Record Layer. Handshake:ssä lähetetään työasemalle viesti. Viesti sisältää tiedon siitä mitä salakirjoitusmenetelmiä työasema osaa. Palvelin valitsee tämän jälkeen salakirjoitusmenetelmän ja lähettää tiedon siitä työasemalle. Jos osapuolien tarvitsee tunnistaa toisensa sertifikaateilla, tapahtuu se tässä vaiheessa. Lopuksi sovitaan käytettävästä istuntoavaimesta. Kaikki viestit lähetetään kättelyn jälkeen salattuna. Record Layer-protokollan vastuulle jää viestien

osioiminen, pakkaaminen, salakirjoittaminen ja eheyden varmistaminen. (Hakala ym. 2006, 390 - 391.)

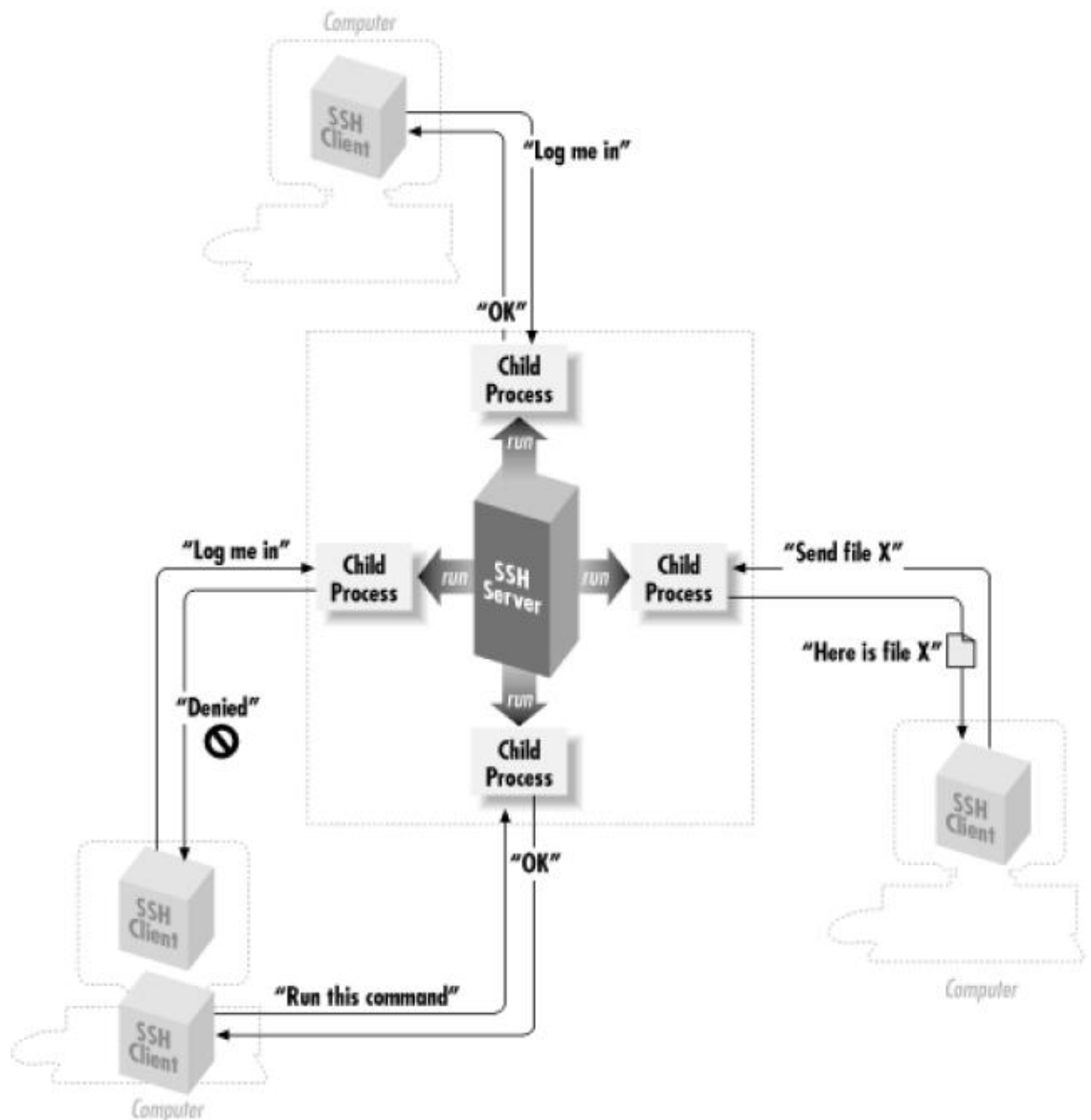
SSL:ää ei voida estää tai pakottaa toimimaan asiakkaan puolelta. SSL vaatii palvelun ylläpitäjältä tarvittavien ohjelmien lisäksi tarvittavia varmenteita. (Hakala ym. 2006, 391.)

4.4 SSH-salaus

Tatu Ylönen kehitti vuonna 1995 SSH-salausprotokollan. Ensimmäinen versio sisälsi käyttäjämäärien kasvun mukana tulleita ongelmia ja rajoituksia, joten 1996 julkaistiin SSH-2. Siinä SSH-1:n sisältämiä ongelmia oli korjattu ja se tuki useampaa algoritmia. Ongelmia ei voitu korjata menettämättä taaksepäin yhteensopivuutta SSH-1:n kanssa. Lisäksi SSH-2 lisenssit olivat erittäin paljon rajoitetumpia kuin SSH-1:n. Tämän vuoksi SSH-2 ei heti korvannut SSH-1 versiota, vaan vanhempi versio oli usean vuoden ajan suosittumpi. (Barrett & Silverman 2001, 31.)

SSH-salausprotokollaa käytetään etäyhteyksien luontiin, tiedon salaamiseen ja lähettämiseen, sekä käyttäjän tunnistamiseen (Barrett & Silverman 2001, 8). Se pyrkii suojaamaan mm. IP-huijauksilta, DNS-huijauksilta, tietojen urkinnalta, tiedon manipuloimiselta ja palvelinyhteyden väärentämiseltä (Acheson & Carasik 2001).

SSH-arkkitehtuuri toimii asiakas/palvelin periaatteella. Kuviossa 4 SSH-asiakasohjelmat lähettävät pyyntöjä SSH-palvelinohjelmalle. Kaikki palvelimen ja asiakkaan välillä tapahtuva tiedonsiirto on suojattu kryptatauksella ja eheyden varmistuksella. (Barrett & Silverman 2001, 20.)



Kuvio 4. SSH-arkkitehtuuri (Barrett & Silverman 2001, 20).

Palvelinohjelma joko hyväksyy tai hylkää asiakkaan pyynnön tunnistamisen perusteella. Tunnistus voi tapahtua salasanan tai julkisen avaimen perusteella. Julkisen avaimen perusteella tunnistamista käytetään usein automatisoiduissa toiminnoissa. Lisäksi SSH voidaan integroida usean tunnistamismenetelmän kanssa. Näitä ovat mm. Kerberos, SecurID, PGP, TIS Gauntlet ja PAM. (Barrett & Silverman 2001, 8.) SSH:ssa on mahdollista muuttaa käytettäväksi

salausalgoritmiksi ei mitään. Tämä on tarkoitettu ainoastaan virheiden korjaukseen, joten voidaan sanoa että yhteyksien välinen tiedonsiirto tapahtuu aina salattuna. SSH:n salaukseen käyttämiä algoritmeja SSH:n versiosta riippuen ovat mm. DES, 3DES, IDEA Blowfish, Twofish, Arcfour ja Cast128-cbc.(Acheson & Carasik 2001.) Salauksen lisäksi SSH hoitaa tiedon eheyden varmistamisen. Eheydellä varmistetaan, ettei tietoa ole muokattu matkanvarrella. (Barrett & Silverman 2001, 24.)

4.4 Salaaminen hajautusfunktiolla

Hajautusfunktioita kutsutaan usein myös hash tai digest-funktioiksi. (Ruohonen 2002, 264). Hajautusfunktiolla salattu viesti on yksisuuntainen, joten sitä ei ole tarkoitettu avattavaksi. Hajautusfunktion läpi käynyt viesti on aina vakiokokoinen, joten alkuperäisen viestin pituutta ei ole mahdollista saada selville hajautusfunktion tuottamasta hash-koodista. (Stallings 2005, 329.)

Hajautusfunktiossa läpikäynyt viesti muutetaan tiivisteeksi. Viestin kaikki merkit vaikuttavat tiivisteeseen tulokseen. Vaikka erilaiset viestit voivat teoriassa tuottaa saman tiivisteeseen, on kahta erilaista viestiä vaikea löytää. (Ruohonen 2006, 264.) Jos kaksi samanlaista viestiä poikkeaa yhdenkin bitin toisistaan, tuottaa hajautusfunktiolla salaaminen erilaisen hash-koodin. (Stallings 2005, 329). Koska saman hash-koodin tuottavat viestit poikkeaisivat täysin toisistaan, vastaanottaja huomaisi välittömästi viestin muuttuneen. (Ruohonen 2006, 264.) Hajautusfunktiota käytetäänkin usein salasanojen tallentamiseen. Tällöin tallennettua salasanaa ei avata, vaan käyttäjän syöttämä salasana muutetaan hajautusfunktiolla hash-koodiksi. Näin voidaan verrata, onko koodi sama kuin tallennetussa salasanassa.

5 Opinnäytetyöskentelyn seurantaan avustavan järjestelmän toteutus

5.1 Järjestelmän vaatimukset

Hanna Rämö oli tutkinut opinnäytetyössään ASP.NET ja selainpohjaisen sovelluksen suunnittelu tietojenkäsittelyn koulutusohjelman opinnäyteseurantajärjestelmälle asetettavia vaatimuksia. Vaatimukset oli hyväksytetty opettajankokouksessa. (Rämö 2008, 32.)

Tarkoituksena oli korvata opinnäytetyöskentelyn seurantaan tarkoitettu paperinen järjestelmä, sähköisellä, reaaliaikaisella ja aikaan ja paikkaan sitoutumattomalla järjestelmällä (Rämö 2008, 35). Uuden järjestelmän keskeisiä vaatimuksia olivat helppokäyttöisyys, reaaliaikainen tietojen ylläpito, sekä tietoturvallisuus. Järjestelmän toiminnoille asetetut vaatimukset voidaan jaotella kahteen osaan: opettajan roolin omaavaan käyttäjään ja opiskelijan rooliin omaavaan käyttäjään. Opiskelijalle kuuluvat omien tietojen päivittäminen, ilmoittautuminen seminaareihin ja seminaaripäivien tarkastelu. Opettajalle kuuluvat seminaari- ja kypsyysnäytepäivien merkitseminen, osallistujien ryhmäkuittaus haun perusteella, ilmoittautuneiden oppilaiden listan näkeminen ja raporttien tulostus.

Jotta järjestelmän ominaisuudet olisivat mahdollista toteuttaa niin, ettei opiskelijalla ole mahdollisuutta päästä käsiksi toisen opiskelijan tietoihin, eikä opettajan lisäämän tilaisuuden muuttamiseen, on järjestelmälle luotava roolipohjainen ja käyttäjän tunnistava kirjautuminen (Rämö 2008, 40, 41). Käyttäjän kirjautumisen helpottamiseksi kirjautumistunnusten täytyisi olla helppomuistisia ja salasana käyttäjän vaihdettavissa (Rämö 2008, 41). Suurin osa Turun ammattikorkeakoulun järjestelmistä on kaupallisia, joten lisäpalikoiden rakentaminen vaatii paljon selvitystyötä järjestelmän toimittajalta ja ammattikorkeakoululta. Tällöin uuden järjestelmän integroinnin kokonaiskustannuksista tulee suuri. Tämän vuoksi Turun ammattikorkeakoulun

integrointeja tehdään vain, jos lisättävä ominaisuus koskee koko koulua. Koska järjestelmä tulee aluksi vain tietojenkäsittelyn koulutusohjelman käyttöön, Hanna Rämön opinnäytteessä ehdotettua järjestelmän liittämistä johonkin jo olemassa olevaan järjestelmään ei toteutettu.

Vaikka järjestelmä toteutettiin roolipohjaisena kirjautumisena, Hanna Rämön vaatimusmäärittelyn mukaista salasanan unohtuessa tarvittavaa ylläpitäjän apua salasanan palautukseen ei tarvita. Tutkittuani eri tunnistautumis- ja ylläpitovaihtoehtoja päädyin järjestelmään, jonka osittaisena ylläpitäjänä toimivat järjestelmän opettajat. He saavat oikeudet lisätä, muokata ja poistaa käyttäjiä sekä uusia käyttäjien salasanat kuitenkin niin, ettei opettaja saa tietoonsa käyttäjän käyttäjätunnusta, muulloin kuin käyttäjää luotaessa, eikä saa tietoonsa käyttäjän salasanaa. Tämän mahdollistaa käyttäjää luotaessa ja salasanaa uusittaessa salasanan automaattinen generointi ja sen lähettäminen käyttäjälle sähköpostiviestinä.

5.2 Toiminnot ja ominaisuudet

5.2.1 Kirjautuminen ja kirjautumistunnukset

Alkuperäisenä ideana oli toteuttaa järjestelmä jonkin jo olemassa olevan järjestelmän yhteyteen. Ollessani yhteydessä Turun ammattikorkeakoulun IT-suunnittelijaan Tuomo Karppiseen ja IT-pääsuunnittelijaan Pauli Lehtoseen kävi ilmi, että järjestelmän liittäminen jo olemassa oleviin järjestelmiin olisi ollut kallis toteuttaa, joten päädyin toteuttamaan täysin erillisen itsenäisen järjestelmän.

Järjestelmä käyttää hyväkseen ASP.NET forms-kirjautumista. Järjestelmään kirjautuminen tapahtuu käyttäjätunnuksen ja salasanan perusteella. Järjestelmässä ei voida käyttää kahta samaa käyttäjätunnusta. Jos käyttäjä yrittää ottaa käyttöön jo olemassa olevan käyttäjätunnuksen, tulee siitä käyttäjälle virheilmoitus. Salasana ja käyttäjätunnus ovat käyttäjän

muutettavissa haluamukseen tunnusten vaihtosivun kautta. Salasanan unohtuessa voi opettaja roolin omaava käyttäjä tilata uuden tunnuksen. Tunnus generoidaan automaattisesti ja lähetetään käyttäjän sähköpostiin. Näin tunnusten uusimista varten ei tarvita järjestelmälle erillistä ylläpitäjää.

Salasana tallennetaan tietokantaan hajautusfunktiolla muutettuna. Käyttäjän kirjautuessa syötetty salasana muutetaan hash-muotoon ja verrataan sitä tietokannan muutettuun salasanaan. Salasanaa ei siis missään vaiheessa muuteta takaisin alkuperäiseen muotoon. Näin lukemalla suoraan tietokannasta salasanoja ei saada selville käyttäjän salasanaa, vaan pelkästään rivi hash-koodia. Koska teoriassa hajautusfunktio saattaa luoda kahdesta eri merkkijonosta saman arvon, käyttäjätunnus on jokaiselle käyttäjälle yksilöllinen.

Kirjautuminen voidaan halutessa suojata varmenteella, jonka käyttö voidaan kohdistaa koko sivustoa tai vain tiettyjä sivuja koskevaksi. Käyttöönoton ohjeistus osiossa käsitellään sovelluksen asetusten muuttamista varmenteen käyttöön ottamiseksi.

Kun käyttäjätunnus ja salasana tunnustetaan oikeiksi, käyttäjälle luodaan hetkellinen eväste. Eväste sisältää käyttäjän käyttäjätunnuksen ja tietokannan perusteella haetun roolin sekä käyttäjän tunnisteiden. Rooli mahdollistaa pääsyn ainoastaan sille oikeuttaville sivuille. Tunnisteiden perusteella käyttäjää koskevat tiedot haetaan tarvittaessa tietokannasta.

Uloskirjautumisessa käyttäjä ohjataan kirjautumissivulle. Jos käyttäjä on kirjautuneena järjestelmään, käyttäjä kirjataan ulos ja käyttäjän istunto lopetetaan. Näin erillistä uloskirjautumissivua ei tarvitse toteuttaa.

Tunnusten vaihto -sivulta käyttäjä voi vaihtaa joko käyttäjätunnuksen tai salasanansa. Vaihtaminen edellyttää nykyisen käyttäjätunnuksen ja salasanan syöttämistä. Tällä estetään käyttäjän unohtaessa kirjautua ulos järjestelmästä, ulkopuolisen käyttäjän mahdollisuus vaihtaa kirjautumistunnuksia.

Salasanaa vaihtaessa pyydetään käyttäjää syöttämään uusi salasana kahteen kertaan. Syötettyjä uusia salasanoja verrataan keskenään. Salasanan vaihto

onnistuu vain niiden täsmätessä. Tällä varmistetaan, ettei käyttäjä tee näppäilyvirhettä valitessaan uutta salasanaa.

Kun käyttäjätunnus tai salasana on vaihdettu, annetaan siitä ilmoitus käyttäjälle ja käyttäjä kirjataan ulos järjestelmästä. Koska käyttäjä tunnistetaan järjestelmässä evästeeseen lisätyn käyttäjätunnuksen perusteella, eväste täytyy luoda uudelleen kirjautumalla sisään.



The screenshot shows the user profile page for 'ONSeuranta > Käyttäjätunnuksen tai salasanan vaihto'. The page has a blue sidebar on the left with navigation links: 'Vaihda käyttäjätunnus tai salasana', 'Omat tiedot', 'Ilmoitukset', 'Kalenteri', 'Käyttäjien hallinta', 'Raporttien tulostaminen', 'Vaihda käyttäjätunnus tai salasana', and 'Kirjaudu ulos'. The main content area has a title 'ONSeuranta > Käyttäjätunnuksen tai salasanan vaihto' and a 'Ohje' link. Below the title are two radio buttons: 'Vaihda salasana' (selected) and 'Vaihda käyttäjätunnus'. There are four input fields: 'Käyttäjätunnus', 'Salasana', 'Uusi salasana', and 'Valvasta uusi salasana'. At the bottom, there are links for 'Vaihda salasana' and 'Peruuta'. The logo of Turun Ammattikorkeakoulu is visible at the top left of the page.

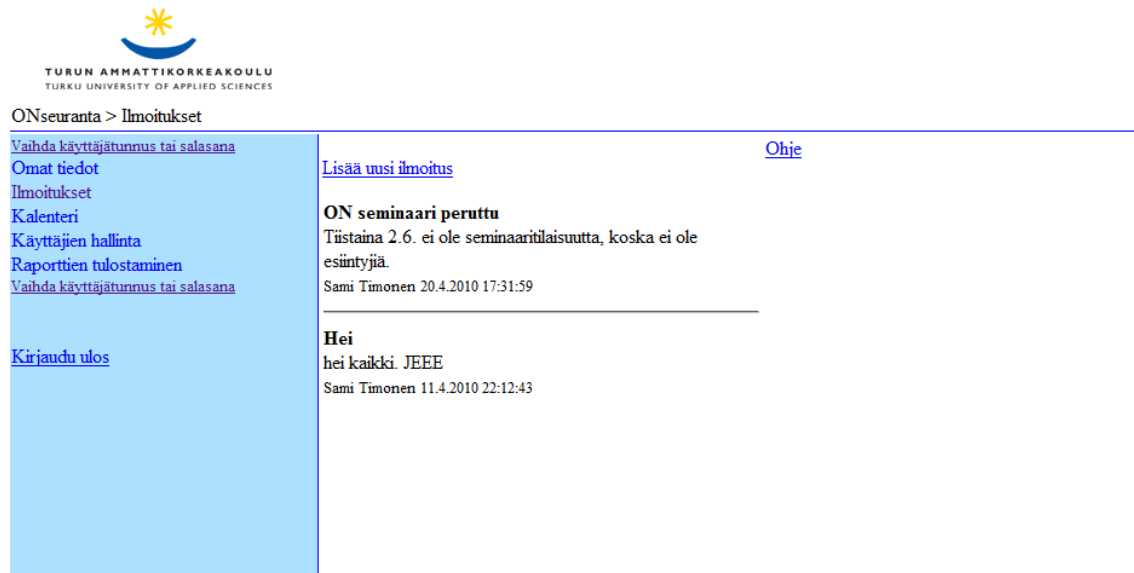
Kuva 2 Tunnusten vaihto –sivu

Kuvassa 2 on tunnusten vaihtoa koskeva sivu. Käyttäjä voi valikoida vaihtaako salasanan vai käyttäjätunnuksen. Tunnusten vaihto edellyttää käyttäjänimen ja salasanan syötön.

5.2.2 Ilmoitukset

Ilmoitukset -sivulla opettajaroolin omaava käyttäjä voi lisätä sivulle otsikon omaavan ilmoitustekstin oppilasroolin ja muiden opettajaroolin omaavien käyttäjien nähtäväksi. Lisätyssä ilmoituksessa näytetään otsikon ja tekstin lisäksi ilmoituksen tehneen nimi sekä lisäys päivä ja aika. Halutessaan käyttäjä voi muokata tai poistaa ilmoituksensa. Käyttäjä ei pysty muokkaamaan tai poistamaan toisten käyttäjien luomia ilmoituksia. Kun käyttäjä muokkaa omaa ilmoitustaan, päivämäärä ja aika muuttuvat muokkaamisaikaa vastaavaksi.

Jotta sivu ei täytyisi vanhoista ilmoituksista, ainoastaan uusimmat kolmekymmentä ilmoitusta näytetään sivulla.



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

ONseuranta > Ilmoitukset

[Vaihda käyttäjätunnus tai salasana](#)

[Omat tiedot](#)

[Ilmoitukset](#)

[Kalenteri](#)

[Käyttäjien hallinta](#)

[Raporttien tulostaminen](#)

[Vaihda käyttäjätunnus tai salasana](#)

[Kirjaudu ulos](#)

[Lisää uusi ilmoitus](#)

ON seminaari peruttu
Tiistaina 2.6. ei ole seminaaritilaisuutta, koska ei ole esiintyjä.
Sami Timonen 20.4.2010 17:31:59

Hei
hei kaikki. JEEE
Sami Timonen 11.4.2010 22:12:43

[Ohje](#)

Kuva 3 Ilmoitukset –sivu

Kuvassa 3 nähdään ilmoituksia koskeva sivu. Sivulla näkyvät opettajien jättämät ilmoitukset järjestettynä ilmoituksen jättöajan mukaan. Opettajakäyttäjällä on näkyvissä Lisää uusi ilmoitus linkki, jota painamalla avautuu mahdollisuus luoda ilmoitus järjestelmään.

5.2.3 Kalenteri ja ilmoittautuminen

Ohjelma sisältää kalenterisivun, jossa opettajaroolin omaava käyttäjä voi lisätä seminaaritilaisuuden valitsemalleen päivälle tai halutessaan jokaiselle valitun päivän viikonpäivälle kyseiselle kuukaudelle. Lisätessä tilaisuutta kirjoitetaan kellonaika ja tila, jossa tilaisuus pidetään. Taulussa näkyvät ilmoittautuneet osallistujat sekä ilmoittautuneet seminaarinpitäjät, heidän opinnäyteaiheensa ja pidettävä seminaarityyppi. Lisäksi opettajaroolissa oleva voi muokata valitsemansa kuukauden tapahtumia sekä merkitä tilaisuuteen ilmoittautuneille opiskelijaroolissa oleville läsnäolot ja suoritukset halutuille seminaareille. Merkitsemisen yhteydessä tallennetaan tietokantaan myös merkinnän tehneen nimi, myöhempää tarkastelua varten.

Jos kalenteriin on merkitty tilaisuuksia, voi oppilasroolissa oleva käyttäjä ilmoittautua haluttuun tilaisuuteen kuuntelijaksi tai esiintyjäksi. Jos käyttäjä ilmoittautuu esiintyjäksi, täytyy käyttäjän valita pitämänsä seminaarityyppi. Käyttäjä ei voi ilmoittautua kuin kerran kuuntelijaksi tai esiintyjäksi yhteen tilaisuuteen. Jos käyttäjä on kuitenkin ilmoittautunut jo kuuntelijaksi tilaisuuteen ja hän päättää myöhemmin myös esiintyä siinä, voi hän ilmoittautua tilaisuuteen esiintyjäksi.

Kalenterissa valitun päivän reunat näkyvät tummanpunaisena katkoviivana. Jos kalenteriin on merkitty tilaisuus, näkyy päivä kalenterissa sinisenä. Jos valittuna päivänä on useita tilaisuuksia, näytetään kaikki päivän tilaisuudet valinta mahdollisuudella. Ensimmäinen tilaisuus on oletuksena valittuna. Valitusta tilaisuudesta näytetään ilmoittautuneet seminaarinpitäjät aiheineen ja seminaarityypeineen. Opettajaroolissa olevalle näytetään lisäksi tilaisuuteen ilmoittautuneet osallistujat.

Opettajaroolin omaava voi halutessaan tallentaa Excel-tauluna valitun kuukauden tapahtumat, valittuun tilaisuuteen ilmoittautuneet esiintyjät ja kuuntelijat sekä valitun tilaisuuden kuitatut suoritukset.

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

ONSeuranta > Kalenteri

[Vaihda käyttäjätunnus tai salasana](#)
[Omat tiedot](#)
[Ilmoitukset](#)
[Kalenteri](#)
[Käyttäjien hallinta](#)
[Raporttien tulostaminen](#)
[Vaihda käyttäjätunnus tai salasana](#)
[Kirjautu ulos](#)

≤ huhtikuu 2010 ≥

ma	ti	ke	to	pe	la	su
		1	2	3	4	
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

[Lisää kalenterin tilaisuuksia](#)

[Näytä ja muokkaa valitun kuukauden tilaisuuksia](#)

[Näytä ja kuittaa suoritukset](#)

Aika: 15.4.2010 12:15
Tila: Lemminkäinen
Ohjaava opettaja: Sami Timonen
Tilaisuus: Kypsyysnäyte

Ilmoittautuneet esiintyjät
sami timonen
ONjärjestelmän luonti Kypsyysnäyte

Aika: 15.4.2010 12:15
Tila: A120
Ohjaava opettaja: Sami Timonen
Tilaisuus: Seminaari

Ilmoittautuneet kuuntelijat

Esiintyjät

Nimi	OpiskelijaNumero	Kypsyysnäyte
sami timonen	0202020	<input checked="" type="checkbox"/>

Osallistujat
[Toteuta valitut merkinnät](#) [Peruuta](#)

Kuva 4 Kalenteri -sivu

Kuvassa 4 on Kalenteri –sivu. Käyttäjä on klikannut kalenterista päivämäärä, jolloin aukeaa kyseisen päivämäärän tilaisuudet. Käyttäjä on valinnut tilaisuuksista keskimmäisen, jolloin näytetään tilaisuuteen ilmoittautuneet. Koska kyseisen käyttäjän rooli on opettaja, voi hän lisätä kalenteriin tilaisuuksia, muokata ja poistaa valitun kuun tilaisuuksia, sekä merkitä valitun tilaisuuden tilaisuuksia. Lisäksi käyttäjä voi tulostaa tietoja klikkaamalla taulukkokuvaketta. Kuvassa käyttäjä on klikannut Näytä ja kuittaa suoritukset –linkkiä, jolloin on avautunut tilaisuuksien suoritusmerkintöjen muokkausvalikko.

5.2.4 Etsintä ja raporttien tulostaminen


Raporttien tulostaminen -sivu on näkyvässä ja käytettävissä ainoastaan opettajaroolin omaavalla käyttäjällä.

Sivulta voidaan katsoa kaikki opiskelijat, opiskelijanumeroineen, ryhmätunnuksineen, aiheineen ja toimeksiantajineen, sekä haluttaessa näyttää vain ne opiskelijat, joilla on tai ei ole aiheetta, tai näyttää pelkät toimeksiantajayritykset.

Opiskelijoista koostuvalta listalta voidaan valitsemalla opiskelija nähdä opiskelijan kuitatut seminaariesitykset ja kuitattujen paikallaolojen määrä ensimmäisissä, toisissa ja loppuseminaareissa. Seminaariesityksistä nähdään seminaarintyyppi, -aika, -paikka ja kuka seminaariesityksen on kuitannut pidetyksi.

Sivu sisältää hakutoiminnon, jossa kirjoittamalla hakukenttään opiskelijan nimi, opiskelijanumero, ryhmätunnus tai opinnäytteen nimi näytetään kriteerillä löytyneet opiskelijoiden nimi, opiskelijanumero ja opinnäytetyöaihe. Näistä voidaan valikoida opiskelija, jonka seminaaritiedot näytetään. Listan ensimmäinen opiskelija on oletuksena valittuna. Seminaaritiedoista nähdään pidetyt seminaarit, niiden aika ja paikka ja sen kuitannut opettaja. Lisäksi näytetään opiskelijan kuitattujen paikallaolojen määrä ensimmäisissä, toisissa ja loppuseminaareissa.

Valitut tai haetut tiedot voidaan haluttaessa tallentaa Excel-taulukkona tietokoneelle.


TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

ONseuranta > Haku ja raportit

[Vaihda käyttäjätunnus tai salasana](#) [Ohje](#)

[Omat tiedot](#)

[Ilmoitukset](#)

[Kalenteri](#)

[Käyttäjien hallinta](#)

[Raporttien tulostaminen](#)

[Vaihda käyttäjätunnus tai salasana](#)

[Kirjaudu ulos](#)

Mistä hahuat tulostettavat tiedot:


Kaikki opiskelijat

Opiskelijat joilla on aihe

Opiskelijat joilla ei ole aihetta


Toimeksiantajat

	Tunnus	Nimi	Sähköposti	Aihe	Toimeksiantaja	Yritys
Valitse	0501235	Ville Peltonen	samitimonen@huukku.com	ON R&D		
Valitse	0202020	sami timonen	samitimonen@huukku.com	ONJärjestelmän luonti	Jumppanen	Turun AMK

 **142s06TJ**
0202020
sami timonen
ONJärjestelmän luonti

Opinaytteen edistyminen **Paikallaolot seminaareissa**

Seminaarityyppi: Kypsyysnäyte Ensimmäinen
Tila: A120 Aika: 15.4.2010 seminaari:0
12:15:00 Toinen
Ohjaava opettaja: Sami seminaari:0
Timonen Loppuseminaari:0



Kuva 5 Raporttien tulostaminen –sivu

Kuvassa 5 on raporttien tulostamista koskeva sivu. Opiskelijat joilla on aihe on valittuna, jolloin nähdään listana kaikki opiskelijat joilla on aihe. Käyttäjä on klikannut sami timonen nimistä opiskelijaa, jolloin on avautunut suoritettua opinnäytettä koskevat tilaisuudet, sekä läsnäolojen määrät seminaareissa. Ylhäällä olevaan hakupalkin avulla voidaan hakea opiskelija nimen, opinnäytteen, opiskelijanumeron, tai toimeksiantajan perusteella. Taulukkokuvakkeita klikkaamalla tulostetaan kuvakkeen yläpuolella olevat tiedot.

5.2.5 Käyttäjienhallinta

Käyttäjienhallinta -sivulta opettajaroolin omaava käyttäjä voi lisätä, muokata ja poistaa käyttäjiä. Lisäksi hän voi uusia halutun käyttäjän salasanan. Opiskelijaroolin omaava käyttäjä ei näe käyttäjienhallintaa eikä omaa siihen käyttöoikeutta.

Käyttäjä voidaan lisätä joko opiskelijaroolille tai opettajaroolille. Lisätessä käyttäjää tarkistetaan, onko syötetty käyttäjätunnus jo jonkin muun käyttäjän käytössä. Oletuksena opiskelijaroolille käyttäjätunnus on opiskelijanumero. Lisätessä käyttäjää opiskelijaroolille tarkistetaan lisäksi, onko opiskelijatunnus jo käytössä. Jos käyttäjätunnus tai opiskelijatunnus on jonkin muun käyttäjän käytössä, annetaan siitä ilmoitus käyttäjälle, eikä uutta käyttäjää lisätä. Kun käyttäjä on lisätty, sovellus generoi salasanan ja lähettää käyttäjätunnuksen ja salasanan uuden käyttäjän sähköpostiosoitteeseen.

Käyttäjienhallinnasta voidaan muokata opettajaroolin omaavan käyttäjän nimeä ja sähköpostia sekä opiskelijaroolin omaavan käyttäjän opiskelijanumeroa nimeä, sähköpostia ja ryhmätunnusta.

Poistettaessa käyttäjää järjestelmästä kysytään varmistus käyttäjän poistoon. Käyttäjä poistetaan ainoastaan, jos varmistus hyväksytään. Kun käyttäjä on poistettu, häntä ei voida enää palauttaa järjestelmään.

Uusittaessa käyttäjän salasanaa kysytään varmistus salasanan uusimiseksi. Jos varmistus hyväksytään, generoidaan uusi salasana ja lähetetään käyttäjän uusi salasana ja käyttäjätunnus hänen sähköpostiinsa. Näin salasana tulee vain salasanan omistajan tietoon.

ONSeuranta > Käyttäjien hallinta

[Vaihda käyttäjätunnus tai salasana](#) [Ohje](#)

[Omat tiedot](#)

[Ilmoitukset](#)

[Kalenteri](#)

[Käyttäjien hallinta](#)

[Raporttien tulostaminen](#)

[Vaihda käyttäjätunnus tai salasana](#)

[Kirjaudu ulos](#)

oppilas

Opiskelijanumero

Etnimi

Sukunimi

Sähköposti

Käyttäjätunnus

Ryhmäätunnus

Lisää

			Tunnus	Etunimi	Sukunimi	Sähköposti	Ryhmäätunnus
Muokkaa	Poista käyttäjä	Uusi salasana	0501235	Ville	Peltonen	samitimonen@hukku.com	142s07TL
Muokkaa	Poista käyttäjä	Uusi salasana	0502303	Onni	Opiskelija	samitimonen@hukku.com	142s07TJ
Muokkaa	Poista käyttäjä	Uusi salasana	0202020	sami	timonen	samitimonen@hukku.com	142s06TJ

Kuva 6 Käyttäjienhallinta –sivu

Kuvassa 6 on kuvattuna käyttäjienhallintaa koskeva sivu. Alasvetovalikossa on vaihtoehdot opettaja ja oppilas. Tätä vaihtamalla nähdään kaikki kyseisen roolin omaavat käyttäjät listassa, sekä kyseiselle roolille käyttäjän lisäys mahdollisuus. Syöttämällä uudelle käyttäjälle tarvittavat tiedot laatikoihin ja klikkaamalla Lisää painiketta luodaan uusi käyttäjä. Klikkaamalla halutun käyttäjän kohdalta Muokkaa linkkiä, voidaan muokata käyttäjän tietoja. Klikkaamalla halutun käyttäjän kohdalta Poista käyttäjä linkkiä poistetaan käyttäjän tiedot järjestelmästä. Uusi salasana linkkiä klikkaamalla rivillä olevalle käyttäjälle lähetetään uusi automaattisesti luotu salasana sähköpostiin.

5.3 Ohjeistus käyttöönottoon

5.3.1 Tietokanta ja tietolähteen määrittäminen

Sovelluksen tarvitsema tietokanta on nimeltään ONSeurantaDB. Se voidaan joko siirtää palvelimelle valmiina ONSeurantaDB.mdf-tiedostona tai suorittaa

tietokannan luova SQL-scripti ONSeurantaDB.sql. Jotkin palvelimen suojausasetukset saattavat estää suoran ONSeurantaDB.mdf-tietokannan käytön, tällöin on suositeltavaa luoda tietokanta SQL-scriptin avulla palvelimella. Luotu tietokanta täytyy olla nimeltään ONSeurantaDB.

Jotta sovellus löytää tietokannan, täytyy web.config-tiedoston connectionStrings-kohdan DataSource muuttaa palvelimen asetuksien mukaiseksi.

```
<connectionStrings>
  <!--Muuta connectionString kohta palvelimen asetuksien mukaiseksi-->
  <add name="CSTietokantayhteys" connectionString="Data Source=HUOM!
MUUTA TÄMÄ;Initial Catalog=ONSeurantaDB;Integrated Security=True"
providerName="System.Data.SqlClient" />
</connectionStrings>
<system.web>
```

Koodin kohta Data Source täytyy muuttaa tietokannan sijaintia vastaavaksi.

5.3.2 Varmenteen käyttö

Sivusto voidaan asentaa käyttämään varmenne-suojausta koko sivustolla tai vain salasanojen käsittelyn yhteydessä. Jos sivustolle ei haluta ottaa käyttöön ssl-suojausta, sslSuojaattu-kansiota ei tarvitse siirtää palvelimelle. Tällöin sivusto ei ole varmenne-suojaattu ja on siksi alttiimpi tietomurroille.

Jos halutaan käyttää ssl-suojausta koko sivustolla, täytyy web.config-tiedostoon forms-kohtaan lisätä määritys *requireSSL="true"*. Tällöin palvelimelle täytyy myös luoda määritys ssl-suojauksen käytölle.

```
<authentication mode="Forms">
  <!--Käytettäessä kokosivun suojausta lisätään määritys requireSSL="true".
  Käytettäessä vain osittaista ssl-suojausta requireSSL="true" jää pois
  ja loginUrl="Kirjaudu.aspx" tilalle laitetaan loginUrl="sslSuojaattu/Kirjaudu.aspx"-->
  <forms name="Kirjaudu" loginUrl="Kirjaudu.aspx"
protection="All" requireSSL="True" enableCrossAppRedirects="false" timeout="30" />
</authentication>
```

Koodissa Forms määrittäykseen on lisätty requireSSL määritys ja arvoksi annettu True.

Jos halutaan suojata vain salasanaa käsittelevät sivut (kirjautuminen, käyttäjienhallinta ja tunnustenvaihto) täytyy web.config-tiedoston forms-kohdan määrittäminen `login="Kirjaudu.aspx"` korvata määrittämisellä `login="sslSuojattu/Kirjaudu.aspx"`. Formsin login-määrittämisessä määritetään sivuston kirjautumissivu. Tälle sivulle käyttäjä ohjataan jos käyttäjä ei ole kirjautunut sisään järjestelmään.

Määrittämisestä lisäksi palvelimelle täytyy siirtää sslSuojattu-niminen kansio ja tämän sisältämät tiedostot `KayttajienHallinta.aspx`, `KayttajienHallinta.aspx.cs`, `Kirjaudu.aspx`, `Kirjaudu.aspx.cs`, `TunnustenVaihto.aspx`, `TunnustenVaihto.aspx.cs`, sekä `we.config`. Sivuston juurihakemistosta voidaan poistaa tiedostot `KayttajienHallinta.aspx`, `Kirjaudu.aspx` ja `TunnustenVaihto.aspx`.

Jotta sivun linkit ohjautuvat oikeaan paikkaan, täytyy ONSeuranta.master sivun `PanelNormaali-näkyvyysmääre` muuttaa näkymättömäksi ja `PanelOsittainSuojattu-näkyvyysmääre` muuttaa näkyväksi. Näkyvyysmääreen muuttaminen tapahtuu muuttamalla paneelin `visible`-arvoa. Arvo `False` muuttaa näkyvyyden näkymättömäksi ja arvo `True` muuttaa näkyvyyden näkyväksi.

```
<asp:Panel ID="PanelNormaali" runat="server" Visible="False">
```

ONSeuranta.master –sivun koodissa `PanelNormaali` näkyvyysmääreelle asetetaan arvon `False`.

```
<asp:Panel ID="PanelOsittainSuojattu" runat="server" Visible="False">
```

Koodin `PanelOsittainSuojattu` näkyvyysmääreelle asetetaan arvo `True`.

Myös osittaisessa varmennuksessa täytyy palvelimelle luoda määrittäminen `sslSuojattu`-kansiolle `ssl-suojauksen` käytölle. Osittainen suojattu yhteys kuluttaa resursseja vähemmän kuin kokosivuston suojattu yhteys, mutta enemmän kuin suojaamaton yhteys.

5.3.4 Sähköpostiasetukset

Käyttäjienluonti sekä salasanan uusiminen vaatii sähköpostin. Asetukset sähköpostia varten asetetaan `KayttajienHallinta.aspx.cs`-tiedoston

lähetäSähköposti-kohdasta muuttaa sähköpostintarjoajan smtp-osoite ja käytettävä portti sekä sähköpostiosoite ja salasana. Lisäksi muutetaan tarvittaessa sähköpostin ssl-määrittystä.

```
SmtClient smtp = new SmtClient("smtp.palvelin.com", 587);
```

```
smtp.UseDefaultCredentials = false;
```

```
smtp.Credentials = new System.Net.NetworkCredential("sähköposti@palvelin.com", "salasana");
```

```
smtp.EnableSsl = true;
```

Koodin *smtp.palvelin.com* tilalle kirjoitetaan sähköpostintarjoajan smtp-osoite, *587* tilalle kirjoitetaan käytettävä portti, *sähköposti@palvelin.com* tilalle kirjoitetaan sähköpostiosoite ja salasana tilalle kirjoitetaan sähköpostin salasana.

5.3.3 Ensimmäisen käyttäjän luonti

Jotta sivuston ensimmäinen käyttäjä voidaan luoda tietokantaan, täytyy sovelluksen ensimmäinen kirjautuminen suorittaa EkaKirjautuminen.aspx-sivun avulla. Tämä tapahtuu muuttamalla web.config-tiedoston forms-kohdan määritys login="Kirjaudu.aspx", määrittämiseksi login="EkaKirjautuminen.aspx". Näin voidaan kirjautua sovellukseen EkaKirjautuminen.aspx, sivuston avulla ja luoda ensimmäisen opettaja-käyttäjän. Opettaja-käyttäjät pystyvät lisäämään sovellukseen uusia käyttäjiä. Kun ensimmäinen opettaja-käyttäjä on luotu, kirjaututaan ulos ja muutetaan web.config-tiedoston forms-kohdan määritys login="EkaKirjautuminen.aspx" takaisin määrittämiseksi login="Kirjaudu.aspx", tai jos käytetään osittaista varmennesuojaa login="sslSuojattu/Kirjaudu.aspx". Tämän jälkeen poistetaan EkaKirjautuminen.aspx-tiedosto palvelimelta. EkaKirjautuminen-sivu on luotu ainoastaan sovelluksen ensimmäisten käyttäjien luomiseksi. On tietoturvan kannalta erittäin tärkeää ettei sivu jää palvelimelle.

6 POHDINTA

Toimeksianto vaikutti aluksi selkeältä järjestelmältä, joka olisi helppo toteuttaa. Ohjelman suunnittelun ja hahmottamisen edistyessä nousi kuitenkin useita ongelmia vastaan. Suurimmat ongelmat koskivat tietoturvaa, kirjautumista ja järjestelmän tulevaa ylläpitoa. Jotta sovelluksesta saatiin edes osittain tietoturvallinen, täytyi selvittää, mitkä asiat vahvistavat ja mitkä heikentävät ASP.NET-pohjaisten sivujen tietoturvallisuutta. Koska järjestelmästä haluttiin ylläpidon kannalta mahdollisimman vähän resursseja vievä, järjestelmään luotiin opettajakäyttäjille osittaiset ylläpidon omaavat oikeudet. Näin käyttäjien lisääminen, muokkaaminen ja salasanan uusiminen ei veisi ylläpidolta resursseja. Tämä kuitenkin esti mahdollisten valmiiden kirjautumismoduulien käytön. Vaikka opettajakäyttäjät omaavat ylläpidon oikeuksia, tämä ei tarkoita sitä, ettei järjestelmä tulisi jonkinlaista ylläpitäjää tarvitsemaan; varsinkin järjestelmän tietokanta tulisi varmuuskopioida päivittäin. Ylläpitäjää tarvitaan kuitenkin viimeistään, kun varmuuskopio joudutaan palauttamaan.

Uskon että suurimpana ja ratkaisevimpana haasteena tulee tästä eteenpäin olemaan järjestelmän käyttöönottoaihe. Itse järjestelmän käyttöönoton ja toimivuuden testaamisen lisäksi käyttöönottoaihe vaatii panostusta, jotta käyttäjät tulevat varmasti käyttämään järjestelmää.

Jotta oppilaat ja opettajat käyttäisivät järjestelmää, täytyisi heille iskostaa, että järjestelmä vaatii toimiakseen että sitä käytetään. Esimerkiksi oppilasta ei voida merkitä seminaarissa läsnä olevaksi, jos hän ei ole ilmoittautunut kyseiseen seminaariin. Opettajilla on järjestelmään luotu osittaiset ylläpitäjän oikeudet, joten heille on myös asetettu osia ylläpitäjän vastuista. Tämän vuoksi järjestelmä vaatii erittäin paljon myös ohjaavalta opettajalta sitoutumista järjestelmän käyttöön. Vaikka merkinnät voidaan myös tehdä jälkikäteen järjestelmään, merkinnät unohtuvat usein helposti.

Järjestelmän käyttöä on pyritty helpottamaan pyrkimällä luomaan järjestelmästä mahdollisimman yksinkertainen ja luomalla jokaisen sivun ominaisuuksille, sekä käytölle ohje-ominaisuus. Lisäksi käyttäjätunnuksen ja salasanan muistamista

helpottaa mahdollisuus vaihtaa ne käyttäjän haluamiksi. Tämä vähentää kirjautumistunnuksen lisäämää kynnystä käyttää järjestelmää.

Järjestelmän käyttöönottovaiheessa täytyisi huomioida järjestelmän tietoturvallisuuden asettamat vaatimukset. Järjestelmään on luotu mahdollisuus olla käyttämättä varmenteita, käyttää vain salasanojen yhteydessä varmennetta tai suojata koko järjestelmä varmenteella. Varmenteen käyttämättä jättäminen luo todella suuren tietoturvariskin, kun taas kokosivun suojaaminen varmenteella saattaa hidastaa järjestelmän käyttöä vanhemmilla koneilla ja hitaammilla Internet-yhteyksillä. Käyttöönottovaiheessa täytyy miettiä, mitä varmennetasoa on järkevintä käyttää.

Vaikka järjestelmä tulee vaatimaan panostusta opettajilta, uskon että järjestelmästä tulee olemaan heille suurta hyötyä. Opettajat pystyvät seuraamaan yksittäisen opiskelijan opinnäytteen edistymistä, ilman että häneen tarvitsee olla jatkuvassa yhteydessä. Oppilaille suurin hyöty tulee olemaan, nykyiseen paperiseen versioon nähden, seminaaritilaisuuksien näkeminen, sekä se, ettei suoritusten kuittaamiseen vaadittavaa paperia tarvitse kantaa mukana.

LÄHTEET

Acheson S.; Carasik A. 2001. The Secure Shell Frequently Asked Questions. Viitattu 12.10.2010. Saatavissa: <http://www.onsight.com/faq/ssh/ssh-faq.html>

Barrett D.; Silverman R. 2001. Sebastopol: O'Reilly.

Brainard, J.; Juels A.; Rivest R.; Szydlo, M.; Yung M. 2006. Fourth-Factor Authentication: Somebody You Know. Viitattu 12.10.2009. Saatavissa: <http://www.rsa.com/rsalabs/staff/bios/ajuels/publications/fourth-factor/ccs084-juels.pdf>

Hakala M.; Vainio M.; Vuorinen O. 2006. Tietoturvallisuuden käsikirja. Helsinki: WSOY.

Järvinen, P. 2003. Salausmenetelmät. Helsinki: WSOY.

Moghadampour G. 2006 C#-ohjelmointi. Helsinki: WSOY.

Ruohonen, M. 2002. Tietoturva. Helsinki: WSOY.

Rämö H. 2008. ASP.NET ja selainpohjaisen sovelluksen suunnittelu. Opinnäytetyö. Tietojenkäsittelyn koulutusohjelma. Turku: Turun ammattikorkeakoulu.

Stallings, W. 2005. Cryptography and Network Security Principles and Practices. 4., uudistettu painos. New Jersey: Prentice Hall.

Troelsen, A. 2007. Pro C# 2008 and the .NET 3.5 Platform. 4., uudistettu painos. New York: Springer-Verlag.

Opinnäytetyön liite 1 (AMK)

Tietojenkäsittely

Tietojärjestelmät

2010

Sami Timonen

OPINNÄYTETYÖSKENTELYN SEURANTAJÄRJESTELMÄ

Määrittelydokumentti



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

SISÄLTÖ

1 JOHDANTO	4
1.1 Tarkoitus ja Tuote	4
1.2 Määritelmät, termit ja lyhenteet	4
1.3 Tähän liittyvät dokumentit	4
1.4 Yleiskatsaus	4
2 YLEISKUVAUS	5
2.1 Ympäristö	5
2.2 Toiminta	5
2.3 Käyttäjät ja roolit	8
2.4 Yleiset rajoitteet	8
2.5 Oletukset ja riippuvuudet	8
3 TIEDOT JA TIETOKANTA	8
4 TOIMINNOT	10
5 ULKOISET LIITTYMÄT	11
5.1 Käyttöliittymä	11
5.2 Laitteistoliittymät	11
5.3 Ohjelmistoliittymät	11
5.4 Tietoliikenneliittymät	11
6 MUUT OMINAISUUDET	12
6.1 Suorituskyky	12
6.2 Käytettävyys, turvallisuus ja suojaukset	12
6.3 Ylläpidettävyys	12
6.4 Siirrettävyys ja yhteensopivuus	13
7 SUUNNITTELURAJOITTEET	13
7.1 Laitteistorajoitteet	13
7.2 Ohjelmistorajoitteet	13

KUVIOT

Kuvio 1. Liittymäkaavio	5
Kuvio 2. Opettajakäyttäjän tapahtumat	6
Kuvio 3. Oppilaskäyttäjän tapahtumat	7
Kuvio 4. Opinnäytetyöskentelyn seurantajärjestelmän luokkakaavio.	10

1 Johdanto

1.1 Tarkoitus ja Tuote

Tämä dokumentti on opinnäytteenä luodun opinnäytetyöskentelyn seurantajärjestelmän määrittelydokumentti. Määrittelydokumentti on luotu Hanna Rämön opinnäytteen ASP.NET ja selainpohjaisen sovelluksen suunnittelu liitteenä olevan määrittelydokumentin pohjalta.

Tarkoituksena oli luoda opinnäytetyöskentelynseuraamista avustava sovellus. Sovelluksella helpotetaan opinnäytetyön ohjaajan ja tekijän välistä yhteistyötä, ylläpitämällä opinnäyteprosessissa mukana olevia tietoja ja mahdollistamalla erilaisten raporttien tulostus näistä tiedoista. Sovellus toimii Internetin yli selainpohjaisesti.

Sovellus tehtiin Turun ammattikorkeakoulun tietojenkäsittelyn koulutusohjelmalle. Sen käyttäjinä tulevat toimimaan tietojenkäsittelyn opiskelijat ja opettajat.

1.2 Määritelmät, termit ja lyhenteet

Tämä dokumentti on opinnäytteenä luodun opinnäytetyöskentelyn

1.3 Tähän liittyvät dokumentit

Tämä dokumentti on liite opinnäytteeseen Selainpohjaisen sovelluksen toteuttaminen .NET ympäristöllä. Tätä dokumenttia seuraa liitteet suunnitteludokumentti ja testausdokumentti.

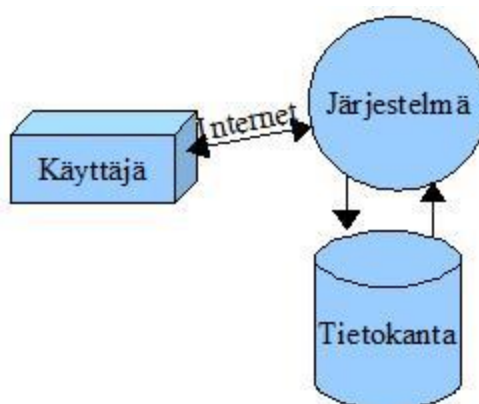
1.4 Yleiskatsaus

Tässä dokumentissa käydään läpi järjestelmän toimintavaatimukset. Eli se pyrkii vastaamaan siihen mitä järjestelmän tekee ja mitä se sisältää.

2 Yleiskuvaus

2.1 Ympäristö

Kuviossa 1 on esitetty opinnäytetyöskentelyn seuranta järjestelmän liittymäkaavio. Käyttäjä hakee, syöttää ja muokkaa tietoja sovelluksen kautta tietokantaan.



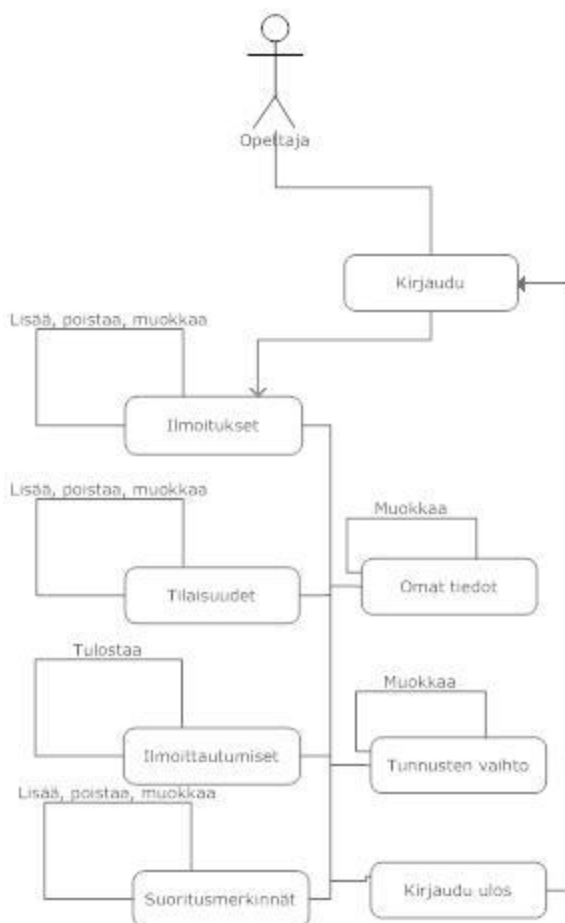
Kuvio 5. Liittymäkaavio

2.2 Toiminta

Sovelluksen käyttäjät ryhmitellään opettajiin ja oppilaisiin. Tapahtumalista:

1. Opettajakäyttäjä tarkastelee, lisää, muokkaa ja poistaa käyttäjiä
2. Opettajakäyttäjä tarkastelee, tulostaa, lisää, muokkaa ja poistaa tilaisuuksia
3. Opettajakäyttäjä tarkastelee, tulostaa, merkitsee ja poistaa suoritusmerkintöjä
4. Opettajakäyttäjä tarkastelee ja tulostaa ilmoittautumisia
5. Opettajakäyttäjä lisää, muokkaa ja poistaa ilmoituksia

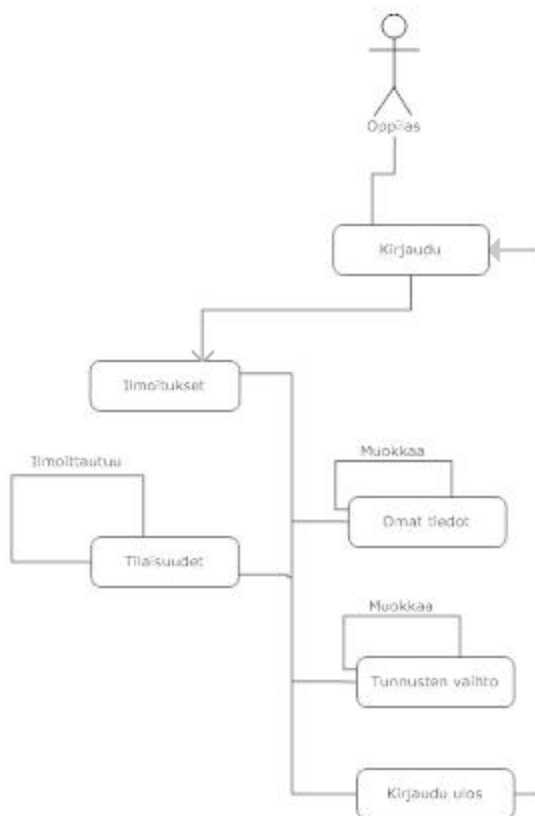
6. Oppilaskäyttäjä tarkastelee tilaisuuksia, sekä ilmoittautuu niihin
7. Käyttäjä tarkastelee ilmoituksia
8. Käyttäjä tarkastelee ja muokkaa omia tietoja
9. Käyttäjä muokkaa omaa käyttäjätunnusta tai salasanaansa



Kuvio 6. Opettajakäyttäjän tapahtumat

Kuviossa 2 kuvataan opettajaroolin omaavan käyttäjää koskevat tapahtumat. Opettaja kirjautuu aluksi järjestelmään, jonka jälkeen näytetään ilmoitukset. Ilmoituksia pystyy lisäämään, poistamaan ja muokkaamaan. Opettaja voi halutessaan katsoa tilaisuuksia, ilmoittautumisia, suoritusmerkintöjä ja omia tietojaan. Siirryttyään katsomaan tilaisuuksia, voidaan niitä lisätä, poistaa ja muokata. Siirryttyään katsomaan ilmoittautumisia, voidaan ne tulostaa.

Siirryttyään katsomaan suoritusmerkintöjä, voidaan niitä lisätä, poistaa ja muokata. Siirryttyä katsomaan omia tietoja voidaan niitä muokata. Siirryttäessä tunnusten vaihtoon voidaan omia käyttäjätunnuksia vaihtaa. Kirjaututtaessa ulos, ohjataan takaisin kirjautumiseen.



Kuvio 7. Oppilaskäyttäjän tapahtumat

Kuviossa 3 kuvataan oppilasroolin omaavan käyttäjää koskevat tapahtumat. Oppilas kirjautuu aluksi järjestelmään, jonka jälkeen näytetään ilmoitukset. Oppilas voi halutessaan katsoa tilaisuuksia ja omia tietojaan, sekä vaihtaa käyttäjätunnuksiaan. Siirryttyään katsomaan tilaisuuksia, voidaan niihin ilmoittautua. Siirryttyä katsomaan omia tietoja voidaan niitä muokata. Siirryttäessä tunnusten vaihtoon voidaan omia käyttäjätunnuksia vaihtaa. Kirjaututtaessa ulos, ohjataan takaisin kirjautumiseen.

2.3 Käyttäjät ja roolit

Järjestelmää tulee käyttämään opinnäyteprosessissa mukana olevat opiskelijat ja ohjaavat opettajat. Molemmat käyttäjäryhmät voivat nähdä ja muokata omia tietojaan, sekä nähdä lisätyt ilmoitukset ja tilaisuudet.

Opiskelijat voivat ilmoittautua esiintyjöiksi tai kuuntelijoiksi tilaisuuksiin.

Opettajat voivat tarkastella kaikkien opiskelijoiden tietoja ja eri hakutoimintojen avulla tulostaa tietoja järjestelmästä. Lisäksi opettajat voivat lisätä, muokata tai poistaa tilaisuuksia ja käyttäjiä, sekä merkitä oppilaiden suorituksia.

2.4 Yleiset rajoitteet

Järjestelmä tulee sijaitsemaan koulun palvelimella. Sovellusta tullaan käyttämään Internetin välityksellä. Jokaisella käyttäjällä on henkilökohtainen käyttäjä tunnus ja salasana järjestelmään.

2.5 Oletukset ja riippuvuudet

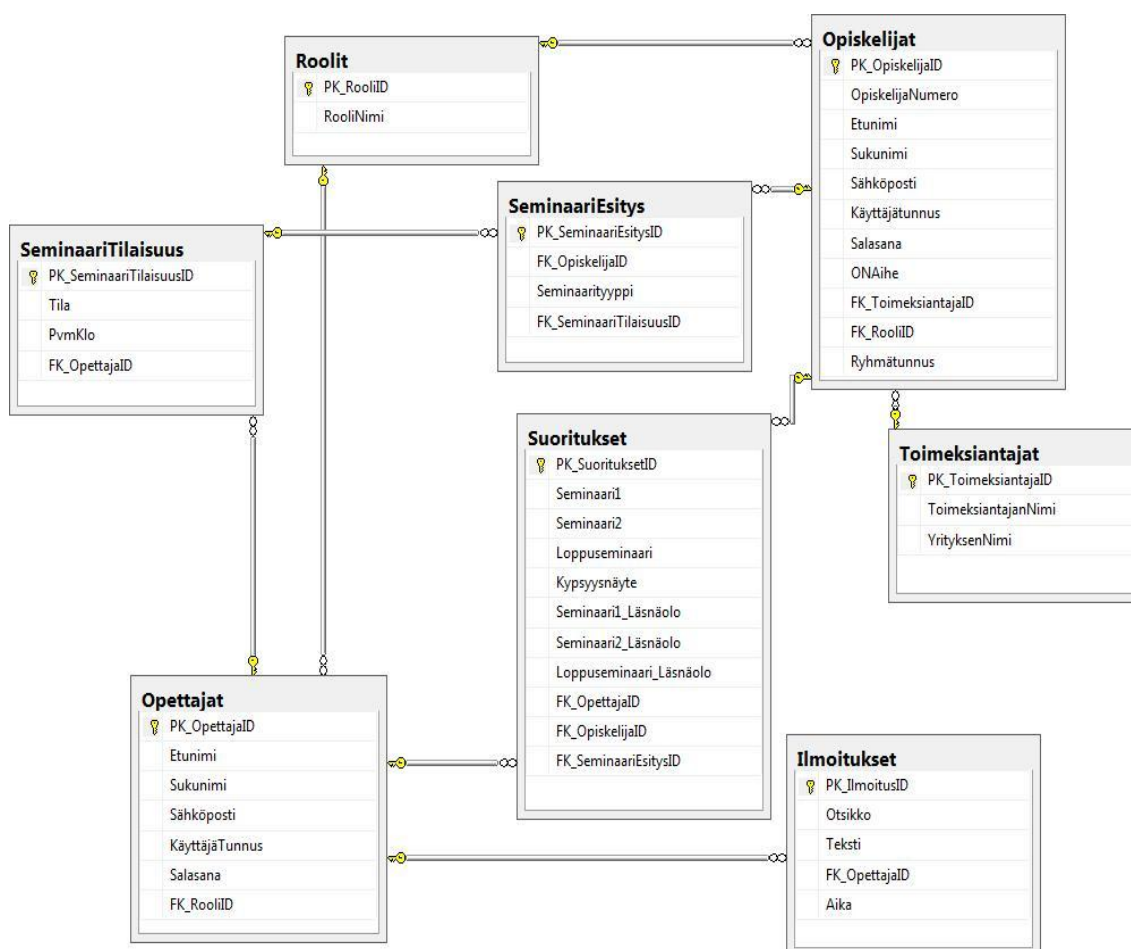
Palvelimella jolla sovellus sijaitsee täytyy olla yhteys tietokantaan. .NET Framework 3.5 tulee olla asennettuna. Sovellusta käytetään Internet-selaimen kautta.

3 Tiedot ja tietokanta

Järjestelmän tietokanta on luotu Microsoft SQL Server 2005:lla. Tietokanta ja sen relaatiot on kuvattu kuviossa 2. Tietokanta sisältää seuraavat taulut:

- Opiskelijat: Sisältää tiedot opiskelijasta. Yhteydessä Roolit- ja Toimeksiantajat-tauluun.
- Opettajat: Sisältää tiedot opettajasta. Yhteydessä Roolit- ja Toimeksiantajat-tauluun.

- Roolit: Taulun avulla tunnistetaan käyttäjän rooli kirjautumisen yhteydessä.
- Toimeksiantajat: Sisältää tiedot opiskelijan opinnäytteen toimeksiantajasta.
- SeminaariTilaisuus: Sisältää tiedot tilaisuuden ajankohdasta ja paikasta
- SeminaariEsitys: Sisältää tiedot tilaisuuteen ilmoittautuneen opiskelijan seminaarityypistä. Yhteydessä Opettajat- ja SeminaariTilaisuus-tauluihin.
- Suoritukset: Sisältää opiskelijalle tilaisuuteen merkityn suorituksen. Yhteydessä Opiskelijat-tauluun ja Opettajat-tauluun, jotta saataisiin selville merkinnän tehnyt opettaja.
- Ilmoitukset: Sisältää ilmoituksessa näkyvät tiedot



Kuvio 8. Opinnäytetyöskentelynseurantajärjestelmän luokkakaavio.

4 Toiminnot

Järjestelmällä on kaksi käyttäjäryhmää. Näiden perusteella toiminnot ovat joko ainoastaan opettajien, oppilaiden tai näiden molempien käytössä. Molemmat käyttäjäryhmät näkevät ja voivat muokata osaa omista tiedoistaan.

Ilmoitukset ovat molempien ryhmien nähtävissä, mutta ainoastaan opettajat voivat lisätä ilmoituksen. Lisäksi opettaja voi muokata tai poistaa lisäämänsä ilmoituksen.

Tilaisuuden tiedot ja tilaisuuteen ilmoittautuneet esiintyjät ovat näkyvissä molemmilla ryhmillä. Tiedot saadaan esille kalenterin avulla. Kalenteriin on merkitty päiväykset, joina on tilaisuus. Opettajat voivat lisätä, muokata tai

poistaa tilaisuuksia. Oppilaat voivat ilmoittautua lisättyihin tilaisuuksiin esiintyjäksi tai kuuntelijaksi. Kun tilaisuus on pidetty opettajat merkitsevät tilaisuuden ilmoittautumiset suoritetuiksi oppilaskohtaisesti.

Opettajat pystyvät tarkastelemaan ja tulostamaan ilmoittautuneita esiintyjä ja kuuntelijoita, sekä tilaisuuden merkattuja suorituksia. Opettajat voivat etsiä yksittäinen opiskelija ja tarkastella tämän suoritusmerkintöjä. Lisäksi opettajat voivat tarkastella kaikkien oppilaiden tietoja ja kaikkia toimeksiantajaryityksiä. Opettaja voi halutessaan tulostaa listan raportiksi.

Opettajat voivat lisätä, muokata ja poistaa käyttäjiä, sekä lähettää salasanan unohtuessa käyttäjälle uusi salasana käyttäjän sähköpostiin. Molemmat ryhmät voivat vaihtaa omaa käyttäjätunnusta ja salasanaansa.

5 Ulkoiset liittymät

5.1 Käyttöliittymä

Käyttöliittymä on selainpohjainen. Käyttäjä ja rooli tunnustetaan järjestelmään käyttäjätunnuksen ja salasanan perusteella. Käyttäjän rooli määrittelee näytettävät toiminnot.

5.2 Laitteistoliittymät

Sovellus vaatii tietokoneen jossa on Internet-yhteyden.

5.3 Ohjelmistoliittymät

Sovelluksen käyttö vaatii Internet-selaimen.

5.4 Tietoliikenneliittymät

Käyttäjällä täytyy olla Internet-yhteys. Järjestelmän palvelimella täytyy olla yhteys tietokantaan.

6 Muut ominaisuudet

6.1 Suorituskyky

Usean käyttäjän samanaikaisella käytöllä ei ole vaikutusta sovelluksen suorituskykyyn.

6.2 Käytettävyys, turvallisuus ja suojaukset

Sovelluksen käyttöä on helpotettu ohjeiden avulla. Turvallisuus on otettu huomioon rajoittamalla käyttäjien pääsyä sovellukseen, vaatimalla kaikilta käyttäjiltä kirjautumista. Lisäksi salasanat tallennetaan tietokantaan hash-muodossa. Näin salasanaa ei saada selville vaikka se luettaisiin tietokannasta. Toisin kuin kryptauksessa, hash-funktiolla muutettua salasanaa ei voida muuttaa takaisin alkuperäiseen muotoon.

Käyttäjälle sovelluksen käytönajaksi luotu eväste, luodaan istuntokohtaiseksi, jolloin eväste tallentuu ainoastaan tietokoneen keskusmuistiin. Käyttöönottoa varten luodaan optiot varmenteiden käyttämiseksi joko koko sovelluksessa, tai sivukohtaisesti. Tällöin suojataan ainoastaan ne kohdat joissa luetaan, luodaan tai muokataan sovelluksen salasanoja.

6.3 Ylläpidettävyys

Ohjelmakoodia on kommentoitu, niiden toimintojen osalta jotka sitä vaativat. Tietokantataulut ja niiden väliset relaatiot on pyritty rakentamaan mahdollisimman selkeiksi ja yksinkertaisiksi.

Opettajilla on mahdollisuus generoida käyttäjälle uusi salasana, joka lähetetään salasanan omistajalle sähköpostilla. Sovelluksen generoima salasana ei tule sen luoneen opettajan tietoon.

6.4 Siirrettävyys ja yhteensopivuus

Sovellusta voidaan käyttää millä tahansa Internet-yhteyden omaavalla tietokoneella.

Tietokanta voidaan sijoittaa eri palvelimelle kuin sovellus, mutta sovelluksella täytyy olla yhteys tietokantaan.

7 Suunnittelurajoitteet

7.1 Laitteistorajoitteet

Kehitystyössä käytettävistä sovelluksista Microsoft Visual Studio 2008 luo suurimmat laitteistorajoitteet. Sen toimintavaatimuksina ovat vähintään seuraavat ominaisuudet:

- 1,6 Ghz prosessori
- 384 megatavua keskusmuistia. Jos käytössä on Windows Vista tai uudempi käyttöjärjestelmä vaatimuksena on 768 megatavua keskusmuistia
- 1024 x 768 resoluution näyttö

7.2 Ohjelmistorajoitteet

Sovelluksen kehitystä varten asennettiin Microsoft Visual Studio 2008 ohjelmankehitysympäristö. Lisäksi asennettiin Microsoft .NET Framework 3.5, IIS-palvelinohjelmisto ja Microsoft SQL server 2005. Visual Studio 2008 toimii Microsoftin Windows XP sp2, Windows Server 2003, Windows Vista ja näitä uudemmissa Microsoftin käyttöjärjestelmissä.

Opinnäytetyön liite 2 (AMK)

Tietojenkäsittely

Tietojärjestelmät

2010

Sami Timonen

OPINNÄYTETYÖSKENTELYN SEURANTAJÄRJESTELMÄ

Suunnitteludokumentti



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

SISÄLTÖ

1 JOHDANTO	4
1.1 Tarkoitus ja kattavuus	4
1.2 Tähän liittyvät dokumentit	4
1.3 Yleiskatsaus	4
2 JÄRJESTELMÄN YLEISKUVAUS	4
2.1 Ympäristö	4
2.2 Toiminta	5
2.3 Käyttäjät	5
2.4 Yleiset rajoitteet	5
2.5 Oletukset ja riippuvuudet	5
3 ARKKITEHTUURIN KUVAUS	6
3.1 Ratkaisuperiaatteet	6
3.2 Tietokanta-arkkitehtuuri	6
4 MODUULIKUVAUKSET	7
4.1 Kirjautuminen	7
4.2 Ilmoitukset	8
4.3 Omat tiedot	9
4.4 Kalenteri	11
4.5 Käyttäjienhallinta	13
4.6 Raporttien tulostaminen	16
4.7 Tunnusten vaihto	18

KUVIOT

Kuvio 1 Kirjautumisen tapahtumakaavio	8
Kuvio 2 Ilmoitukset -sivun tapahtumakaavio	9
Kuvio 3 Omat tiedot -sivun tapahtumakaavio	10
Kuvio 4 Kalenteri -sivun tapahtumakaavio	12
Kuvio 5 Käyttäjienhallinta -sivun tapahtumakaavio	15
Kuvio 6 Raporttien tulostaminen -sivun tapahtumat	17
Kuvio 7 Tunnusten vaihto -sivun tapahtumakaavio	19

1 Johdanto

Tarkoitus ja kattavuus

Tämä dokumentti on opinnäytteenä luodun opinnäytetyöskentelyn seurantajärjestelmän suunnitteludokumentti. Se on luotu Hanna Rämön opinnäytteen ASP.NET ja selainpohjaisen sovelluksen suunnittelu liitteenä olevan suunnitteludokumentin pohjalta. Dokumentti tarkentaa opinnäytetyöskentelynseuraamista avustavan sovelluksen määrittelydokumentin kuvausta. Dokumentissa kuvataan tekninen toteutus määrittelydokumentin toiminnallisista määrittelyistä. Se sisältää sovelluksen ratkaisuperiaatteet ja kuvauksen sovelluksen toiminnasta.

Tähän liittyvät dokumentit

Tämä dokumentti on liite opinnäytteeseen Selainpohjaisen sovelluksen toteuttaminen .NET-ympäristöllä. Tätä dokumenttia edeltää liite määrittelydokumentti ja seuraa liite testausdokumentti.

Yleiskatsaus

Tässä dokumentissa käydään läpi sovelluksen tekninen toteutus. Eli dokumentti pyrkii vastaamaan siihen miten sovellus on luotu.

2 Järjestelmän yleiskuvaus

2.1 Ympäristö

Sovellus toimii rajapintana käyttäjän ja tietokannan välillä. Tietokannassa ylläpidetään opinnäytetyöntekijöiden ja ohjaavien opettajien tietoja, seminaaritulaisuuksien ja näiden esiintyjien tietoja ja ohjaavien opettajien esille

laittamia ilmoituksia. Sovellus on web-sovellus jonka käyttö vaatii Internet-yhteyden.

2.2 Toiminta

Toiminnot jakautuvat käyttäjienhallintaan, ilmoitusten tarkasteluun ja hallintaan, käyttäjän tietojen ylläpitoon, ilmoitusten tietojen tulostukseen ja seminaarien hallintaan, niihin ilmoittautumisiin ja läsnäolojen merkintään. Toiminnot ovat tarkemmin kuvattuna määrittelydokumentin neljännessä luvussa.

2.3 Käyttäjät

Järjestelmän käyttäjiä ovat opinnäytetyöprosessin opiskelijat ja näiden ohjaavat opettajat. Opettajilla on oikeus hallita käyttäjiä. Määrittelydokumentin luvussa 2.3 on tarkemmin kuvattuna käyttäjät, niiden roolit ja roolien käyttöoikeudet.

2.4 Yleiset rajoitteet

Sovellusta käytetään Internetin välityksellä. Sen käyttö vaatii kirjautumista järjestelmään, joten käyttäjä täytyy lisätä järjestelmään. Jotta käyttäjälle voidaan lähettää kirjautumiseen vaadittava salasana ja käyttäjätunnus täytyy hänellä olla sähköpostiosoite.

2.5 Oletukset ja riippuvuudet

Sovelluksen käyttö vaatii Internet-selaimen. Sovellus on luotu ASP.NET-tekniikalla selainriippumattomaksi. Tietokanta voidaan sijoittaa eri palvelimella kuin sovellus. Sovellus vaatii kuitenkin yhteyden tietokantaan.

3 Arkkitehtuurin kuvaus

3.1 Ratkaisuperiaatteet

Järjestelmä on suunniteltu ja luotu tietojenkäsittelyn koulutusohjelmaa varten. Järjestelmä on kuitenkin pyritty toteuttamaan niin, että järjestelmä voidaan laajentaa myös muiden koulutusohjelmien käyttöön.

Selainpohjaisuus mahdollistaa aikaan ja paikkaan sitoutumattoman pääsyn järjestelmään, sekä opinnäytetyöskentelyn seuraamisen reaaliaikaisesti. Sovelluksen luontiin valittiin Microsoftin ASP.NET. Se mahdollistaa Internet-pohjaisten, tietoturvallisten, käyttäjän kannalta alustariippumattomien ja toiminnoiltaan monipuolisten sovellusten luonnin. Ohjelmointikieleksi valittiin C#. Se on oliopohjainen ja sisältää usean muun ohjelmointikielen parhaimmat ominaisuudet, joten se on tällä hetkellä kehittynein .NET-arkkitehtuurin tarjoamista ohjelmointikielistä.

Opettajille on mahdollistettu käyttäjienhallinta, sekä muiden käyttäjien salasanan uusiminen. Tämä vähentää järjestelmän ylläpidon tarvetta, sekä helpottaa käyttäjien pääsyä järjestelmään. Käyttäjä voi muuttaa omaa käyttäjä tunnusta ja salasanaansa. Tämä parantaa käytettävyyttä, koska käyttäjä voi valikoida käyttäjätunnuksen ja salasanan, jotka hänen on helppo muistaa.

3.2 Tietokanta-arkkitehtuuri

Sovelluksen tietokanta on luotu Microsoft SQL Server 2005:lla. Tietokantataulut ja niiden väliset relaatiot ovat kuvattuna määrittelydokumentin kolmannessa luvussa. Tietokannantauluissa käytetyt identifiointisarakeet on määritelty uniqueidentifier-tietotyyppiä. Generoimalla tunniste automaattisesti varmistetaan että kaikki luodut tietokannantaulun rivit saavat eri tunnisteiden. Uniqueidentifier on 32 merkin heksadesimaali merkkijono, joten käytännössä on lähes mahdotonta että sama tietokone loisi kaksi samaa tunnistetta.

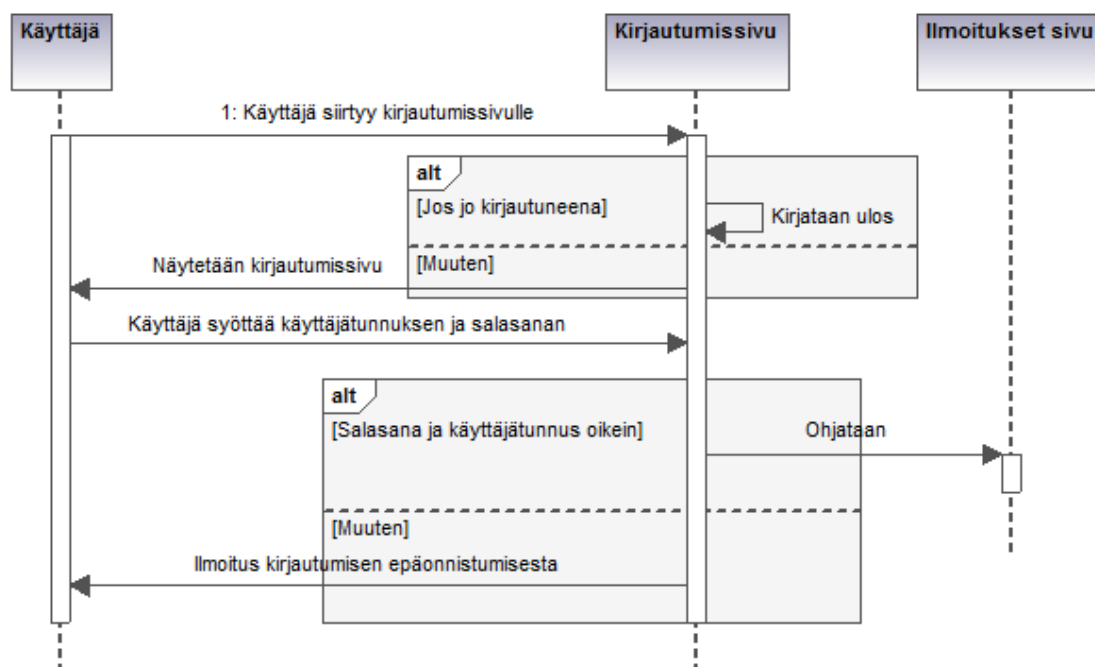
4 Moduulikuvaukset

4.1 Kirjautuminen

Kirjautuminen järjestelmään on luotu ASP.NET-tekniikan tarjoamalla Forms-tunnistautumisella. Sovellukseen on rakennettu oma kirjautumissivu, jonka kautta käyttäjän syöttämää käyttäjätunnusta ja salasanaa verrataan tietokannassa olevien Opettaja- ja Oppilas-taulun riveihin. Koska salasana on tallennettuna tietokantaan hash-muotoon, täytyy syötetty salasana muuttua ennen vertausta hash-funktiolla.

Jos syötetyn käyttäjätunnuksen ja salasanan perusteella löytyy käyttäjä, palautetaan tämän käyttäjän rooli, joka tallennetaan sovelluksen käyttöoikeuden omaavaan evästeeseen. Roolin perusteella käyttäjälle mahdollistetaan tai estetään sovelluksen toimintoja. Lisäksi evästeeseen tallennetaan tietokannasta haettu käyttäjän tunniste. Tällä tunnisteella käyttäjä tunnistetaan järjestelmässä. Eväste luodaan käyttäjälle istuntokohtaiseksi, joten eväste tallentuu ainoastaan käyttäjän tietokoneen keskusmuistiin kovalevyn sijasta. Tietoturvan lisäämiseksi käyttäjän evästeen sisältö kryptataan.

Käyttäjän yrittäessä sovellukseen ilman sovelluksen käytön omaavaa tunniste-evästettä käyttäjä ohjautuu kirjautumissivulle. Kun käyttäjä kirjautuu ulos sovelluksesta, ohjataan hänet kirjautumissivulle. Sivulla käyttäjä kirjataan ulos sovelluksesta ja käyttäjän istunto lopetetaan.



Kuvio 9 Kirjautumisen tapahtumakaavio

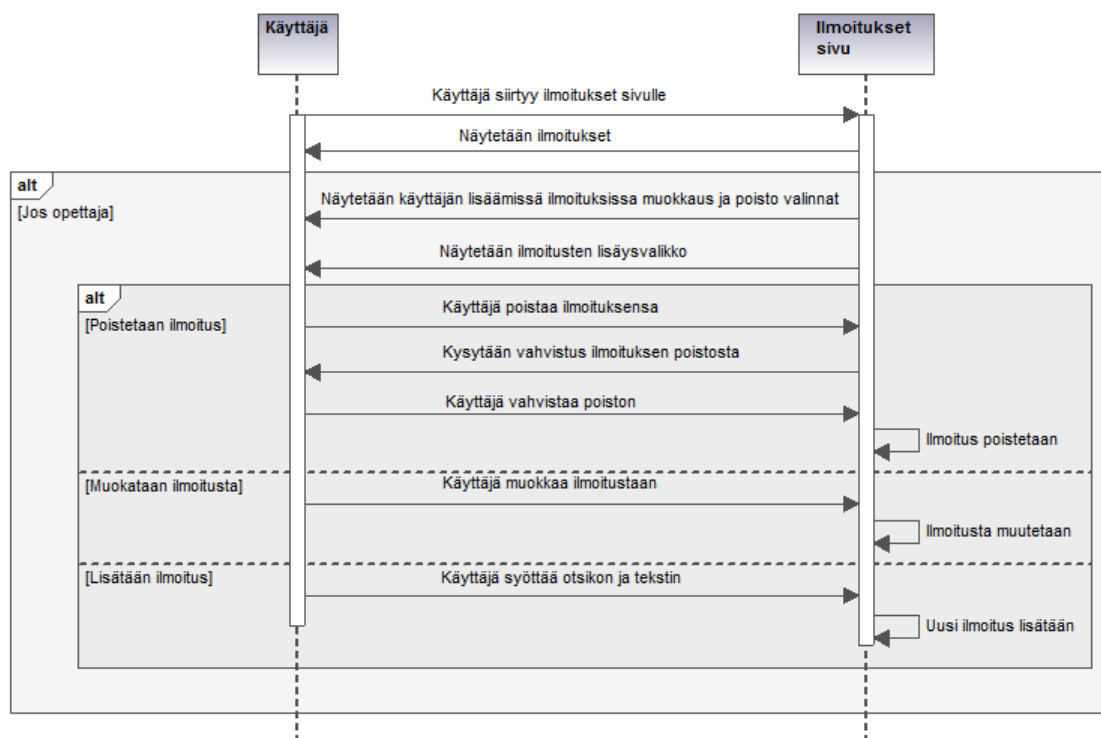
Kuviossa 1 esitetään käyttäjän kirjautumista kuvaavat tapahtumat. Ensin käyttäjä siirtyy kirjautumissivulle. Jos käyttäjä on jo kirjautuneena sisälle, kirjataan hänet ulos järjestelmästä. Kun käyttäjä on syöttänyt salasanan ja käyttäjätunnuksensa, tarkistetaan ne. Jos tunnukset olivat oikeita, ohjataan käyttäjä ilmoitukset sivulle, muuten annetaan ilmoitus kirjautumisen epäonnistumisesta.

4.2 Ilmoitukset

Ilmoitukset -sivu on asetettu sovelluksen oletussivuksi. Näin opettajien jättämät ilmoitukset ovat heti nähtävissä oppilaan kirjautuessa järjestelmään. Opettajilla on nähtävissä valikko, josta opettaja voi lisätä haluamansa ilmoitustekstin, sekä ilmoituksen otsikon. Tietokantaan tallennetaan tekstin ja otsikon lisäksi, haettu lisääjän tunniste, sekä tallennushetken aika ja päiväys. Tietokannasta haetaan ja näytetään sivulla otsikon ja tekstin lisäksi tunnisteiden avulla haettu lisääjän nimi, sekä lisäyksen aika ja päivämäärä.

Jokaisella ilmoitusta näytettäväksi luotaessa kerralla tarkistetaan onko käyttäjän tunniste sama kuin ilmoituksen lisääjän tunniste. Jos on, näytetään kyseisessä

ilmoituksessa myös mahdollisuus muokata tai poistaa ilmoitus. Kun ilmoitusta muokataan, korvataan sen aika ja päivämäärä, muokkaushetken ajalla ja päivämäärä. Poistettaessa ilmoitusta kysytään varmistus ilmoituksen poistoon.



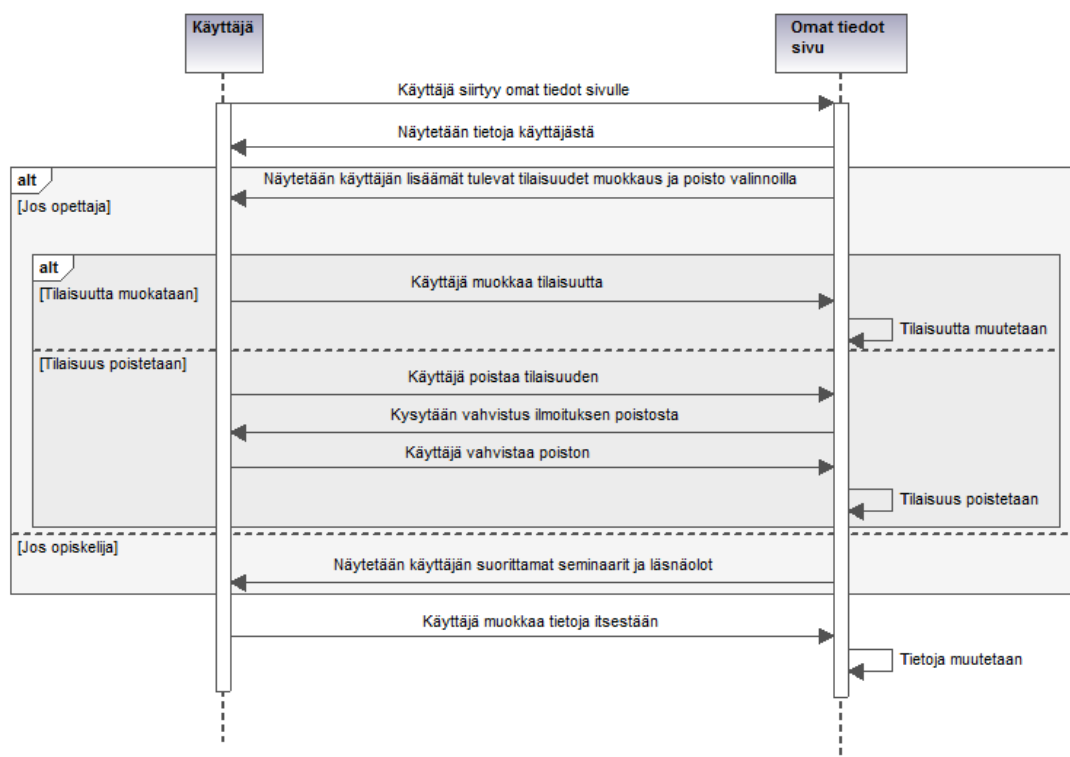
Kuvio 10 Ilmoitukset -sivun tapahtumakaavio

Kuviossa 2 esitetään Ilmoitukset sivua koskevat tapahtumat. Käyttäjän siirryttyä ilmoitukset sivulle tarkistetaan onko käyttäjän rooli opettaja. Jos käyttäjä on opettaja, niin näytetään ilmoitusten lisäysvalikko, sekä käyttäjän lisäämissä ilmoituksissa muokkaus ja poistovalinnat. Opettajakäyttäjän poistaessa ilmoitusta kysytään vahvistus ilmoituksen poistolle.

4.3 Omat tiedot

Sovelluksen omat tiedot sivulle on molemmilla, opettajilla ja oppilailta pääsy. Sivulta käyttäjällä on nähtävissä käyttäjää itseään koskevat tiedot. Tämä tarkoittaa opettajien kohdalla opettajan nimeä, sähköpostia ja tulevia ilmoittamiaan seminaareja. Opettaja voi vaihtaa omaa sähköpostiosoitettaan, poistaa ilmoittamiaan tulevia seminaarejaan ja muokata näiden seminaarien ajankohtaa ja paikkaa.

Oppilaat näkevät sivulla opiskelijanumeronsa, nimensä, sähköpostinsa, sekä opinnäytetyönsä aiheen, toimeksiantajan nimen ja toimeksiantajayrityksen. Lisäksi oppilas näkee omat suorittamansa seminaarit, sekä läsnäolojen määrä kussakin seminaarityypissä. Oppilas pystyy muokkaamaan omaa sähköpostiosoitettaan, opinnäytetyön nimeä, toimeksiantajan nimeä ja toimeksiantajayrityksen nimeä.



Kuvio 11 Omat tiedot -sivun tapahtumakaavio

Kuviossa 3 käyttäjän siirryttyä Omat tiedot -sivulle katsotaan onko käyttäjän rooli opettaja. Jos on, näytetään kyseisen opettajan lisäämät tulevat tilaisuudet. Opettaja voi joko muokata tai poistaa tilaisuuksia. Poistettaessa tilaisuutta, kysytään varmistus poistolle. Käyttäjän ollessa oppilas näytetään suoritetut seminaarit ja läsnäolot seminaareissa. Molemmat käyttäjät voivat muokata tietoja itsestään.

4.4 Kalenteri

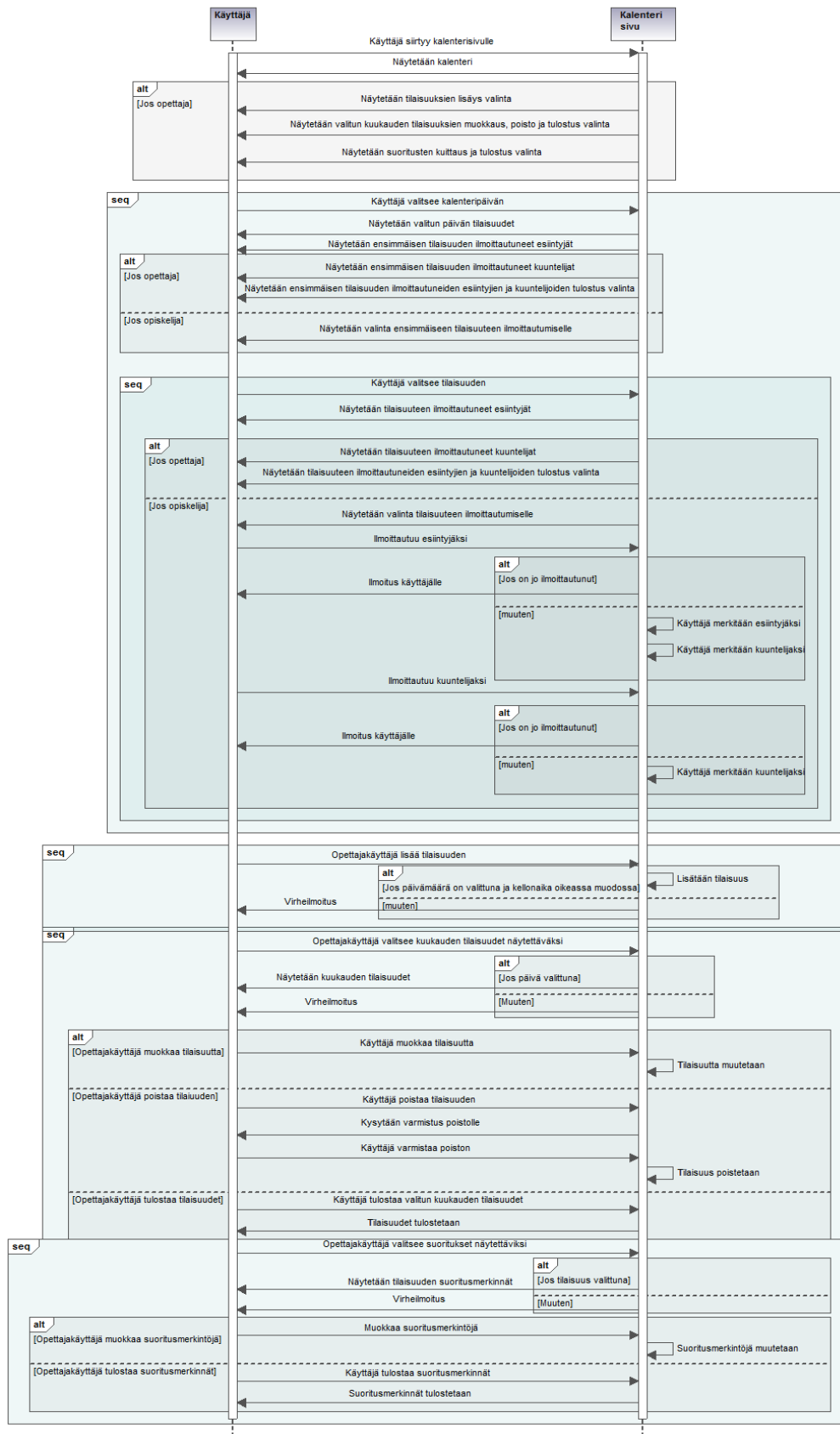
Kalenteri -sivu sisältää kalenteriobjektin, johon on merkattuna päivät jolloin on tilaisuuksia. Tilaisuus-taulusta haettuja päivämääriä verrataan kalenterin päivän luonnin yhteydessä. Samalla myös piilotetaan ne päivät jotka eivät ole valitun kuukauden päiviä. Kun käyttäjä valitsee kalenterista jonkin päivän, haetaan tietokannasta päivän perusteella tilaisuudet. Tilaisuuksista näytetään niiden aika, tila, lisänneen opettajan nimi ja tilaisuuden tyyppi. Listasta voidaan valikoida tilaisuus. Ensimmäinen tilaisuus on listan luonnin yhteydessä valittuna. Valitusta tilaisuudesta näytetään tilaisuuteen ilmoittautuneiden esiintyjien nimet, opinnäytetöiden nimet ja seminaarityypit. Opettajalle näytetään lisäksi kuuntelijaksi ilmoittautuneiden nimet.

Valittuaan tilaisuuden oppilas voi ilmoittautua tilaisuuteen. Jos tilaisuus on tyypiltään seminaari, täytyy oppilaan ensin valita osallistuuko tilaisuuteen kuuntelijaksi tai esiintyjäksi. Jos oppilas ilmoittautuu esiintyjäksi, täytyy valikoida vielä minkä seminaari aikoo pitää. Ennen ilmoittautumista varmistetaan seminaariin ilmoittautumisen hyväksyntä.

Opettaja pystyy lisäämään tilaisuuden kalenteriin. Ennen lisäämistä täytyy valikoida kalenterista päiväys, syöttää tilaisuuden kellonaika ja tila jossa tilaisuus pidetään. Tämän jälkeen valitaan onko tilaisuus tyypiltään seminaari vai kypsyysnäyte. Jos valikoidaan seminaari, täytyy valikoida lisätäänkö tilaisuus vain kyseiselle päivälle, vai koko kuukaudeksi valituille viikonpäiville. Tämän jälkeen kysytään vielä varmistus tilaisuuden lisäämiseksi.

Valittuaan kalenterista päivän, opettaja voi poistaa valitun kuukauden tilaisuuksia ja muokata tilaisuuksien ajankohtaa ja paikkaa.

Valittuaan tilaisuuden opettaja voi lisätä ilmoittautuneiden seminaarien pitäjille ja seminaareissa läsnäolijoille suoritusmerkinnät, tai poistaa halutut merkinnät. Ennen toteuttamista kysytään varmistus.



Kuvio 12 Kalenteri -sivun tapahtumakaavio

Kuviossa 4 on kuvattuna Kalenteri -sivua koskevat tapahtumat. Käyttäjän siirryttyä sivulle, tarkistetaan käyttäjän rooli. Käyttäjän ollessa opettaja mahdollistetaan valinnat tilaisuuksien lisäämiselle, tilaisuuksien muokkaamiselle ja poistamiselle, sekä suoritusmerkintöjen muokkaamiselle. Kun käyttäjä valitsee kalenterista päivän, näytetään kyseisen päivän tilaisuudet, sekä ensimmäisen tilaisuuden esiintyjät. Käyttäjän ollessa opettaja näytetään lisäksi ensimmäisen tilaisuuden kuuntelijat ja valinta tilaisuuden esiintyjien ja kuuntelijoiden tulostukseen.

Kun käyttäjä valitsee tilaisuuden, näytetään tilaisuuden esiintyjät. Käyttäjän ollessa opettaja, näytetään myös tilaisuuden esiintyjien ja kuuntelijoiden tulostus mahdollisuus. Käyttäjän ollessa opiskelija, näytetään mahdollisuus ilmoittautua tilaisuuteen joko esiintyjäksi tai kuuntelijaksi. Jos käyttäjä ilmoittautuu esiintyjäksi, merkitään hänet samalla tilaisuuteen myös kuuntelijaksi.

Opettajakäyttäjän valitessa kuukauden tilaisuudet näytettäväksi täytyy päivä olla valittuna kalenterista, jos ei ole annetaan siitä virheilmoitus. Tämän jälkeen voidaan tilaisuuksia muokata ja poistaa, sekä tulostaa ne. Poistettaessa tilaisuutta kysytään varmistus poistolle.

Opettajakäyttäjän valitessa suoritusmerkinnät näytettäväksi, täytyy tilaisuus johon merkinnät tehdään olla valittuna, jos ei ole annetaan siitä virheilmoitus. Tämän jälkeen käyttäjä voi lisätä tai poistaa ilmoittautuneiden opiskelijoiden suoritusmerkintöjä, sekä tulostaa ne.

4.5 Käyttäjienhallinta

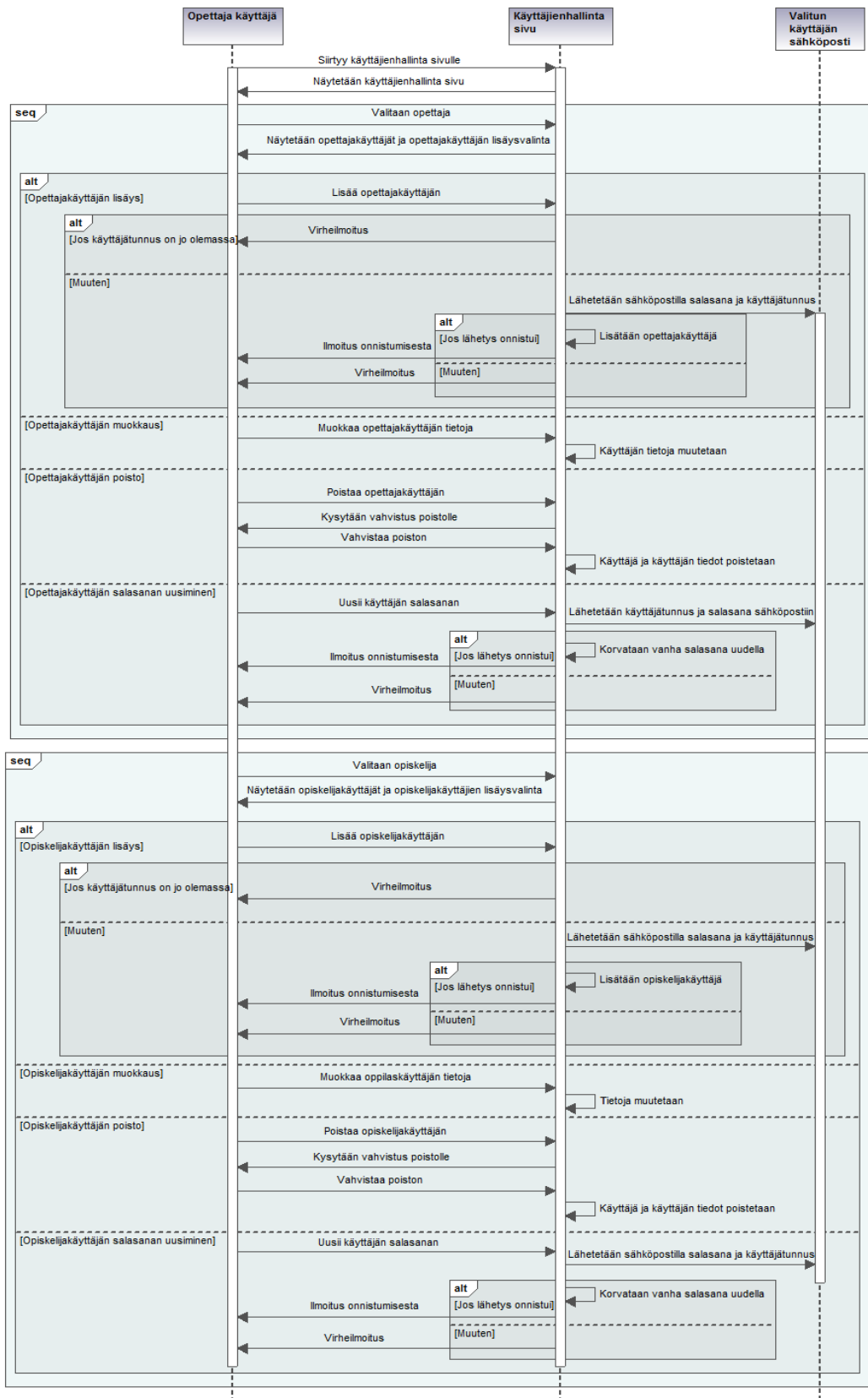
Käyttäjienhallinta on ainoastaan opettajien käytettävissä. Käyttäjienhallinnasta pystytään lisäämään käyttäjä järjestelmään, muokkaamaan käyttäjien tietoja, poistamaan käyttäjiä, sekä uusimaan käyttäjän salasanan.

Käyttäjää muokattaessa tai poistettaessa valikoidaan näytetäänkö lista opiskelijoista vai oppilaista. Lista sisältää valikot muokkaamiselle, poistamiselle ja salasanan uusimiselle.

Kun salasana uusitaan, generoidaan uusi salasana ja haetaan käyttäjän käyttäjätunnus ja lähetetään ne käyttäjän sähköpostiosoitteeseen. Uutta salasanaa ja käyttäjätunnusta ei näytetä sen uusijalle. Lopuksi näytetään ilmoitus salasanan onnistuneesta lähetyksestä.

Kun käyttäjä lisätään, valitaan ensin onko uusi käyttäjä oppilas vai opettaja. Tämän valinnan mukaan annetaan luotavalle käyttäjälle rooli. Ennen luontia täytyy syöttää lisättävän käyttäjän etunimi, sukunimi, tuleva käyttäjätunnus ja sähköpostiosoite. Opiskelijaa lisättäessä, tarvitsee lisäksi syöttää opiskelijan opiskelijanumero ja ryhmätunnus. Kun opiskelijanumeron on syöttänyt, ehdotetaan sitä myös käyttäjätunnukseksi.

Ennen kuin käyttäjä lisätään järjestelmään, tarkistetaan että tarvittavat tiedot on syötetty, käyttäjä tunnus on vähintään kuusi merkkiä pitkä ja sähköpostiosoite on kelvollisen muotoinen. Lisäksi käydään läpi onko käyttäjätunnus ja opiskelijanumero jo käytössä. Tämän jälkeen generoidaan käyttäjälle salasana, tallennetaan se hash-muodossa tietokantaan. Lisäksi lähetetään salasana ja käyttäjätunnus uuden käyttäjän sähköpostiosoitteeseen. Salasanaa ei näytetä käyttäjän lisääjälle. Lopuksi annetaan ilmoitus onnistuneesta käyttäjän lisääksestä.



Kuvio 13 Käyttäjienhallinta -sivun tapahtumakaavio

Kuviossa 5 on kuvattuna Käyttäjienhallinta -sivua koskevat tapahtumat. Käyttäjä voi valita opettajan tai opiskelijan. Tällöin näytetään lista valitun roolin omaavista käyttäjistä ja kyseisen roolille käyttäjän lisäysvalinta. Kun käyttäjää lisätään, tarkistetaan onko käyttäjätunnus jo olemassa. Jos on, annetaan siitä virheilmoitus. Muuten lähetetään annettuun uuden käyttäjän sähköpostiin käyttäjätunnus ja salasana. Jos lähetys onnistui, luodaan uusi käyttäjä ja annetaan ilmoitus onnistumisesta. Muuten lähetetään virheilmoitus.

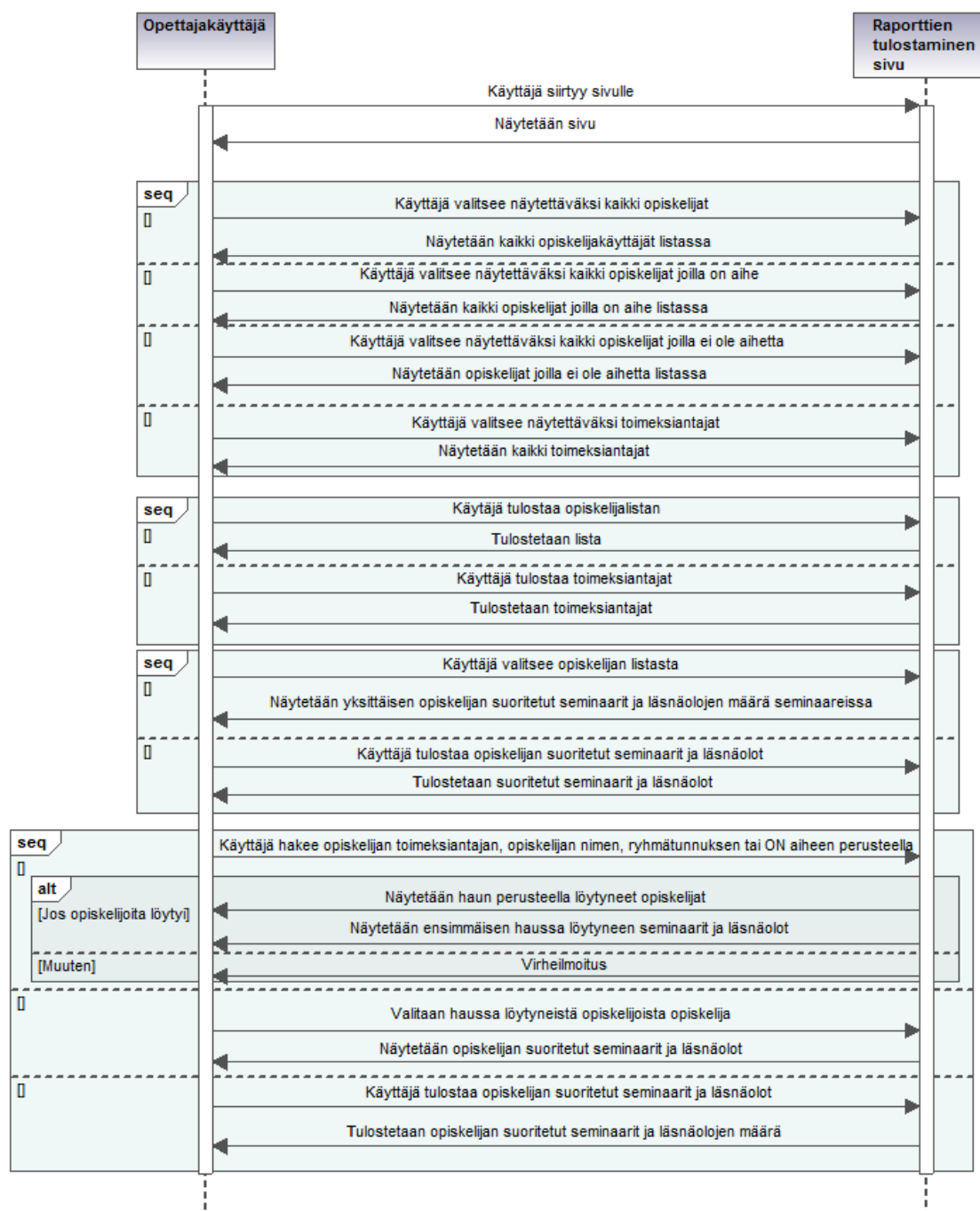
Poistettaessa käyttäjää kysytään varmistus poistolle. Uusittaessa salasanaa lähetetään käyttäjän jonka salasana uusitaan sähköpostiin käyttäjätunnus ja uusi salasana. Jos lähetys onnistui, uusitaan salasana ja annetaan ilmoitus onnistumisesta. Muuten lähetetään virheilmoitus.

4.6 Raporttien tulostaminen

Raporttien tulostaminen -sivu on ainoastaan opettajien käytettävissä. Sivulta voidaan valikoida näytettäväksi listana kaikki opiskelijat, opiskelijat joilla on aihe, opiskelijat joilla ei ole aihetta, sekä kaikki toimeksiantajat. Opiskelijoiden tiedoista näytetään opiskelijoiden opiskelijanumero, nimi, sähköpostiosoite, opinnäytteen aihe, sekä toimeksiantajan nimi ja yritys. Toimeksiantajista näytetään ainoastaan yrityksen nimi. Opiskelijoista voidaan valikoida yksittäinen opiskelija, josta näytetään suoritettut seminaari ja läsnäolot kussakin seminaarityypissä.

Sivu sisältää myös hakukentän, josta voidaan hakea tietokannasta opiskelijat etunimen, sukunimen, etu- ja sukunimen, ryhmätunnuksen ja opinnäytetyön aiheen perusteella. Haetuista opiskelijoista näytetään ryhmätunnus, opiskelijanumero, nimi ja opinnäytteen aihe. Lisäksi voidaan valikoida opiskelija jonka suoritettut seminaarit ja läsnäolot näytetään.

Valittu lista opiskelijoista tai toimeksiantajista voidaan kuvaketta painamalla tallentaa Excel-tauluksi. Kun yksittäisen opiskelijan tiedot on valittu, voidaan ne myös tallentaa Excel-tauluksi.



Kuvio 14 Raporttien tulostaminen -sivun tapahtumat

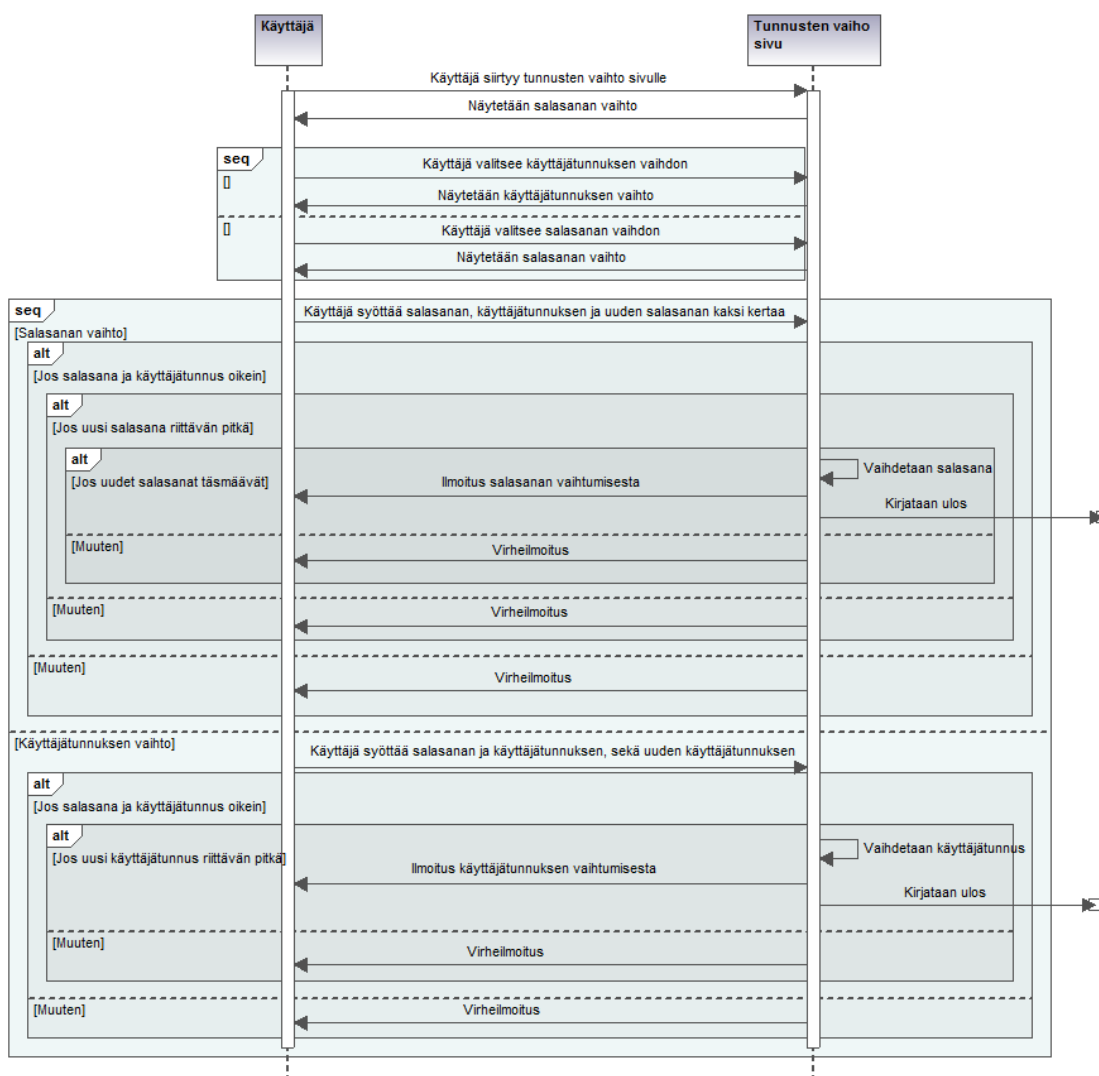
Kuviossa 6 on kuvattu Raporttien tulostaminen -sivua koskevat tapahtumat. Käyttäjä voi valita näytettäväksi kaikki opiskelijat, kaikki opiskelijat joilla on aihe, kaikki opiskelijat joilla ei ole aihetta tai toimeksiantajat. Käyttäjä voi halutessaan tulostaa valitut opiskelijat tai toimeksiantajat. Käyttäjä voi lisäksi valita

opiskelijalistasta opiskelijan jonka suoritettut seminaari- ja läsnäolomerkinnot näytetään. Käyttäjä pystyy tämän jälkeen tulostamaan merkinnät.

Käyttäjä pystyy hakemaan opiskelijoita toimeksiantajan, opiskelijan nimen, ryhmätunnuksen tai opinnäytteen aiheen perusteella. Jos haulla ei löydy tuloksia, annetaan siitä virheilmoitus. Hausta löytyneistä opiskelijoista ensimmäinen on valittuna. Tämän suoritettut seminaari- ja läsnäolomerkinnot näytetään. Käyttäjä voi vaihtaa valittuna olevaa opiskelijaa. Käyttäjä voi tulostaa valitun opiskelijan seminaari- ja läsnäolomerkinnot.

4.7 Tunnusten vaihto

Käyttäjät voivat vaihtaa omaa käyttäjätunnustaan ja salasanaansa. Käyttäjätunnusta tai salasanaa vaihtaessa täytyy käyttäjän syöttää ensin salasana ja käyttäjätunnuksensa. Tällä varmistetaan, ettei ulkopuolinen pääse niitä muuttamaan. Salasanaa vaihtaessa täytyy syöttövirheiden varalta, uusi salasana syöttää kahteen kertaan. Salasanaa vaihtaessa tarkistetaan että salasanat täsmäävät.



Kuvio 15 Tunnusten vaihto -sivun tapahtumakaavio

Kuviossa 7 kuvataan tunnusten vaihtoa kuvaavat tapahtumat. Käyttäjä siirryttyä Tunnusten vaihto -sivulle näytetään salasanan vaihto valinta. Käyttäjä voi valita joko käyttäjätunnuksen tai salasanan vaihto valikon näytettäväksi. Käyttäjän vaihtaessa salasanaa käyttäjä syöttää voimassaolevan salasanaan ja käyttäjätunnuksensa, sekä uuden salasanaan kahdesti. Jos salasana ja käyttäjätunnus eivät täsmää tai käyttäjän syöttämät uudet salasana eivät täsmää tai uusi salasana on liian lyhyt, annetaan käyttäjälle virheilmoitus. Muuten salasana vaihdetaan, käyttäjälle lähetetään ilmoitus onnistuneesta vaihdosta ja käyttäjä kirjataan ulos järjestelmästä.

Käyttäjän vaihtaessa käyttäjätunnusta käyttäjä syöttää voimassaolevan salasana ja käyttäjätunnuksensa, sekä uuden käyttäjätunnuksen. Jos salasana ja käyttäjätunnus eivät täsmää tai uusi käyttäjätunnus on liian lyhyt, annetaan käyttäjälle virheilmoitus. Muuten käyttäjätunnus vaihdetaan, käyttäjälle lähetetään ilmoitus onnistuneesta vaihdosta ja käyttäjä kirjataan ulos järjestelmästä.

Opinnäytetyön liite 3 (AMK)

Tietojenkäsittely

Tietojärjestelmät

2010

Sami Timonen

OPINNÄYTETYÖSKENTELYN SEURANTAJÄRJESTELMÄ

Testausdokumentti



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

SISÄLTÖ

1 JOHDANTO	4
2 TAVOITTEET	4
3 TESTAUSYMPÄRISTÖ.....	4
4 JATKOTOIMENPITEET	4
5 OHJELMANKEHITYKSEN AIKANA SUORITETUT TESTIT	5
6 LOPPUTESTAUKSESSA SUORITETUT TESTIT	5
6.1 Ensimmäisen kirjautumisen testaus	5
6.2 Käyttäjienhallinnan testaus	5
6.3 Kirjautumisen ja kirjautumistunnustenvaihdon testaus	6
6.4 Ilmoitusten testaus	6
6.5 Omat tiedot testaus	6
6.6 Kalenterin testaus	7
6.7 Raporttien tulostamisen testaus	7
6.8 Testauksen viimeinen vaihe	8

1 Johdanto

Tämä dokumentti on opinnäytteenä luodun opinnäytetyöskentelyn seurantajärjestelmän testausdokumentti. Dokumentissa käydään läpi kuinka sovelluksen toiminnot on testattu ohjelmankehityksen eri vaiheissa ja kuinka sovelluksen lopputestaus on suoritettu.

2 Tavoitteet

Testausten tavoitteena oli löytää sovelluksen mahdolliset virheet ja toimintoja rajoittavat tekijät. Lopputestauksen tavoitteena oli lisäksi tarkastaa, että järjestelmä täyttää sille määritellyt vaatimukset.

3 Testausympäristö

Ohjelmankehityksen aikana testaukseen käytettiin ohjelmankirjoituksessa käytettyä tietokonetta johon oli asennettuna Windows Vista, IIS 7 ja Microsoft SQL Server 2005.

Lopputestaus suoritettiin palvelimella johon oli asennettuna tuki ASP.NET 4.0:lle ja Microsoft SQL Server 2008 Express. Sovelluksen toimintaa testattiin Mozilla Firefox 3.5, Microsoft Internet Explorer 8 ja Apple Safari 4 selaimilla.

4 Jatkotoimenpiteet

Ohjelmankehityksen aikana toteutettujen testausten aikana löydetyt pienemmät virheet korjattiin heti ja suuremman ongelman löytyessä pyrittiin etsimään kaikki muutoksien aiheuttamat ongelmat.

Lopputestauksen aikana löydetyt virheet koottiin yhteen ja korjattiin mahdollisuuksien mukaan. Tämän jälkeen suoritettiin vielä viimeinen testaus, jossa testattiin lopputestauksen aikana löytyneiden virheiden korjausten toimivuus.

5 Ohjelmankehityksen aikana suoritettut testit

Ohjelmankehityksen aikana luodut testit suoritettiin heti tietyn ohjelman toiminnon luonnin jälkeen. Jos toiminto ei toiminut halutulla tavalla, korjattiin ongelma ja suoritettiin uusi testaus. Koska tämänkaltaiset yksikkötestit ovat lyhyitä, nopeita toteuttaa ja erittäin yleisiä, niitä ei tässä dokumentissa kirjata eritellysti.

6 Lopputestauksessa suoritettut testit

6.1 Ensimmäisen kirjautumisen testaus

Jotta ensimmäinen käyttäjä voitiin luoda järjestelmään, täytyi järjestelmään kirjautua ensimmäistä käyttäjän luontia varten luodun sivun avulla. Tämän sivun toimivuus testattiin luomalla ensimmäinen opettajakäyttäjä. Tässä vaiheessa havaittiin virhe klikattaessa Omat tiedot linkkiä. Virhe johtuu siitä ettei tässä kirjautumisessa käyttäjistä ole tietokannasta mitään merkintöjä. Koska tätä kirjautumista ei tulla, eikä saa käyttää muuta kuin ensimmäistä käyttäjää luotaessa, virheen korjaaminen ei ole oleellinen järjestelmän toiminnan kannalta.

Kun järjestelmän ensimmäinen opettajaroolin omaava käyttäjä oli luotu, testattiin luodulla käyttäjällä kirjautuminen ja luotiin käyttäjät muita testaa- jia varten.

6.2 Käyttäjienhallinnan testaus

Käyttäjienhallinnan toimivuus testattiin luomalla uusia opettaja- ja oppilasroolien omaavia käyttäjiä. Tämän jälkeen käyttäjien tietoja muutettiin, osa luoduista käyttäjistä poistettiin ja salasanan uusimista testattiin.

Salasanan uusimisen yhteydessä jotkin sähköpostiviestit eivät saapuneet perille. Tämä luultavimmin johtuu siitä, että testauksessa käytettiin gmail sähköpostia. Jotkut sähköpostipalvelun tarjoajista luokittelevat gmail

sähköpostiosoitteesta lähetetyt automaattiset viestit roskaposteiksi. Tämä luultavimmin korjaantuu, kun järjestelmään otetaan käyttöön palvelimen oma virallinen sähköposti.

Oppilasroolin omaavilla käyttäjillä testattiin, että heiltä on evätty pääsy käyttäjienhallintaan.

6.3 Kirjautumisen ja kirjautumistunnustenvaihdon testaus

Kirjautumisen toimivuus testattiin kirjautumalla järjestelmään opettajaroolin ja oppilasroolin omaavilla käyttäjillä. Lisäksi järjestelmään yritettiin kirjautua sisään pelkän käyttäjätunnuksen, pelkän salasanan, keksityn käyttäjätunnuksen ja toimivan salasanan avulla, sekä keksityn salasanan ja toimivan käyttäjätunnuksen avulla. Tämän jälkeen käyttäjät vaihtoivat, joko salasanaansa, käyttäjätunnustaan tai molempia ja testattiin järjestelmään kirjautumista uudestaan sisälle.

Kirjautumisen, sekä tunnustenvaihdossa ei löydetty virheitä.

6.4 Ilmoitusten testaus

Ilmoitusten testaus suoritettiin kirjautumalla järjestelmään opettajaroolilla ja luomalla ilmoitus. Tämän jälkeen kirjaututtiin toisella opettajaroolin omaavalla käyttäjällä, sekä oppilasroolin omaavalla käyttäjällä ja todettiin etteivät he pysty muokkaamaan luotua ilmoitusta. Järjestelmään kirjaututtiin uudestaan ilmoituksen jättäneenä käyttäjänä ja testattiin jätetyn ilmoituksen muokkaaminen, sekä ilmoituksen poisto.

Ilmoitusten testauksessa ei löydetty virheitä.

6.5 Omat tiedot testaus

Omat tiedot -sivun toimivuus testattiin opettajaroolin ja oppilasroolin omaavalla käyttäjällä.

Testauksessa ei havaittu virheitä.

6.6 Kalenterin testaus

Kalenterin toimintoja testattiin opettajaroolin, sekä oppilasroolin omaavilla käyttäjillä. Opettajaroleilla luotiin ensin seminaari ja kypsyysnäyte tilaisuuksia kalenteriin, muokattiin tapahtumia ja osa tapahtumista poistettiin. Tämän jälkeen oppilasrooleilla kirjauduttiin järjestelmään ja ilmoitauduttiin luotuihin tilaisuuksiin kuuntelijoiksi ja esiintyjiksi. Tämän jälkeen opettajaroleilla kirjauduttiin uudestaan järjestelmään ja kuitattiin osa oppilaskäyttäjien suorituksista, sekä poistettiin tilaisuuksia. Tämän jälkeen tulostettiin esiintymislistat Excel-tauluina.

Testauksessa havaittiin suoritusten kuittauksen valikon näkyminen valitessa tapahtumaa. Ongelma korjattiin sijoittamalla suoritusten kuittauksen valikko oikeaan kohtaan sivua.

6.7 Raporttien tulostamisen testaus

Raporttien tulostaminen testattiin kalenterimerkintöjen luontien jälkeen. Raportit avattiin Microsoft Excel 2007 ja OpenOffice.org 3.0 Calc ohjelmilla. Testauksessa tulostettiin tiedot kaikista opiskelijoista, opiskelijoista joilla on aihe, opiskelijoista joilla ei ole aiheita, lista toimeksiantaja yrityksistä ja yksittäisen opiskelijan tiedot. Tämän jälkeen testattiin hakukentän toimivuus, hakemalla opiskelija etunimen, sukunimen, ryhmätunnuksen, opinnäytteen aiheen, etunimen ja sukunimen sekä opiskelijanumeron perusteella. Lopuksi testattiin, ettei opiskelजारoolin omaava käyttäjä pääse raporttien tulostamista varten luodulle sivulle.

Testauksessa havaittiin etunimen ja sukunimen perusteella luodun haun antavan listan kaikista annetun etunimen ja annetun sukunimen omaavista henkilöistä. Ongelma korjattiin muuttamalla haun suorittavaa sql-lausetta.

6.9 Ohjeiden näkyvyyden testaus

Ohjeiden näkyvyys testattiin jokaisella sivulla erikseen, opettajakäyttäjällä ja opiskelijakäyttäjällä.

Ohjeiden testauksessa ei löydetty virheitä.

6.8 Testauksen viimeinen vaihe

Kun valitut virheet oli koottu ja korjattu, asennettiin järjestelmän uusi versio palvelimelle. Tämän jälkeen suoritettiin korjattujen virheiden testaus.

Testauksen viimeisessä vaiheessa ei löytynyt virheitä.