LAPIN AMK
Lapland University of Applied Sciences

ENSIHOIDON OPPIMISYMPÄRISTÖN KAMERAVALVONTAJÄRJESTELMÄ

Jani Mikkola

Opinnäytetyö
Tekniikka ja liikenne
Tietotekniikan koulutusohjelma
Insinööri (AMK)

2015

LAPIN AMK
Lapland University of Applied Sciences

Opinnäytetyön tiivistelmä

Tekniikka ja liikenne
Tietotekniikan koulutusohjelma

Tekijä	Jani Mikkola	Vuosi	2015
Ohjaaja	Maisa Mielikäinen		
Toimeksiantaja	Ohjelmistotekniikan laboratorio pLAB		
Työn nimi	Ensihoidon oppimisympäristön kameravalvontajärjestelmä		
Sivu- ja liitemäärä	48 + 5		

Opinnäytetyön aiheena oli toteuttaa www-käyttöliittymä, jonka kautta on mahdollista seurata opetusta reaaliajassa. Työn idea syntyi, kun pohdittiin ENVI-oppimisympäristössä sijaitsevien verkkokameroiden hyödyntämismahdollisuuksia. ENVI on sosiaali- ja terveysalan opiskelijoiden ensihoidon virtuaalinen oppimisympäristö. Työn toimeksiantajana on Lapin Ammattikorkeakoulun ohjelmistotekniikan laboratorio pLAB.

Opinnäytetyön tavoitteena oli toteuttaa www-sivusto jonka kautta muodostetaan yhteys Axiksen ip-kameroihin. Usean kameran suoratoistoa tulisi pystyä katsomaan yhtäaikaisesti samalta selainikkunalta. Käyttöliittymän avulla tulisi mahdollistaa kameraryhmien luominen huoneittain sekä huoneiden suojaaminen harjoitteiden ajaksi. Sivusto toteutettiin CakePHP-sovelluskehyksellä, joka on toteutettu PHP-ohjelmointikielellä. Työssä käytettiin myös JavaScript-ohjelmointikieltä sekä jQuery-komponentteja. Käyttöliittymän ja kameroiden välinen kommunikaatio mahdollistetaan Axiksen Vapix -ohjelmointirajapinnan avulla. Käyttäjien todennuksessa hyödynnetään Microsoft Active Directory -käyttäjätietokantaa. Työn keskeisenä ongelmana oli opetuksen aikaisten istuntojen turvaamiseen käytettävän menetelmän valinta sekä sen toteuttaminen sivustolle.

Työn tuloksena saatiin toteutettua www-sivusto, joka mahdollistaa ENVI-oppimisympäristön harjoitteiden seuraamisen etänä. Sivusto tarjoaa tarpeelliset työkalut kameroiden hallitsemiseksi käyttöliittymässä sekä opetuksen aikaisten turvallisten istuntojen luomiseksi.

Avainsanat CakePHP, Axis Ip-kamera, Www-käyttöliittymä, LDAP

LAPIN AMK
Lapland University of Applied Sciences

Abstract of Thesis

School of Technology, Communication and Transport
Information Technology Programme

Author	Jani Mikkola	Year	2015
Supervisor(s)	Maisa Mielikäinen		
Commissioned by	Software Engineering Laboratory pLAB		
Subject of thesis	Emergency care learning environment surveillance system		
Number of pages	48 + 5		

The subject of this Bachelor's thesis was to create a web user interface, through which it would be possible to follow teaching in real time. There was a need to rethink the possibilities to utilize the network cameras in the ENVI learning environment. ENVI is a virtual learning environment for the social and health care students. This thesis project was commissioned by pLAB. pLAB is a software engineering laboratory that is a part of Lapland University of Applied Sciences.

The objective of this thesis was to create a website through which one can connect to the Axis' IP cameras. It should be possible to simultaneously watch the live feed from multiple cameras with just one browser window. Through the user interface it should be possible to create different camera groups by room, and to secure a room for the duration of the teaching. The website was created with the CakePHP framework, which is created with the PHP programming language. The JavaScript programming language and the jQuery components were also applied in the project. The communication between the user interface and the cameras is made possible with the Axis' Vapix application programming interface. The Microsoft Active Directory user database was used for the user authentication. The central problem in this project was to choose the method that would be used to secure the sessions during teaching and its implementation to the website.

The result of this thesis was a website that allows following the practical teaching remotely. The website offers the necessary tools to control the cameras in the user interface, and to create safe sessions during teaching.

Key words CakePHP, Axis Network Camera, Web user interface, LDAP

SISÄLLYS

KUVIOLUETTELO 1
ESIMERKKIKOODI -LUETTELO 2
KÄSITELUETTELO 3
1 JOHDANTO 5
2 SOVELLUSKEHYS JA RAJAPINNAT 6
2.1 CakePHP 6
2.1.1 Yleistä 6
2.1.2 Nimeämiskäytännöt 7
2.1.3 Ydinkirjastot 8
2.2 Vapix 10
3 KÄYTETYT TYÖVÄLINEET JA TEKNOLOGIAT 13
3.1 Axis IP -kamerat 13
3.2 LDAP 16
3.3 Bcrypt ja PHP 18
4 SIVUSTON SUUNNITTELU 20
5 SIVUSTON TOTEUTUS 24
5.1 Käyttöliittymä 24
5.2 Tietokanta 26
5.3 Pääkäyttäjän työkalut 27
5.4 Kameroiden näkymät 32
5.5 LDAP ja sisäänkirjautuminen 36
5.6 Istuntojen suojaaminen 38
6 TESTAUS JA JATKOKEHITYS 42
6.1 Testaus 42
6.2 Jatkokehitys 43
7 YHTEENVETO 45
LÄHTEET 46
LIITTEET 48

KUVIOLUETTELO

Kuvio 1. MVC-pyyntö CakePHP-sovelluksessa (CakePHP 2015a, 44) 6
Kuvio 2. Suositukset nimeämiseen sivustolla 7
Kuvio 3. HTTP-pyynnön syntaksi 10
Kuvio 4. IP-kameran yleiset näkyvät osat ja liitännät (Axis 2014a, 16) 13
Kuvio 5. Vasemmalla Axis M1034-W-verkkokamera ja oikealla Axis P3304 - verkkokamera 14
Kuvio 6. Kameroiden hallintasivu 15
Kuvio 7. Hakemistopalvelun puurakenne 16
Kuvio 8. Työvaiheet siirtymineen 20
Kuvio 9. Käyttöliittymän ja kameran välinen kommunikointi 21
Kuvio 10. Sivuston käyttäjäryhmät ja niiden oikeudet 22
Kuvio 11. Web-sovelluksen alustava sivustokartta 23
Kuvio 12. Sivuston navigaatiorakenne 25
Kuvio 13. Lapin Ammattikorkeakoulun värikartta (Lapin AMK 2013, 6) 26
Kuvio 14. Tietokannan käsitemalli.....26
Kuvio 15. Näkymä uuden huoneen lisäämiselle.....28
Kuvio 16. Huoneiden hallintataulukko käyttöliittymässä.....30
Kuvio 17. Näkymä uuden kameran lisäämiselle tietokantaan 31
Kuvio 18. Sivuston päänäkymä huoneineen 32
Kuvio 19. Huoneen näkymä kameroineen 33
Kuvio 20. Käyttäjän tunnistukseen käytettävä lomake 36
Kuvio 21. Lomake istunnon luomiseen huoneelle 38
Kuvio 22. Päivämäärän ja ajan valintaan käytetty komponentti 39
Kuvio 23. Näkymä sessioavaimen tarkistukseen 40

ESIMERKKIKOODI-LUETTELO

Esimerkkikoodi 1. Merkkijonon tiivistäminen *password_hash()*-funktiolla (PHP 2015c) 19
Esimerkkikoodi 2. Tiivisteiden vertaaminen keskenään *password_verify ()*-funktiolla (PHP 2015d) 19
Esimerkkikoodi 3. Käyttöliittymän runko 24
Esimerkkikoodi 4. CakePHP-tietokanta-asetukset 27
Esimerkkikoodi 5. Hiiren sijainnin kaappaaminen klikatessa 29
Esimerkkikoodi 6. Käsittelijän *add()* -funktio 29
Esimerkkikoodi 7. Funktio huoneen muokkaamisesta käsittelijässä 30
Esimerkkikoodi 8. Suoratoiston esittäminen näkymässä 33
Esimerkkikoodi 9. Valitun esikatselukuvan esittäminen suurempana 34
Esimerkkikoodi 10. Suoratoiston katsominen koko näytöllä 35
Esimerkkikoodi 11. UserController *login()* -funktio 36
Esimerkkikoodi 12. Käyttäjän uloskirjaaminen 37
Esimerkkikoodi 13. JavaScript-funktio sessioavaimen generoimiselle 38
Esimerkkikoodi 14. Aktiivisen istunnon luominen käyttäjälle 40
Esimerkkikoodi 15. Ote *view()*-funktion aktiivisen session tarkistamisesta 41

KÄSITELUETTELO

Active Directory	Hakemistopalvelu, joka sisältää tietoa Windows-toimialueen käyttäjistä, tietokoneista ja verkon resursseista (Wikipedia 2015).
Bootstrap	Sovelluskehys, joka sisältää työkalut käyttöliittymän työstämiseen (Bootstrap 2015).
CGI	Common Gateway Interface on standardoitu viestintämenetelmä asiakasohjelman ja palvelimen välillä. (Axis 2013c).
Ctp	CakePHP:n mallitiedosto, johon sivuston näkymät kasataan.
ENVI	Sosiaali- ja terveysalan opiskelijoiden virtuaalinen oppimisympäristö Lapin AMKin tiloissa.
FPS	Frames per second eli kuvataajuus.
FTP	File Transfer Protocol on protokolla joka mahdollistaa tiedonsiirron kahden tietokoneen välille (Microsoft Windows 2015).
HTTP	Hypertext Transfer Protocol on protokolla, jota selaimet sekä palvelimet käyttävät tiedonsiirtoon.
JavaScript	Skriptikieli, joka mahdollistaa sisällön luomisen www-sivulle (Ruohonen 2002, 110).
LDAP	Lightweight Directory Access Protocol on protokolla, joka mahdollistaa verkkohakemistopalvelujen käyttämisen.
Motion JPEG	Videonpakkausformaatti.
MVC	Lyhenne sanoista Model View Controller. Ohjelmistoarkkitehtuurityyli.
MySQL	Relaatiotietokanta, jota käytetään www-palvelujen taustalla (Heinisuo 2004, 34).
ORM	Lyhenne sanoista Object Relational mapping. Mahdollistaa datan muuntamisen yhteen sopimattomien tyyppijärjestelmien välillä.

PHP	Lyhenne sanoista Hypertext Preprocessor. Ohjelmointikieli www-sivujen luomiseen. (Heinisuo 2004, 16.)
PIR	Passiivinen infrapunatunnistin.
PTZ	Lyhenne sanoista Pan Tilt Zoom. Kameran pyörimisen, kulman ja polttovälialueen hallinta. (Axis 2014a.)
RTSP	Real Time Streaming Protocol. Hallinta protokolla palvelimen lähettämälle suoratoistolle. (Axis 2013a.)
SHA	Secure Hash Algorithm. Tiivistefunktio.
Sovelluskehys	Kokoelma kirjastoja ja toiminnallisuuksia, joiden avulla www-sovellus toteutetaan.
TCP	Transmission Control Protocol. Protokolla tietokoneiden väliseen tiedonsiirtoon.
Xampp	Www-sovellusten kehitystyökalu, joka sisältää Apache-palvelimen, MySQL-tietokannan ja PHP:n.

1 JOHDANTO

Virtuaaliset oppimisympäristöt tarjoavat turvallisen ja helposti lähestyttävän tavan tosielämän tilanteiden harjoittelemiseen. Opinnäytetyön aiheena on toteuttaa verkkosivusto, jonka avulla oppimisympäristön harjoitteiden seuraaminen mahdollistetaan etänä. Aiheen idea syntyi ENVI-oppimisympäristössä sijaitsevien verkkokameroiden hyödyntämisestä opetusvälineenä. ENVI on sosiaali- ja terveysalojen virtuaalinen oppimisympäristö, joka on tarkoitettu hoitotilanteiden harjoitteluun. Oppimisympäristö simuloi moniaistillisia ja aidontuntuisia työtilanteita. (Lapin AMK 2015a.)

Opinnäytetyön tavoitteena oli www-sivuston toteuttaminen, joka tarjoaa mahdollisuuden seurata reaaliaikaisesti ENVI-oppimisympäristön harjoitteita. Käyttöliittymään kootaan yhteen oppimisympäristössä sijaitsevat IP-kamerat. Harjoitteiden aikaisen suoratoiston katseluoikeutta rajoitetaan suojatulla istunnolla. Verkkokameroiden sekä käyttöliittymän välinen tiedonsiirto mahdollistetaan HTTP-protokollan avulla. Sivuston toteutuksessa käytettiin CakePHP-sovelluskehystä.

Opinnäytetyön toinen luku esittelee käytetyn sovelluskehyksen sekä kameroiden ohjelmointirajapinnan. Kolmannessa luvussa kerrotaan olennaisimmat sivuston toteutuksessa käytetyt työvälineet sekä teknologiat. Luvussa neljä käydään läpi vaiheittain sivuston suunnitteluprosessi. Viidennessä luvussa siirrytään sivuston toteutuksen läpikäymiseen suunnittelun mukaisessa järjestyksessä. Kuudes luku esittelee ominaisuuksia, joiden toteuttaminen sivustolle olisi mahdollista tulevaisuudessa sekä testauksessa huomioitavia asioita. Viimeisessä kappaleessa pohditaan prosessin toteutumista sekä sen onnistumista.

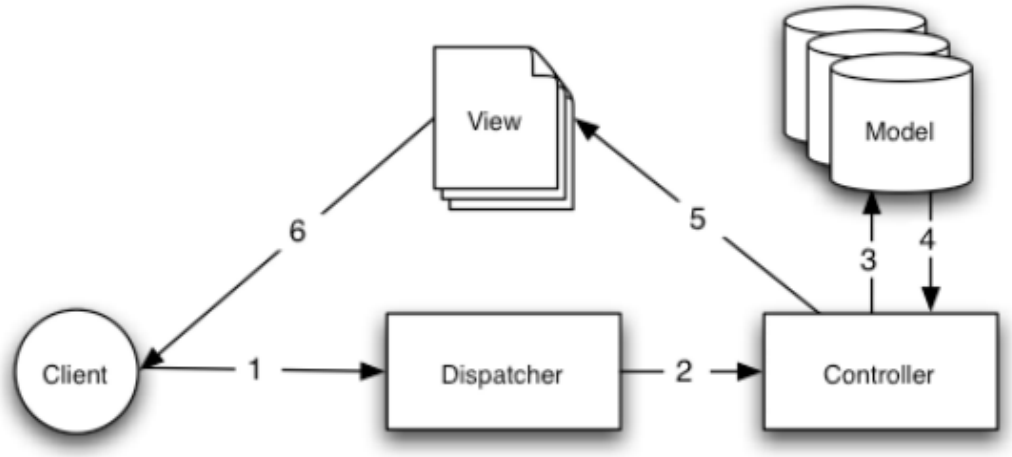
Opinnäytetyön tilaajana on Lapin Ammattikorkeakoulun ohjelmistotekniikan laboratorio pLAB, joka oli mukana toteuttamassa ENVI-hanketta. Ohjelmistotekniikan laboratorion osaamisalueina ovat reaaliaikaiset 3D-visualisointiympäristöt, ohjelmisto- ja mittausteknologiat, mobiilisovellukset, www-sivustot sekä työpöytäsovellukset (Lapin AMK 2015b). Opinnäytetyön aihe valittiin, koska se tarjosi mahdollisuuden tutustua uusiin mielenkiintoisiin teknologioihin.

2 SOVELLUSKEHYS JA RAJAPINNAT

2.1 CakePHP

2.1.1 Yleistä

CakePHP on PHP:lle tarkoitettu sovelluskehys, joka sisältää tarvittavat työkalut kehityksen nopeaan aloittamiseen. Se noudattaa MVC-ohjelmistokehitysarkkitehtuurityyliä, joka mahdollistaa helposti hallittavan ja ylläpidettävän kokonaisuuden. MVC-arkkitehtuuri jakaa ohjelman kolmeen osaan, jotka ovat: malli (Model), näkymä (View) sekä käsittelijä (Controller). Mallitaso hoitaa sovelluksen datan vastaanottamisen sekä käsittelyn. Esimerkiksi lomakkeiden sisällön tarkistus suoritetaan mallissa. Näkymätasolla tarkoitetaan selaimella näkyvää osaa sovelluksesta. Viimeinen taso on käsittelijä, joka hoitaa käyttäjän pyynnöt. (CakePHP 2015a, 41–43.)



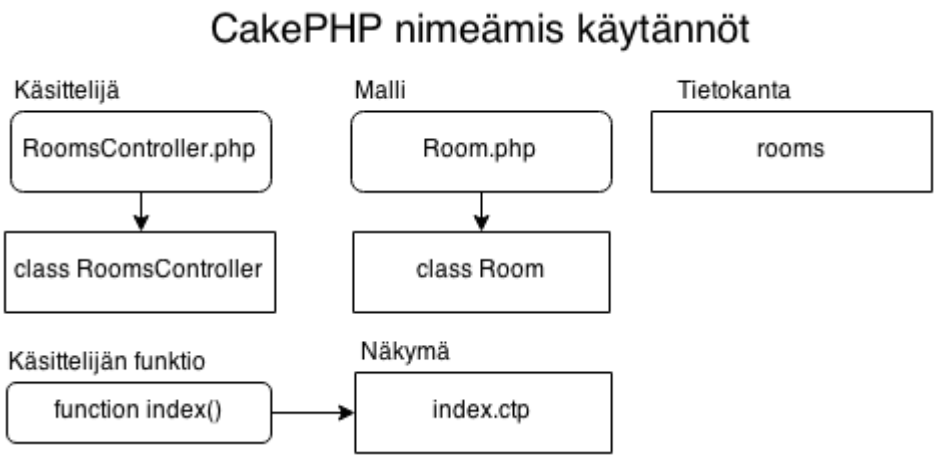
Kuvio 1. MVC-pyyntö CakePHP-sovelluksessa (CakePHP 2015a, 44)

Tyypillinen MVC-pyyntö CakePHP-sovelluksessa alkaa käyttäjän pyytäessä haluamaansa sivua sovelluksen sisällä (Kuvio 1). Lähettäjä (Dispatcher) käsittelee pyynnön ohjaamalla sen oikeaan käsittelijään. Käsittelijä hyödyntää malleja datan validoinnissa ja muokkaamisessa. Käsittelijästä pyyntö ohjataan oikeaan näkymään, jossa mallin sekä käsittelijän tarjoama tieto esitetään käyttäjän selaimella. (CakePHP 2015a, 44.)

Web-käyttöliittymä on toteutettu CakePHP-versiolla 2.6, joka oli sivuston kehityksen alkaessa vakain versio. Maaliskuussa 2015 julkaistiin versio 3.0.0, joka vie sovelluskehystä huomattavasti eteenpäin. Uudistuksia ja parannuksia vanhempiin versioihin on esimerkiksi kokonaan uusi ORM, nopeampi reititys, uudelleenkirjoitettu lokalisaatiokirjasto sekä paranneltu testaustyökalu. (CakePHP 2015e.)

2.1.2 Nimeämiskäytännöt

CakePHP sisältää paljon erilaisia käytäntöjä, joiden opetteleminen ja noudattaminen vähentävät huomattavasti kehittäjän työmäärää. Käytäntöjä oikein käytettäessä ylimääräisen konfiguraation tarve poistuu. Käsittelijät suositellaan nimeämään monikossa, luokat nimetään yhteenkirjoitettuna ja isoin alkukirjaimin sekä loppuun lisätään Controller-pääte. Esimerkiksi nimettäessä käsittelijän luokan nimellä RoomsController CakePHP olettaa PHP-tiedoston kantavan samaa nimeä. Mallit suositellaan nimeämään yksikössä sekä yhteenkirjoitettuna isoin alkukirjaimin. Malli, joka on sidoksissa esimerkin käsittelijään, tulisi nimetä seuraavasti: *Room*. Tietokantataulut nimetään mallin mukaan monikossa ja pienin kirjaimin, esimerkiksi *rooms*. Näkymät suositellaan nimeämään käsittelijän funktioiden mukaan. Tällöin näkymän toiminnot saadaan suoraan yhdistettyä oikeaan funktioon ilman ylimääräistä asetusten muokkaamista. (CakePHP 2015a, 22–25.) Nimeämiseen liittyvät käytänteet esitetään tiivistettynä kuviossa 2.



Kuvio 2. Suositukset nimeämiseen sivustolla

2.1.3 Ydinkirjastot

CakePHP sisältää kehitystä helpottavia kirjastoja, jotka tarjoavat yleisimpiä ominaisuuksia www-sivujen kehitykseen. Kirjastot voidaan jakaa viiteen ryhmään niiden tarkoitusperän mukaisesti. Kyseiset kirjastot ovat: yleiset, käyttäytymistä koskevat, komponentit, avustajat ja työkalut. Omien kirjastojen luominen on myös mahdollista. (CakePHP 2015b.)

Yleisiä kirjastoja ovat: *globaalit vakiot*, *globaalit funktiot*, *sovellusluokka*, *järjestelmän tapahtumat* ja *kokoelmat*. *Globaaleja funktioita* käytetään esimerkiksi näkymien sekä käsittelijöiden lokalisoinnissa, merkkijonojen käsittelyssä sekä testaamisessa. *Sovellusluokasta* löytyy kirjastoja pakettien hallintaan, luokkien lataamiseen, teemojen ja liitännäisten paikallistamiseen sekä luokkien kuormittamiseen. *Järjestelmän tapahtumia* koskeva kirjasto sisältää tapahtumien kuuntelijoihin sekä hallintaan liittyviä funktioita. *Kokoelma*-objektit tarjoavat yhtenäisen toimintatavan monien erilaisten objektien hallitsemiseen, esimerkiksi niiden lataamiseen. (CakePHP 2015c.)

Käyttäytymistä koskevat kirjastot lisäävät ominaisuuksia malliin. Niiden avulla pystytään uudelleen hyödyntämään haluttuja toiminnollisuuksia ilman perintää. Valmiita käyttäytymismalleja ovat *puurakenne*, *sisällön kääntäminen*, *rajaaminen* sekä *käytönvalvontalista*. *Puukäyttäytymistä* hyödynnetään halutessa varastoida dataa tietokantaan hierarkkisesti, esimerkiksi verkkokaupan katalogia toteutettaessa. Tärkeä ominaisuus on mallien *callback*-kutsut, jotka sisältyvät käyttäytymisen kirjastoihin. Ne ovat funktioita, joiden avulla saadaan esimerkiksi lomakkeiden tietoja käsiteltyä ennen tietokantaan tallennusta. (CakePHP 2015d.)

Komponenttikirjastot sisältävät perustyökaluja, joita käsittelijät voivat hyödyntää perustehtävien suorittamisessa. Niiden avulla käsittelijöiden kuorma vähenee huomattavasti, eikä samoja ominaisuuksia tarvitse ohjelmoida uudelleen jokaiseen käsittelijään. Valmiita komponentteja ovat *istunnot*, *käyttäjien todennus*, *sivuston turvallisuus*, *pyyntöjen käsittely*, *evästeet* sekä *sivutus*. (CakePHP 2015f.)

Avustajakirjastot helpottavat sivuston näkymien luomista. Niiden tarkoituksena on helpottaa ja nopeuttaa perustoimintojen ohjelmointia. CakePHP:n ydinkirjastossa on kymmenen valmista avustajaa, esimerkiksi *html-, lomake-, istunto-* ja *välimuistiavustajat*. Suuri osa CakePHP:n avustajista tarjoaa vaihtoehtoisen tavan kirjoittaa html-koodia. (CakePHP 2015g.) Taulukossa 1 vertaillaan perinteisen html:n ja avustajien tarjoamia ratkaisuja keskenään.

Taulukko 1. CakePH-avustajien vertailu HTML-tageihin

Html	**CakePHP-avustajan vastike**	**Tarkoitus**
<form id="RecipeAddForm" method="post" action="/recipes/add">	echo $this->Form->create('Recipe');	Luo lomakkeen näkymään
<link rel="stylesheet" type="text/css" href="/css/form.css" />	echo $this->Html->css('forms');	Tyylitiedoston käyttöönotto näkymässä
Tallennus onnistui.	echo $this->Session->setFlahs('Tallennus onnistui');	Ilmoitusikkuna onnistuneesta tapahtumasta
User account	echo $this->Paginator->sort('user_id', 'User account');	Sivutuksen lisääminen näkymään

Viimeistä CakePHP:n ydinkirjastoa kutsutaan *työkalut*-kirjastoksi. Kirjasto sisältää kuusitoista komponenttia, joiden toiminnallisuudet helpottavat lukuisten ominaisuuksien käyttöönottoa sivustolla. Tämän kappaleen loppu sisältää lyhyet kuvaukset olennaisimmista työkaluluokista. *Välimuisti*-luokkaa hyödynnetään resurssien luomisen ja lukemisen nopeuttamiseen. *Sähköposti*-luokkaa käytetään sähköpostiviestin lähettämiseen sovelluksen sisällä. *Kansio- ja tiedosto*-luokat auttavat esimerkiksi datan lukemisessa ja kirjoittamisessa tiedostoihin. *HttpSocket*-luokka mahdollistaa kommunikoinnin ulkoisten verkkopalveluiden kanssa. *Taivuttaja*-luokan avulla saadaan käsiteltyä merkkijonoja. *Kansainvälistämis- ja lokalisointi*-kirjastojen avulla mahdollistetaan sivuston sisällön kääntäminen usealle kielelle. *Turvallisuuskirjastosta* löydetään tarvittavat luokat esimerkiksi salasanojen tiivistämiseen. (CakePHP 2015h.)

2.2 Vapix

```
http://<servername>/axis-cgi/<subdir>[/<subdir>...]<cgi>.<ext>[?<argument>=<value>[&<
argument>=<value>...]]
```

Kuvio 3. HTTP-pyynnön syntaksi

Vapix on Axiksen tarjoama ohjelmointirajapinta, jonka avulla kehittäjät pystyvät hyödyntämään kameroita omissa projekteissaan. Kameroita on mahdollista hyödyntää rajapinnan avulla työpöytä-, mobiili- sekä selainsovelluksissa. Rajapinta mahdollistaa pääsyn esimerkiksi kameran videokuvaan, ääneen, I/O-portteihin sekä PTZ-ominaisuuksiin. (Axis 2015b.) Vapix tarjoaa kaksi mahdollista lähestymistapaa kameroiden ominaisuuksien käyttämiseen: HTTP- ja RTSP-ohjelmointirajapinnat (Axis 2013a). Opinnäytetyössä hyödynnettiin HTTP-ohjelmointirajapintaa

Taulukko 2. Esimerkki HTTP-pyynnöistä kameran palvelimelle sekä palautettavista toiminnoista (mukaillen Axis 2013b)

HTTP pyyntö	**Palautetaan**
http://servername/axis-cgi/mjpeg/video.cgi	Motion JPEG suoratoisto kamerasta
http://servername/axis-cgi/restart.cgi	Käynnistää kamera uudelleen
http://servername/axis-cgi/audio/receive.cgi	Äänen suoratoisto kamerasta

Verkkokameran toiminnollisuudet saadaan sovellukseen lähettämällä HTTP-pyyntöjä selaimelta kameran palvelimelle (Taulukko 2). Palvelupyyntö sisältää verkkokameran palvelimen nimen, viitteen *axis-cgi*-hakemistoon sekä mitä hakemistosta haetaan (Kuvio 3). Palvelimen nimellä tarkoitetaan kameran nimeä tai ip-osoitetta. *Axis-cgi*-hakemisto sisältää tarvittavat funktiot kameran hallintaan. (Axis 2013c.) Taulukosta 3 on luettavissa esimerkkejä tarjolla olevista hakemistoista käyttötarkoituksineen.

Taulukko 3. Cgi-hakemiston toiminnallisuuksia (mukaillen Axis 2015c)

CGI hakemisto	**Käyttötarkoitus**
axis-cgi	Juurikansio. Sisältää loput hakemistot ja toiminnot (esim. juuressa sijaitsee kameran hallintaan liittyvät toiminnot, kuten restart.cgi)
audio	Kameran ääneen liittyvät toiminnot
mjpeg	Kameran Motion JPEG-videokuvaan liittyvät toiminnot
jpeg	Kameran still-kuviin liittyvät toiminnot
io	IO-porttien hallintaan liittyvät toiminnot
ptz	Pan/tilt/zoom toiminnot

Verkkokameran suoratoisto saadaan näkymään HTTP-pyynnöllä, jossa palvelimelta pyydetään Motion JPEG -kuvaa käyttöliittymään. Lisättäessä mjpeg-hakemiston *video.cgi*-komennon HTTP-pyyntöön kameran palvelin lähettää sivustolle vastauksena liikkuvaa kuvaa. Suoratoiston laatuun on mahdollista vaikuttaa taulukossa 4 esitellyillä muuttujilla.

Taulukko 4. CGI-pyynnön muuttujia Motion JPEG -suoratoistolle (mukaillen Axis 2013a, 12–14)

Muuttuja	**Sallitut arvot**	**Kuvaus**
fps=<int>	etumerkitön kokonaisluku	Kuvataajuuden määrittäminen verkkokameran suoratoistolle
resolution=<string>	merkkijono	Palautetun kuvan resoluutio
compression=<int>	0 … 100	Kuinka paljon kuvaa pakataan. Mitä korkeampi arvo, sitä huonompi kuvanlaatu ja pienempi koko
rotation=<int>	0, 90, 180, 270	Pyörittää kuvaa myötä päivään

Pyydettävän suoratoiston laatu määrittää kameran palvelimelta lähetettävien pakettien koon. Kuvanlaatu sekä -taajuus voidaan pitää suurena näkymissä, joihin ei pyydetä useamman kameran suoratoistoa kerralla. Halutessa sivustolla voidaan esittää useamman kameran suoratoisto yhtäaikaisesti. Tällöin olisi suositeltavaa pakata kuvaa pienemmäksi, jotta selainikkunan lataaminen ei hidastuisi merkittävästi. Lähetettävien pakettien kokoa voi myös pienentää asettamalla kuvataajuuden arvo mahdollisimman alhaiseksi.

3 KÄYTETYT TYÖVÄLINEET JA TEKNOLOGIAT

3.1 Axis IP -kamerat

Axis on maailman johtavia verkkokameroiden valmistajia. Yritys työllistää yli 40 maassa yli 1940 työntekijää sekä jälleenmyyjiä yli 70 maassa. Ensimmäisen verkkokameran yritys toi markkinoille vuonna 1996. Nykyään yritys valmistaa IP-kameroiden lisäksi video-enkoodereita, verkkovideotallentajia ja ohjelmistoja tuotteidensa hallintaan. (Axis 2015a.)

Kuvio 4. IP-kameran yleiset näkyvät osat ja liitännät (Axis 2014a, 16)

Kuviosta 4 on nähtävillä verkkokameran ulkoiset osat. IP-kamera voidaan ajatella videokameran ja tietokoneen yhdistelmänä. Se kykenee kommunikoimaan tietokoneverkoissa, joten videokuvaa pääsee seuraamaan reaaliajassa etänä laitteilla, kuten tietokoneella tai puhelimella. Verkkokameroilla on oma IP-osoite, jonka syöttäminen selaimen osoiteriville mahdollistaa suoratoiston katsomisen sekä pääsyn laitteen asetuksiin. (Axis 2014a, 15–16.)

ENVI-tiloissa on useita verkkokameroita, joita käytetään harjoitteiden seuraamiseen. Kamerat ovat sijoitettuina huoneisiin ryhmittäin mahdollisimman laajan sekä kattavan yleiskuvan muodostamiseksi. Huoneissa käytössä olevat verkkokameramallit ovat M1034-W sekä P3304 (Kuvio 5).

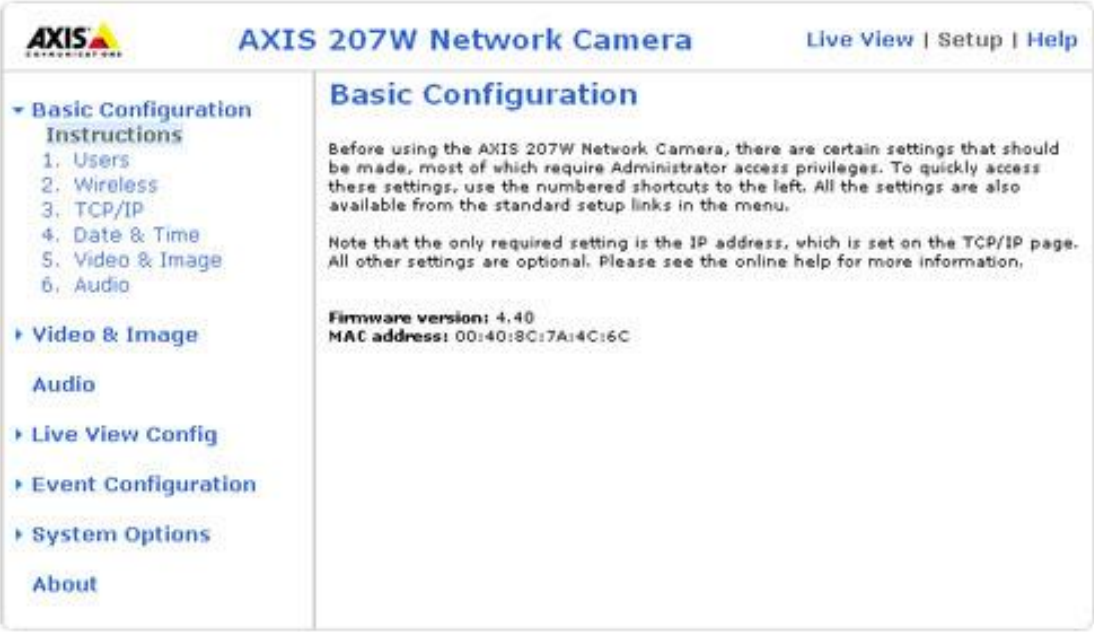
Kuvio 5. Vasemmalla Axis M1034-W -verkkokamera ja oikealla Axis P3304 -verkkokamera

Axiksen M1034-W-verkkokamera on kooltaan pieni: leveys 59 mm, korkeus 95 mm ja paksuus 41 mm. Laite sisältää I/O-portin, mikrofonin, kaiuttimen, PIR-sensorin ja mahdollisuuden käyttää langatonta verkkoa yhteyden muodostamiseen. (Axis 2014b, 1–2.) I/O(input/output)-portti mahdollistaa lisälaitteiden kytkemisen kameraan sekä niiden hallitsemisen (Axis 2014a, 56). PIR-sensoria käytetään lämpösäteilyyn pohjautuvan liikkeen havaitsemiseksi (Axis 2014a, 119). Kamera kuvaa maksimissaan HDTV 720p tasoista videokuvaa resoluutiolla 1280 x 800 (Axis 2014b, 3). Valittu M10-sarjan verkkokamera (Taulukko 5) mahdollistaa äänen käytön kaksisuuntaisesti. Kamera lähettää sekä vastaan ottaa ääntä tarpeen vaatiessa. Kyseinen ominaisuus mahdollistaa kameran käytön esimerkiksi opetustilanteissa. Opettajan ei siis tarvitse olla fyysisesti läsnä ohjatakseen harjoituksen kulkua.

Taulukko 5. Axiksen M10-sarjan verkkokameroiden vertailu (mukaillen Axis 2014b, 2)

Model	Resolution	Adjustable focus	Audio	PIR sensor White LED	I/O ports
M1004-W	1280 x 800	x			
M1013	800 x 600	x			
M1014	1280 x 800	x			
M1025	1920 x 1080	x			
M1033-W	800 x 600	x	x	x	x
M1034-W	1280 x 800	x	x	x	x
M1054	1280 x 800		x	x	x

Toinen ENVI-oppimisympäristöön valituista kameroista on Axiksen P3304. Merkittävin eroavaisuus M1034-W-kameraan on PTZ-ominaisuus, joka mahdollistaa kameran suoratoiston videokuvan liikuttamisen käyttöliittymässä. Laite on kooltaan suurempi: leveys 144 mm, korkeus 94 mm ja syvyys 132 mm. Kameran kuvaa HDTV 720p tasoista videokuvaa resoluutiolla 1280 x 800, sekä tarjoa mahdollisuuden kaksisuuntaiseen äänentoistoon. (Axis 2014c, 1–2.)

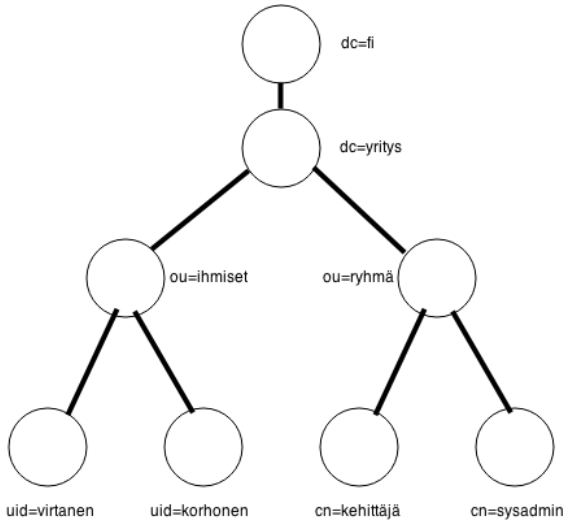


Kuvio 6. Kameroiden hallintasivu

Axiksen verkkokameroiden hallintasivulta on mahdollista säätää laitteen asetuksia (Kuvio 6). Sivulle päästään syöttämällä selaimen osoiteriville verkkokameran IP-osoite. Perusasetuksista voidaan määrittää kameran IP-osoite, langattoman verkon asetukset sekä aika- ja päivämääräasetukset. Käyttäjätunnusten hallinta onnistuu perusasetusten kautta. Videoasetusten kautta voidaan säätää kuvanlaatua koskevia arvoja sekä vaihtaa videonpakkausformaattia. Äänenhallinta sivuilla on mahdollista vaikuttaa mikrofonin sekä kameran kaiuttimen asetuksiin. Suoratoistoasetuksista pystytään määrittämään selainkohtaisesti, mitä katseluohjelmaa halutaan oletuksena käyttää. Tapahtuma asetusten sivulta voidaan määrittää, miten kamera reagoi esimerkiksi liikettä havaittaessa.

3.2 LDAP

LDAP (Lightweight Directory Access Protocol) on protokolla, joka mahdollistaa verkkohakemistopalvelujen käyttämisen. Hakemistopalvelulla tarkoitetaan esimerkiksi Active Directorya, joka varastoi informaation puurakenteen mukaisesti (Kuvio 7). Puurakenne on verrattavissa tietokoneen kovalevyn hakemistorakenteeseen. Hakemistopalvelun tarjoavalle palvelimelle lähetettävät palvelupyynnöt merkitään yksilöllisen tunnisteen mukaan. Yksilöllinen tunniste on vastike kovalevyn hakemistopolulle. (PHP 2015a.) LDAP-pyynnössä oleva yksilöllinen tunniste voisi olla esimerkiksi seuraava: *”cn=kehittäjä, ou=ryhmä, dc=yritys, dc=fi”.*



Kuvio 7. Hakemistopalvelun puurakenne

LDAP-protokollaa hyödynnetään usein käyttäjien tunnistuksessa. Tunnistusprosessi aloitetaan yhteyden muodostamisella hakemistopalveluun. Yhdistämiseen vaaditaan hakemistopalvelimen osoite ja portti. Onnistuneen yhteyden muodostamisen jälkeen muodostetaan sidos hakemistoon. Seuraava vaihe on suorittaa haku yksilöllisen tunnisteen avulla. Haussa etsitään annettujen tietojen perusteella vastaavia arvoja hakemistopalvelimelta. Haku palauttaa numeerisen arvon vastaavien merkintöjen määrän mukaisesti. Mikäli vastaavuuksia ei löydy syötettyjä käyttäjätunnuksia ei ole hakemistossa. Arvon ollessa suurempi kuin nolla tulokset noudetaan hakemistopalvelimelta taulukkoon. Hakujen suorittamisen jälkeen muodostetaan uudelleen sidos hakemistoon. Sidokseen lisätään sisäänkirjautumislomakkeeseen syötetyt tiedot. Sidoksen muodostamisen onnistuessa käyttäjä on syöttänyt tunnuksensa lomakkeeseen oikein, jolloin hänet kirjataan sisään palveluun. Esimerkin mukaisessa LDAP-sessiossa hyödynnetään taulukossa 6 esiteltyjä funktioita. Taulukko kattaa vain murto-osan tarjolla olevista LDAP-operaatioista.

Taulukko 6. Keskeisimmät ldap-funktiot (mukaillen PHP 2015a)

Funktio	Kuvaus
ldap_connect()	Yhteyden muodostaminen hakemistopalvelimeen. Parametreina palvelimen osoite ja portti.
ldap_bind()	Yhteyden sitominen hakemistoon. Parametreina yhteyden tiedot, asiakkaan tunnus ja salasana.
ldap_search()	Hakutoiminto tiedon hakemiseen hakemistopalvelusta. Parametreina yhteyden tiedot, yksilöllinen tunniste ja haun rajaus ehdot.
ldap_get_entries()	Haun mukaisten merkintöjen noutaminen hakemistopalvelusta taulukkoon. Parametreina yhteyden tiedot ja suoritetun haun tulos.
ldap_add()	Uuden tiedon lisääminen hakemistoon. Parametreina yhteyden tiedot, yksilöllinen tiedon tallentamisen tunniste ja uuden merkinnän tiedot.

ldap_delete()	Tiedon poistaminen hakemistosta. Parametreina yhteyden tiedot ja yksilöllinen tunniste, jossa poistettava tieto sijaitsee.
ldap_unbind()	Puretaan hakemistopalveluun luotu yhteys. Parametrina yhteyden tiedot.

3.3 Bcrypt ja PHP

Bcrypt on adaptiivinen salausmenetelmä, joka pohjautuu Blowfish-salauskirjoitusalgoritmiin. Tiivisteen luominen kyseisellä menetelmällä on hidasta verrattuna muihin tiivistys-algoritmeihin, mutta tuloksena saatu tiiviste on parempi. Tiivistämiseen kuluvaan aikaan voidaan vaikuttaa vaihtamalla generoitavan salasanan hintaa. Korkeampi hinta parantaa luotua tiivistettä, mutta hidastaa prosessia. (Hopkins 2011.) Yleisimpien tiivistealgoritmien heikkoutena pidetään Mooren lakia, jonka mukaan tietokoneisiin laitettavien transistoreiden määrää tuplaantuu kahden vuoden välein. Tietokoneet nopeutuvat ja pystyvät täten murtamaan entistä nopeammin tiivistettyjä salasanoja. Bcrypt sopeutuu Mooren lakiin tiivistämisessä käytetyn hinnan ansiosta. (Peterse 2011.)

Hinta on numeerinen arvo väliltä 04–31, joka toimii perustana Bcrypt-tiivisteen iteraatioiden määrässä (PHP 2015b). Matalammalla hinnalla tiivisteen generoiminen nopeutuu, mutta laadussa kärsitään. Korkeamman hinnan käyttäminen lisää tiivistyskertojen määrää. Hinnan vaihtaminen tiivistettä generoitaessa vaikuttaa koko lopputulokseen (Peterse 2011). Samasta merkkijonosta voidaan siis saada useita eri hajautusarvoja pelkän hinnan vaihtamisen avulla. Tämä vaikeuttaa huomattavasti alkuperäisen merkkijonon etsimistä tilanteessa, jossa vertaillaan tiivisteitä toisiinsa. Bcryptin luoma tiiviste on aina pituudeltaan 60 merkkiä (PHP 2015c).

PHP tarjoaa suoran tuen Bcrypt-tiivisteen käytölle. Käytettävissä on kolme funktiota, joiden avulla saadaan suoritettua vaadittavat toimenpiteet merkkijonojen

käsittelemiseksi. Edellä mainitut funktiot ovat: *password_hash(), crpyt() ja password_verify(). Crypt()*-funktion käyttöä ei suositella, koska sen avulla generoidut tiivisteet ovat heikompia verrattuna *password_hash()*-funktioon (PHP 2015b).

```
string password_hash ( string $password , integer $algo [, array
$options ] )
```

Esimerkkikoodi 1. Merkkijonon tiivistäminen *password_hash()*-funktiolla (PHP 2015c)

Funktion *password_hash()* avulla mahdollistetaan merkkijonojen tiivistäminen PHP-ohjelmointikielellä (Esimerkkikoodi 1). Parametreina funktiolle syötetään tiivistettävä merkkijono, käytettävä tiivistysalgoritmi ja mahdolliset lisävaihtoehdot. Funktiolle ilmoitetaan salasanan tiivistyksessä käytettävän Bcrypt-algoritmia korvaamalla *$algo* parametri arvolla *PASSWORD_BCRYPT*. Bcryptin tukemat lisävaihtoehdot ovat hinta ja suolaus. (PHP 2015c.) Suolauksella tarkoitetaan merkkijonoa, joka liitetään yhteen annetun salasanan kanssa. Sen tarkoituksena on hankaloittaa esimerkiksi sanakirjahyökkäysten tekemistä. *Password_hash()*-funktio hoitaa suolauksen generoimisen automaattisesti, mikäli arvo jätetään tyhjäksi. Suositeltavaa on jättää suolauksen generoiminen funktiolle, sillä tällöin jokaiselle salasanalle luodaan aina uusi suolaus tiivistyksen yhteydessä. (PHP 2015c.)

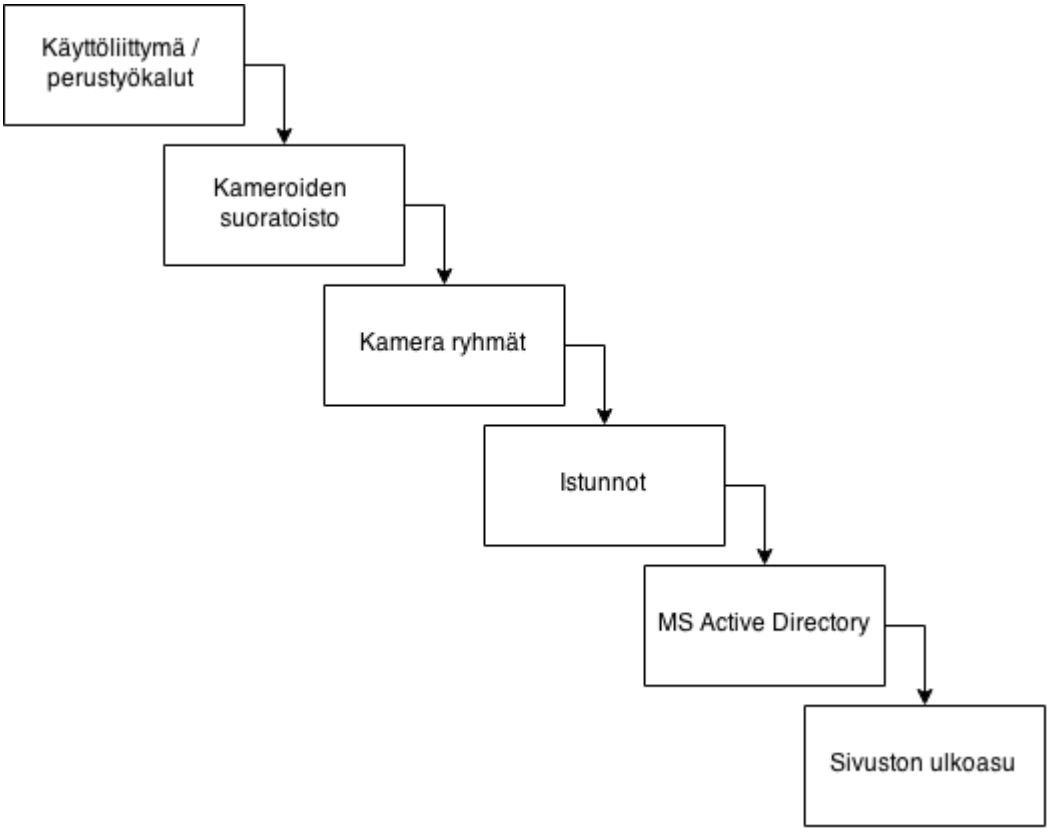
```
boolean password_verify ( string $password , string $hash )
```

Esimerkkikoodi 2. Tiivisteiden vertaaminen keskenään *password_verify ()*-funktiolla (PHP 2015d)

Funktiota *password_verify()* hyödynnetään tarkistettaessa *password_hash()*-funktiolla luotuja tiivisteitä. Parametreina funktio vaatii käyttäjän syöttämän salasanan ja tietokantaan tallennetun tiivisteen(Esimerkkikoodi 2). (PHP 2015d.) *Password_verify()*-funktio helpottaa käyttäjien tunnistamista sisäänkirjautumislomakkeen yhteydessä. Tiivisteiden vastatessa toisiaan funktiosta palautettu arvo on tosi, jonka jälkeen käyttäjälle voidaan esimerkiksi luoda aktiivinen istunto sivustolle.

4 SIVUSTON SUUNNITTELU

Työn tavoitteena oli luoda www-sovellus, jota käytetään ENVI-tilojen opetuksen ja harjoitteiden seurantaan. Tiloissa on lukuisia IP-kameroita, joiden kautta opetuksen seuraaminen etänä on mahdollista. Kamerat ovat asennettuina tiloihin, mutta ongelmana on että kuka tahansa pääsee seuraamaan opetusta halutessaan. Toimivan käyttöliittymän lisäksi tarkoituksena on toteuttaa ympäristöstä turvallinen, jotta opiskelijat saavat suorittaa harjoitteensa tarvittaessa ilman pelkoa ulkopuolisista katselijoista.

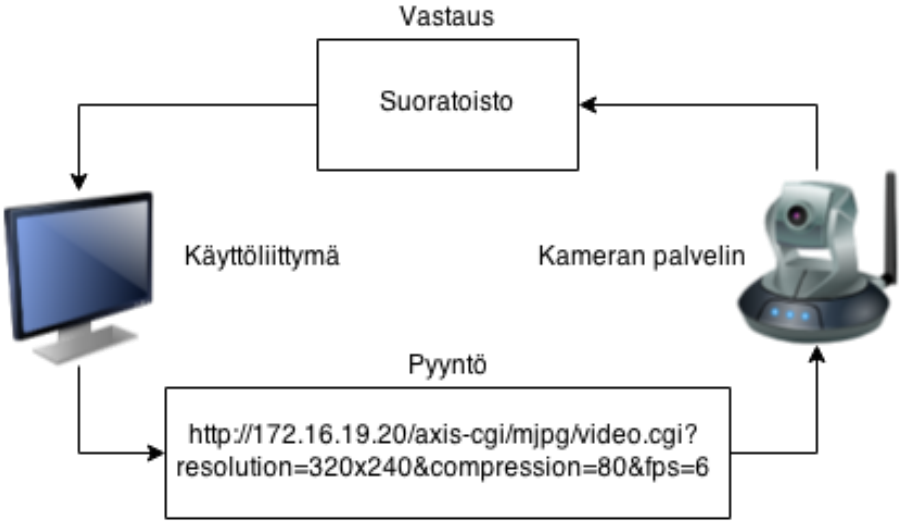


Kuvio 8. Työvaiheet siirtymineen

Suunnitteluvaiheessa sivuston sisältö jaettiin toteutuksen mukaan omiin kokonaisuuksiin, jotta kehityksessä on helppo ottaa työn alle yksi kokonaisuus kerrallaan. Kuviossa 8 on kuvattu työvaiheet toteutettavien kokonaisuuksien mukaan. Käyttöliittymän toteutuksessa luodaan sivustolle runko, johon sivuston ominaisuudet saadaan liitettyä kehityksen edetessä. Perustyökaluja sivustolla ovat huoneiden

ja kameroiden hallinta. Huoneella tarkoitetaan ENVI-tiloissa sijaitsevaa osaa, esimerkiksi synnytysosastoa. Työkalujen avulla on mahdollista huoneiden ja kameroiden luominen, muokkaaminen sekä poistaminen. Huoneet saattavat sisältää useampia kameroita eripuolilla tilaa. Kyseiset ryhmät on otettava huomioon sovelluksen työkalujen kehityksessä, jotta kameroiden hyödyntäminen olisi mahdollisimman vaivatonta.

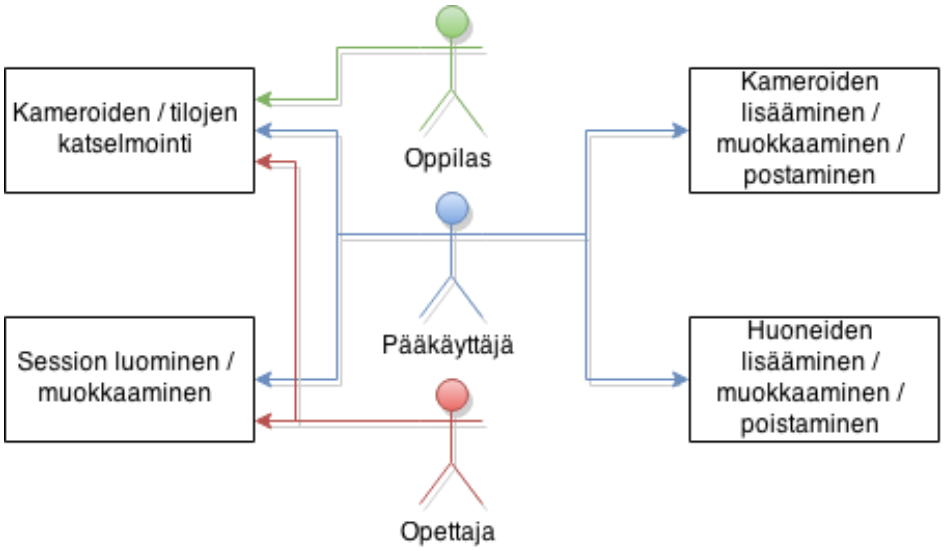
Huone sisältää maksimissaan neljä kameraa. Kameroiden suoratoisto saadaan käyttöliittymän näkymiin lähettämällä HTTP-pyyntö halutulle kameralle. Palvelupyyntö sisältää kameran IP-osoitteen ja parametrit koskien suoratoiston tyyppiä sekä laatua. Käyttöliittymän näkymän täytyy latautua nopeasti hitaammillakin yhteyksillä. Asettamalla pyydettävän suoratoiston laatu mahdollisimman alhaiseksi saadaan näkymän latausnopeutta optimoitua. Kuvan laadun täytyy olla katsojaa miellyttävä, joten pakkaamiselle ja kuvataajuudelle asetettavat arvot eivät saa olla liian alhaiset. Kuvio 9 havainnollistaa käyttöliittymän sekä kameran välisen kommunikaation suoratoistoa pyydettäessä.



Kuvio 9. Käyttöliittymän ja kameran välinen kommunikointi

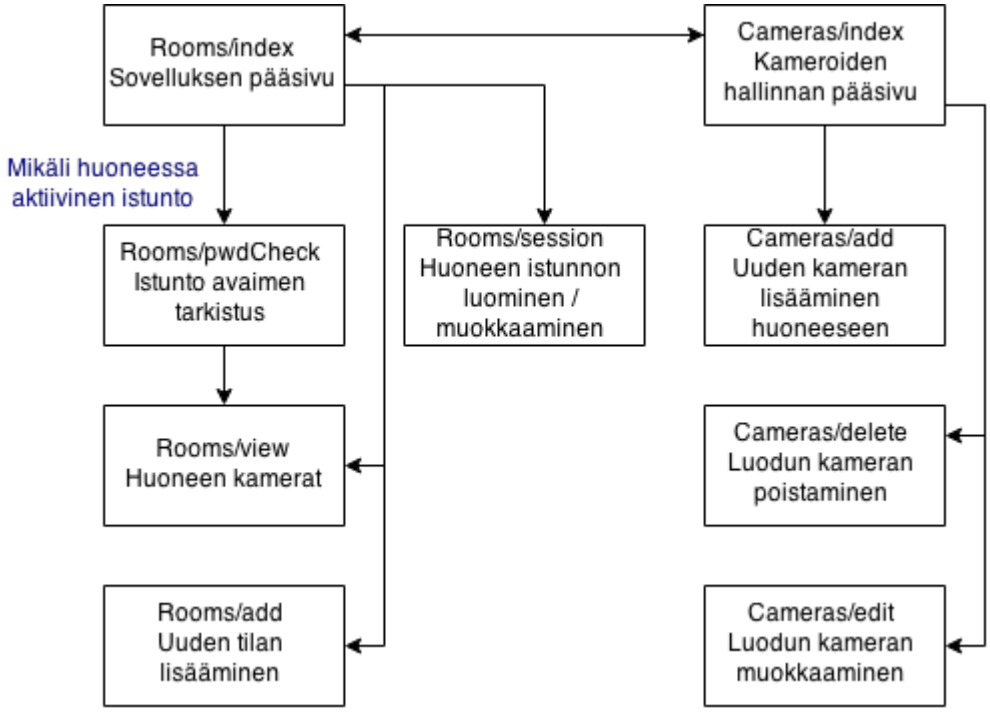
Sivuston käyttäjät ovat pääasiassa Lapin Ammattikorkeakoulun sosiaali- ja terveysalan henkilökunta sekä opiskelijat. Organisaatiolla on käytössä Microsoftin tarjoama Active Directory, jonka kautta hoidetaan henkilöiden sisään kirjautumi-

nen esimerkiksi korkeakoulun koneille sekä sähköpostiin. Sovellus sisältää osioita, joihin kenellä tahansa ei tulisi olla pääsyä, kuten istuntojen luominen sekä kameroiden ja huoneiden hallinta. LDAP-protokollan avulla voidaan hyödyntää Active Directory – hakemistopalvelua käyttäjien tunnistuksessa. Active Directory mahdollistaa henkilön sisäänkirjautumisen hänen olemassa olevilla tunnuksilla, joten se on tarjolla olevista ratkaisuista käyttäjäystävällisin. Sovelluksessa on kolme käyttäjäryhmää: oppilas, opettaja ja pääkäyttäjä. Ryhmät sekä niille annetut oikeudet näkyvät kuviosta 10.



Kuvio 10. Sivuston käyttäjäryhmät ja niiden oikeudet

Oppilailla on pääsy vain kameroiden sekä tilojen katselmointiin, joten heidän ei tarvitse kirjautua sisään palveluun käyttääkseen sitä. Kameroiden katseluoikeutta voi rajoittaa luomalla halutulle tilalle istunnon. Istunnon tekemisen hoitaa opettaja tai pääkäyttäjä. Tällöin huoneeseen pääsee vain syöttämällä luodun istuntoavaimen, joka välitetään tarvittaessa halutuille käyttäjille. Istuntojen tarkoituksena on rajoittaa pääsyä kameroiden suoratoistoon opetustilanteen aikana. Istunto on voimassa määritetyn ajan, jonka jälkeen tila on jälleen kaikille avoin. Huoneiden sekä kameroiden hallintaan koskeviin sivuihin on vain pääkäyttäjällä oikeus. Tällä pyritään minimoimaan mahdolliset vahingon aiheuttajat, joita voi syntyä useamman henkilön tehdessä muutoksia huoneisiin tai kameroihin käyttöliittymän kautta.



Kuvio 11. Web-sovelluksen alustava sivustokartta

Ylläolevassa kuviossa 11 on havainnollistettu sivuston rakenne sekä kerrottu sivujen tarkoitus lyhyesti. Pääsivulle sijoitetaan ENVI-tilan karttapohja, jonka päälle sijoitetaan huoneet kameraryhmineen. Kyseiseltä sivulta on pääsy huoneiden suoratoiston katsomiseen sekä tilojen hallintaa koskeviin ominaisuuksiin. Käyttöliittymän ylätunnisteen navigaatiovalikosta on mahdollista siirtyä kameroiden hallintaan koskeville sivuille.

5 SIVUSTON TOTEUTUS

5.1 Käyttöliittymä

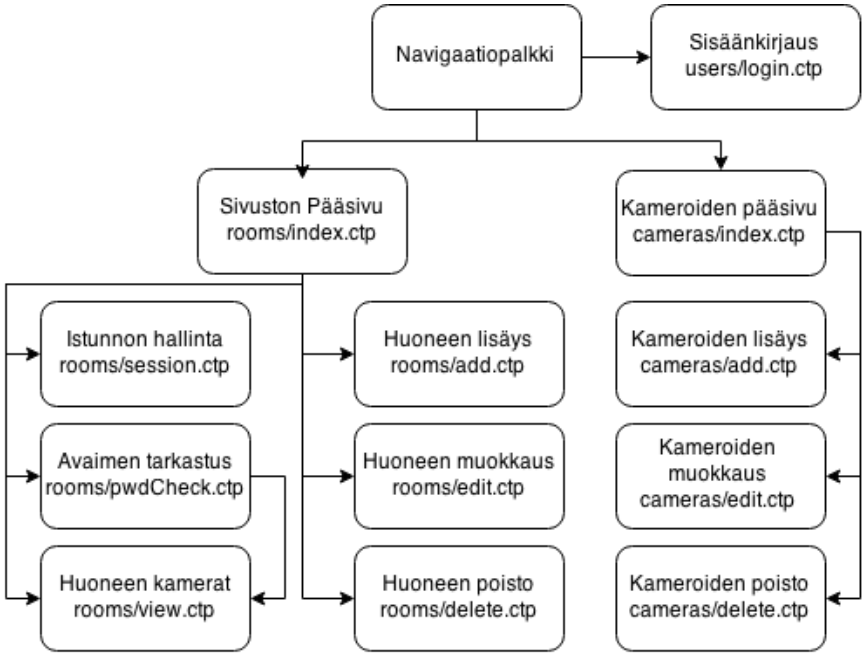
Toteutuksen alkaessa ensimmäinen vaihe oli Xampp-kehitysympäristön asentaminen joka sisältää palvelimen, tietokannan ja PHP:n. Sivusto toteutetaan käyttäen CakePHP-sovelluskehystä. Asennusprosessi on erittäin yksinkertainen: tarvitaan vain kopio CakePHP-paketista, joka sijoitetaan kehitysympäristön *htdocs*-kansioon. Käyttöliittymässä hyödynnetään Bootstrap-sovelluskehystä, joka sisältää kattavan työkaluvalikoiman. Sen avulla saadaan toteutettua sivuston ulkoasu.

CakePHP:n paketti sisältää sivustopohjan, jota mukaillen aloitettiin käyttöliittymän toteutus. Pohjasta karsittiin pois kaikki ylimääräinen mitä sivustolla ei tulla tarvitsemaan. Jäljellä olevaan runkoon lisätään tarvittavat elementit (DIV-tagit yms.) sekä muokataan olemassa olevien elementtien luokkia vastaamaan bootstrapin tarjoamia. Esimerkkikoodi 3 sisältää käyttöliittymän rungon, jonka pohjalle sivuston toteutus aloitettiin.

```
<!DOCTYPE html>
<html>
<head>
    <title>
        <!-- Browser tab title -->
    </title>
</head>
<body>
    <!-- Header -->
    <nav class="navbar navbar-inverse" role="navigation">
        <div class="container">
            <div class="nav-header">

            </div>
            <div id="navbar" class"collapse navbar-collapse">

                <!-- Navigation links on the site -->
                <ul class="nav navbar-nav">
                </ul>

                <!-- Language selection and login/logout -->
                <ul class="nav navbar-nav navbar-right">
                </ul>
            </div>
        </div>
    </nav>
    <!-- End of Header -->
    <div class="container">
        <!-- Content -->
        <div id="content">
            <?php echo $this->Session->flash(); ?>
            <?php echo $this->fetch('content'); ?>
        </div>
        <!-- End of Content -->
        <!-- Footer -->
        <div id="footer">
        </div>
        <!-- End of Footer -->
    </div>
</body>
</html>
```

Esimerkkikoodi 3. Käyttöliittymän runko

Käyttöliittymä koostuu kolmesta osasta: ylätunniste (header), sisältö (content) sekä alatunniste (footer). Ylätunniste sisältää navigoinnin, kielivalinnan sekä sisäänkirjautumisen. Navigaatioelementin kautta päästään huoneiden ja kameroiden pääsivuille. Huoneiden sekä kameroiden pääsivu sisältää linkit kuviosta 12 nähtäviin näkymiin. Peruskäyttäjät pääsevät pelkästään huoneiden pääsivulle, avaimen tarkistus sivulle ja huoneen kameroiden katselmointi sivulle. Pääsy sivuston muihin näkymiin vaatii sisäänkirjautumisen. Näkymien sisältö sijaitsee ctp-tiedostoissa, jotka kaiutetaan käyttöliittymän sisältöelementtiin komennolla *$this->fetch('content').*



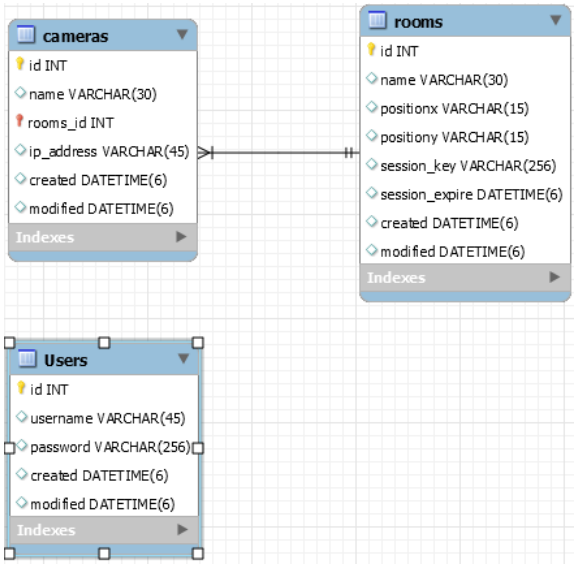
Kuvio 12. Sivuston navigaatiorakenne

Käyttöliittymän värit mukailevat Lapin AMKin graafisen ohjeistuksen määrittämää värikarttaa (Lapin AMK 2013, 6). Alapuolella oleva kuvio 13 sisältää sivustolla käytettävän päävärin, joka on murretun punainen. Kyseistä väriä sävyineen käytetään pääasiassa ylätunnisteessa, otsikoissa sekä lomakkeissa.

Kuvio 13. Lapin Ammattikorkeakoulun värikartta (Lapin AMK 2013, 6)

5.2 Tietokanta

Sivuston kehityksessä käytetään MySQL-tietokantaa, joka tulee asennetun kehitysympäristön mukana. Kuviosta 14 nähdään tietokannan taulut sarakkeineen. Kamera- ja huonetaulujen välinen relaatio on ”yhden suhde moneen”. Tämä tarkoittaa sitä, että yhdessä huoneessa voi olla monta kameraa. Relaation ilmoittaminen CakePHP:lle tapahtuu huoneen ja kameran malli-luokassa. Huoneen mallille ilmoitetaan sen sisältävän useita kameroita seuraavan muuttujan avulla: *public $hasMany = ’Cameras’*. Kamerat kuuluvat huoneeseen, joten mallille ilmoitetaan relaatiosta muuttujalla: *public $belongsTo = ’Room’*.



Kuvio 14. Tietokannan käsitemalli

Käyttäjän luodessa uutta huonetta tietokantaan tallennetaan tilan nimi sekä sen sijainnin koordinaatit. Session luomisen yhteydessä huoneen tietoihin lisätään sessioavain sekä session päättymisajankohta. Kameraa luodessa tauluun tallennetaan nimi, joka viittaa kameran sijaintiin huoneessa. Tietokantaan tallennetaan myös kameran IP-osoite sekä huone, jonne laite on sijoitettu. Käyttäjätaulu sisältää sivuston pääkäyttäjien tiedot. Pääkäyttäjillä on oikeudet sivuston kaikkiin toiminnallisuuksiin. Tietokantataulujen kentät *created* sekä *modified* ovat CakePHP:n asettamia suosituksia, joihin tallennetaan milloin kirjaus on luotu sekä milloin sitä on viimeksi muokattu.

```
public $default = array(
    'datasource' => 'Database/Mysql',
    'persistent' => false,
    'host' => 'localhost',
    'login' => 'root',
    'password' => '',
    'database' => 'cameraapplication',
    'prefix' => '',
    //'encoding' => 'utf8',
);
```

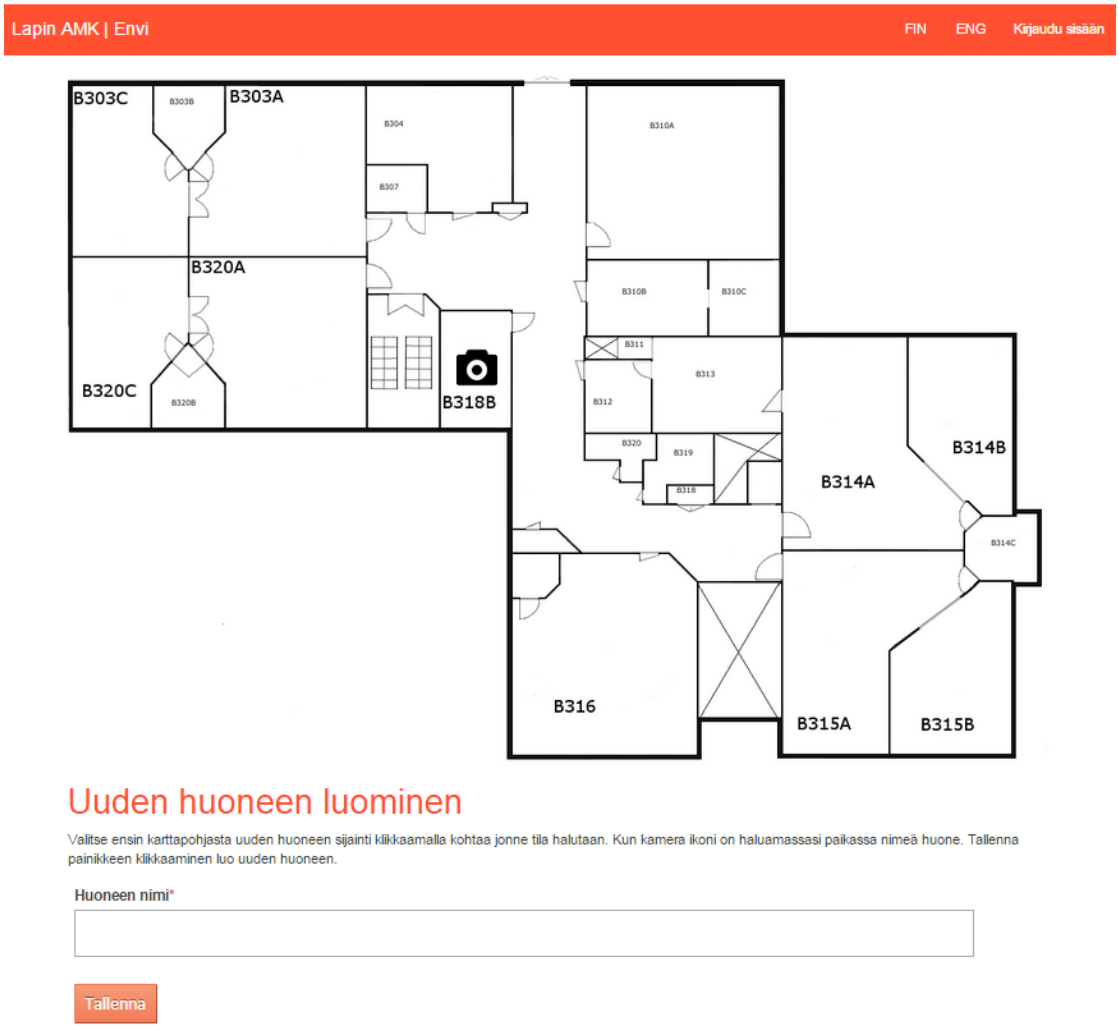
Esimerkkikoodi 4. CakePHP-tietokanta-asetukset

Taulujen alustamisen jälkeen muodostetaan yhteys CakePHP:lla tietokantaan. Kyseinen toimenpide suoritetaan täyttämällä *database.php*-tiedostoon yhteyden muodostamiseksi vaadittavat tiedot (Esimerkkikoodi 4). Yhteyden toimimiseksi tarvitaan käytettävän tietokannan tyyppi, isännän osoite, tietokannan tunnukset sekä tietokannan nimi.

5.3 Pääkäyttäjän työkalut

Käyttöliittymän sekä tietokannan valmistuttua aloitetaan perustyökalujen ohjelmoiminen, joita pääkäyttäjä tarvitsee sisällön hallintaan. Ensimmäisenä lähdetään toteuttamaan huoneiden hallintaan liittyviä toimintoja. Huoneita täytyy pystyä luomaan, muokkaamaan sekä poistamaan. MVC-arkkitehtuurin mukaisesti luodaan huoneelle malli, tarvittavat näkymät sekä käsittelijä. RoomsController-

käsittelijään (Liite 1) ohjelmoidaan näkymien toiminnallisuudet, esimerkiksi kuinka käyttäjän käskyihin reagoidaan näkymissä.



Kuvio 15. Näkymä uuden huoneen lisäämiselle

Uuden huoneen lisääminen tapahtuu kuvion 15 mukaisessa lomakkeessa. Ensimmäiseksi käyttäjä valitsee huoneen sijainnin klikkaamalla karttapohjaa. Valittuun paikkaan luodaan kameraryhmä, johon myöhemmin lisätään kameroita erillisellä työkalulla. Käyttäjän hiiren kilkkaaminen karttapohjaelementin sisällä aktivoi *mousePosition()* JavaScript-tapahtuman (Esimerkkikoodi 5).

```
function mousePosition(event)
{
    //Get the mouse click position in the div
    var x = event.layerX;
    var y = event.layerY;

    //Save the values into form fields
    document.getElementById('RoomPositionx').value = x;
    document.getElementById('RoomPositiony').value = y;

    //Camera image from the view
    var icon = document.getElementById('draggable');

    //Change images position in view
    icon.style.left = x + 'px';
    icon.style.top = y + 'px';
}
```

Esimerkkikoodi 5. Hiiren sijainnin kaappaaminen klikatessa

Muuttujiin x ja y tallennetaan klikkauksen sijainti hyödyntäen JavaScriptin layerX sekä layerY-ominaisuuksia. Sivu sisältää kaksi piilotettua tekstinsyöttökenttää, joihin yllä mainittujen muuttujien arvot asetetaan. Kamera-kuvakkeen sijainti vaihdetaan klikattuun kohtaan esimerkkikoodin 5 lopussa. Viimeisenä vaiheena on nimetä tila. *Tallenna*-painikkeen klikkaaminen tallentaa huoneen tietokantaan, mikäli lomakkeen tiedot läpäisevät mallin validoinnin. Mallin tarkastuksessa katsotaan ovatko kentät täytetyt ja sisältääkö merkkijono kiellettyjä merkkejä. Tietokantaan tallennus tapahtuu käsittelijän *add()*-funktiossa (Esimerkkikoodi 6).

```
public function add()
{
    if($this->request->is('post'))
    {
        $this->Room->create();
        if($this->Room->save($this->request->data))
        {
            $this->Session->setFlash(__('roomSaved'), 'flash_success');
            return $this->redirect(array('action' => 'index'));
        }
        $this->Session->setFlash(__('roomFailure'), 'flash_failure');
    }
}
```

Esimerkkikoodi 6. Käsittelijän *add()* -funktio

Käsittelijän *add()*-funktiossa tarkastetaan, klikkaako käyttäjä lomakkeen tallenna-painiketta (Esimerkkikoodi 6). Seuraavaksi luodaan uusi huone. Tietokantaan tallennuksen onnistuessa käyttäjä siirretään takaisin pääsivulle. Siirron jälkeen käyttäjälle ilmoitetaan tallennuksen onnistumisesta. Mikäli tallennus epäonnistuu, esimerkiksi lomakkeen tiedot eivät läpäise mallin validointia, ilmoitetaan siitä käyttäjälle.

Kaikki huoneet

Nimi	Aktiivinen istunto	Istunnon hallinta	Huoneen hallinta
B303A	Ei istuntoa	Istunnon luominen huoneelle	Muokkaa Poista
B314A	Ei istuntoa	Istunnon luominen huoneelle	Muokkaa Poista
B318B	Ei istuntoa	Istunnon luominen huoneelle	Muokkaa Poista
B320C	2015-03-02 23:39:00	Muokkaa istuntoa	Muokkaa Poista

Kuvio 16. Huoneiden hallintataulukko käyttöliittymässä

Huoneiden hallintasivuille pääsee kuviossa 16 esitetyn taulukon linkkien kautta. Tilan muokkaamiseen käytetään pohjana samaa näkymää kuin huonetta luodessa. Näkymässä sijaitsevaa lomaketta täytyy kuitenkin hieman muokata, jotta saadaan haluttu lopputulos. Lisätessä lomakkeeseen piilotetun kentän *id*, CakePHP olettaa kyseessä olevan tietokantaan tallennetun tiedon muokkaamista koskeva tapahtuma. Käsittelijän *edit($id)*-funktio tarkistaa, onko haetulla tunnisteella dataa tietokannassa. Mikäli haetulla tunnisteella on olemassa oleva kirjaus, käsittelijä siirtää tiedon kannasta näkymässä olevaan lomakkeeseen (Esimerkkikoodi 7).

```
public function edit($id = null) {
    if (!$id) {
        throw new NotFoundException(__('Invalid room'));
    }

    $room = $this->Room->findById($id);
    if (!$room) {
        throw new NotFoundException(__('Invalid room'));
    }

    if ($this->request->is(array('post', 'put'))) {
        $this->Room->id = $id;
        if ($this->Room->save($this->request->data)) {
            $this->Session->setFlash(__('roomEditSaved'));
            return $this->redirect(array('action' => 'index'));
        }
        $this->Session->setFlash(__('roomFailure'));
    }

    if (!$this->request->data) {
        $this->request->data = $room;
    }
}
```

Esimerkkikoodi 7. Funktio huoneen muokkaamisesta käsittelijässä

Käsittelijän *muokkaa*-funktio sisältää parametrin id, jonka avulla saadaan muokattua oikea huone. Parametrin arvo saadaan klikattaessa kuvion 16 *muokkaa*-

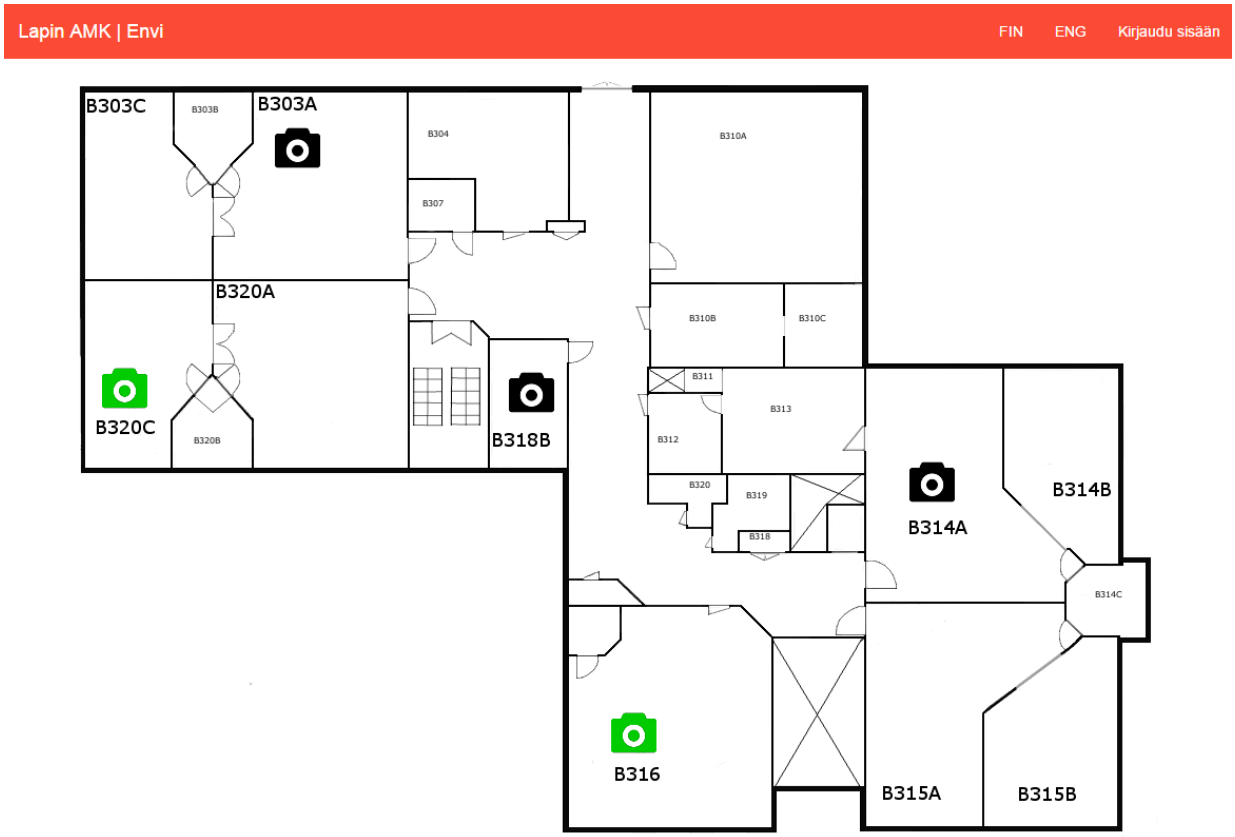
hyperlinkkiä. Linkki sisältää tiedon siitä, mihin käsittelijän toimintoon napsauttaessa ohjataan sekä mikä on muokattavan huoneen tunniste tietokannassa. Käsittelijän *edit()*-funktiossa siirretään huoneen tiedot näkymän lomakkeeseen. Muokkaamisen jälkeen päivitetty data tallennetaan tietokantaan.

Huoneen poistaminen tapahtuu päänäkymässä sijaitsevan hallintataulukon kautta (Kuvio 16). Taulukon linkki sisältää käsittelijässä kutsuttavan funktion ja poistettavan huoneen tunnisteen. Linkin klikkaaminen aktivoi JavaScript-varoituksen, joka kysyy käyttäjältä varmistuksen huoneen poistamiseksi. Hyväksyessään poiston siirrytään käsittelijän *poista*-funktioon, joka hävittää tietokannasta valitun huoneen.

Kuvio 17. Näkymä uuden kameran lisäämiselle tietokantaan

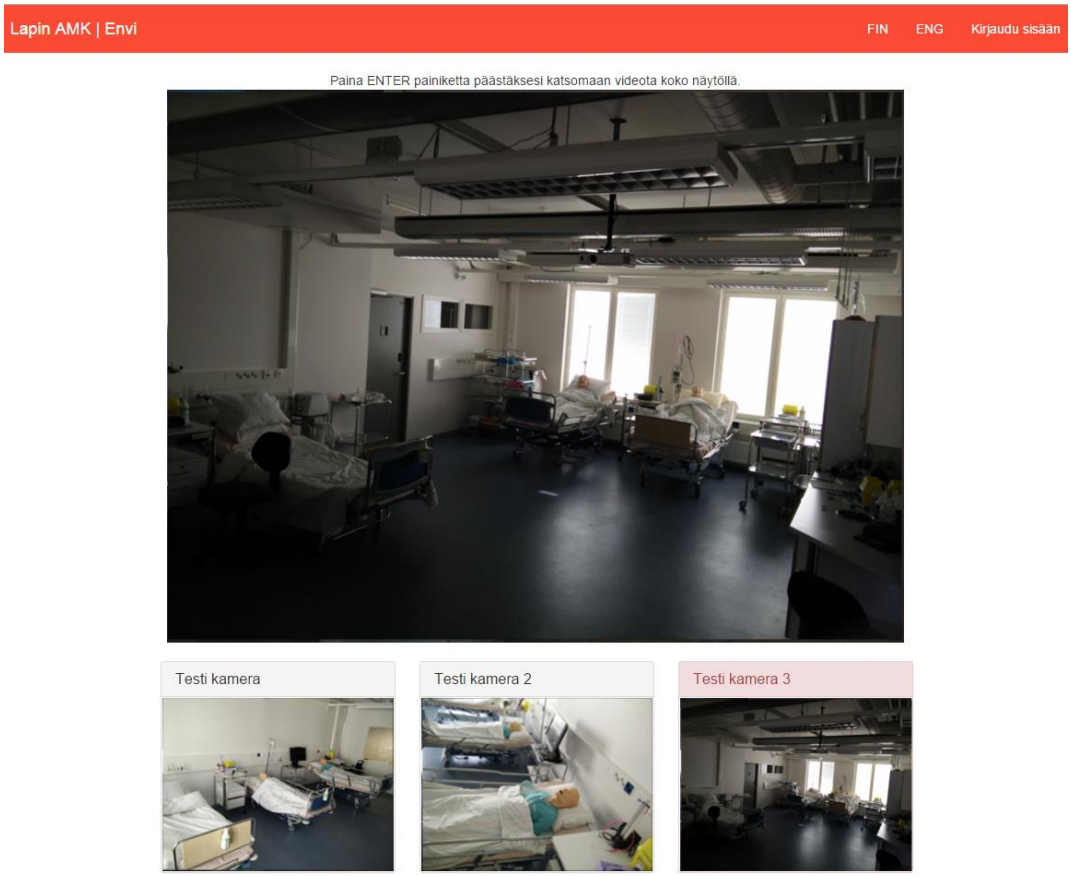
Kameroiden lisääminen huoneisiin tapahtuu kuvion 17 mukaisella lomakkeella. Kenttiin syötetään kameran nimi ja IP-osoite, joka on määritettynä laitteelle. Huoneen valinta tapahtuu alasveto-valikosta, joka sisältää kaikki tietokannassa olevat tilat. Tietokanta osiossa selitetty mallien relaatio mahdollistaa huoneiden kirjausten hakemisen tietokannasta kameroiden käsittelijässä sekä datan esittämisen näkymässä. Kameroiden lisäämisen, muokkaamisen ja poistamisen toiminnot vastaavat rakenteeltaan huoneitten käsittelijää sekä näkymiä.

5.4 Kameroiden näkymät



Kuvio 18. Sivuston päänäkymä huoneineen

Kameroiden suoratoistoa pääsee katsomaan pääsivun kautta klikkaamalla karttapohjalla sijaitsevia kamera-kuvakkeita (Kuvio 18). Karttapohjassa sijaitsevat mustat ikonit ovat vapaille huoneille, joten erillistä salasanaa ei vaadita. Vihreät ikonit kertovat käyttäjälle huoneessa olevan meneillään istunnon, joka vaati salasanan sisäänpääsynä. Huoneen näkymä kameroineen on esitetty kuviossa 19.



Kuvio 19. Huoneen näkymä kameroineen

Kameroiden esikatseluelementit sijaitsevat rivissä, jotta huoneen yleiskuva on helposti nähtävillä (Kuvio 19). Esikatseluikkunan klikkaaminen avaa videokuvan isompana kameroiden yläpuolella olevaan elementtiin. Valitun IP-kameran kuva on mahdollista saada näkymään koko näytöllä painamalla Enter-näppäintä. Esimerkkikoodi 8 havainnollistaa, kuinka kameroiden esikatselukuvat saadaan esiin näkymässä.

```
foreach ($cameras as $c)
{
    echo '<div class="col-md-3 col-centered">';
        echo '<div class="panel panel-default">';
            echo '<div class="panel-heading">';
                echo '<h4 class="panel-title">'.$c['Camera']['name'].'</h4>';
            echo '</div>';
            echo '<div class="panel-body">';
                echo $this->Html->image('http://viewer:password@'.$c["Camera"]["
                    ip_address"].'/axis-cgi/mjpg/video.
                    cgi?resolution=320x240&compression=80&fps=6', array('class' =>
                    'img-responsive', 'alt' => $c['Camera']['ip_address']));
            echo '</div>';
        echo '</div>';
    echo '</div>';
}
```

Esimerkkikoodi 8. Suoratoiston esittäminen näkymässä

Huoneen käsittelijässä suoritetaan tietokantakysely, joka hakee valitun tilan tunnisteen perusteella sinne kuuluvat kamerat (Liite 1). Tulos tallennetaan *cameras*-muuttujaan, joka käydään läpi näkymän foreach-silmukassa. Kameran palvelimelle lähetetään HTTP-pyyntö kuvan tunnisteen lähteessä, joka sisältää verkkokameran IP-osoitteen ja cgi-pyynnön. Cgi-pyynnössä määritetään, mitä kamera lähettää vastauksena käyttöliittymään. Esimerkkikoodissa 8 pyydetään Motion JPEG -kuvaa resoluutiolla 320 x 240 pakattuna 80 % alkuperäistä pienemmäksi, ja fps-arvoksi on asetettu 6.

```
$(document).ready(function() {
    $('.panel > .panel-body > img').click(function() {
        //Save the clicked images alternate text value into variable
        var ipAddress = $(this).attr('alt');
        //Change videoPlayer divs source attribute
        $('#videoPlayer').attr('src', 'http://viewer:password@' + ipAddress + '/axis
            -cgi/mjpg/video.cgi?resolution=800x600&compression=25&fps=25');
    });
    //Change the active preview divs color
    $('.panel').click(function() {
        $('.panel').removeClass('panel-danger');
        $('.panel').addClass('panel-default');
        $(this).addClass('panel-danger');
    });
});
```

Esimerkkikoodi 9. Valitun esikatselukuvan esittäminen suurempana

Kameroiden katselunäkymä sisältää div-elementin, joka sivun auetessa on tyhjä. Esikatselukuvakkeen klikkaaminen asettaa valitun elementin aktiiviseksi vaihtamalla DIV-elementin luokkaa (Esimerkkikoodi 9). Aktiivisen kameran IP-osoite otetaan talteen kuva-elementin attribuutista. Tyhjän elementin sisältämälle kuvatagille annetaan lähteeksi aikaisemmin tallennettu IP-osoite, sekä lisätään cgi-pyyntö haluttuine arvoineen. Kameran palvelimelta pyydettävä videomateriaali halutaan resoluutiolla 800 x 600 pakattuna 25 % alkuperäisestä sekä kuvataajuudeksi 25 kuvaa sekunnissa.

```
//Find the video player element fromt the view
var videoElement = document.getElementById('videoPlayer');

function toggleFullScreen()
{
    //Check if not in full screen mode
    if(!document.mozFullScreen && !document.webkitFullScreen)
    {
        //Toggle full screen
        if(videoElement.mozRequestFullScreen)
        {
            videoElement.mozRequestFullScreen();
        }
        else
        {
            videoElement.webkitRequestFullScreen(Element.ALLOW_KEYBOARD_INPUT);
        }
    }
    else
    {
        //Cancel full screen
        if(document.mozCancelFullScreen)
        {
            document.mozCancelFullScreen();
        }
        else
        {
            document.webkitCancelFullScreen();
        }
    }
}
//Event listener that calls toggleFullScreen function
//when enter is pressed by the user
document.addEventListener('keydown', function(e) {
    if(e.keyCode == 13) {
        toggleFullScreen();
    }
}, false);
```

Esimerkkikoodi 10. Suoratoiston katsominen koko näytöllä

Suoratoistoa voi katsoa halutessaan koko näytöllä. Enter -painikkeen painaminen aktivoi esimerkkikoodi 10 mukaisen *toggleFullScreen()* -funktion, joka suurentaa aktiivisen kameran videokuvan kattamaan koko näytön. Funktiossa tarkistetaan, onko videokuva sillä hetkellä suurennettuna. Tarkistuksesta palautetun arvon ollessa epätosi, aktivoidaan kokonäytön tila. Poistuminen takaisin käyttöliittymän näkymään tapahtuu painamalla ESC-näppäintä.

5.5 LDAP ja sisäänkirjautuminen

Kuvio 20. Käyttäjän tunnistukseen käytettävä lomake

Sivustolla käytetään LDAP-protokollaa henkilökunnan tunnistuksessa. Tapaukset, jolloin esimerkiksi opettajan tulee kirjautua sisään palveluun, ovat istunnon luominen ja muokkaaminen. Sisäänkirjautuminen tapahtuu lomakkeella (Kuvio 20), joka sisältää käyttäjätunnusten tarkistamiseen vaaditut kentät. Users-käsittelijän *login()*-funktio (Esimerkkikoodi 11) huolehtii käyttäjän tunnistamiseen liittyvistä toimenpiteistä.

```
public function login()
{
    //Mikäli käyttäjä on jo kirjautuneena sisään ohjaa pääsivulle
    if($this->Session->read('Auth.User'))
    {
        $this->Session->setFlash(__('alreadyLoggedIn'), 'flash_success');
        $this->redirect(array('controller' => 'users', 'action' => 'index'));
    }
    //Sisään kirjautumis lomake ei ole tyhjä
    elseif(!empty($this->request->data))
    {
        //Kirjasto joka huolehtii tunnistautumisen Active Directorya vasten
        $ldap = new ldap;
        if($ldap->auth($this->Auth->request->data['User']['username'],
            $this->request->data['User']['password']))
        {
            //Vapauta salasana kentän arvo jottei sitä tallenneta sessio muuttujaan
            unset($this->request->data['User']['password']);
            //Käyttäjän kirjaaminen sisään
            $this->Auth->login($this->request->data['User']);
            $this->Session->setFlash(__('loginSuccess'), 'flash_success');
            return $this->redirect($this->Auth->redirectUrl());
        }
        else
        {
            $this->Session->setFlash(__('loginFailed'), 'flash_failure');
        }
    }
}
```

Esimerkkikoodi 11. UserController *login()* -funktio

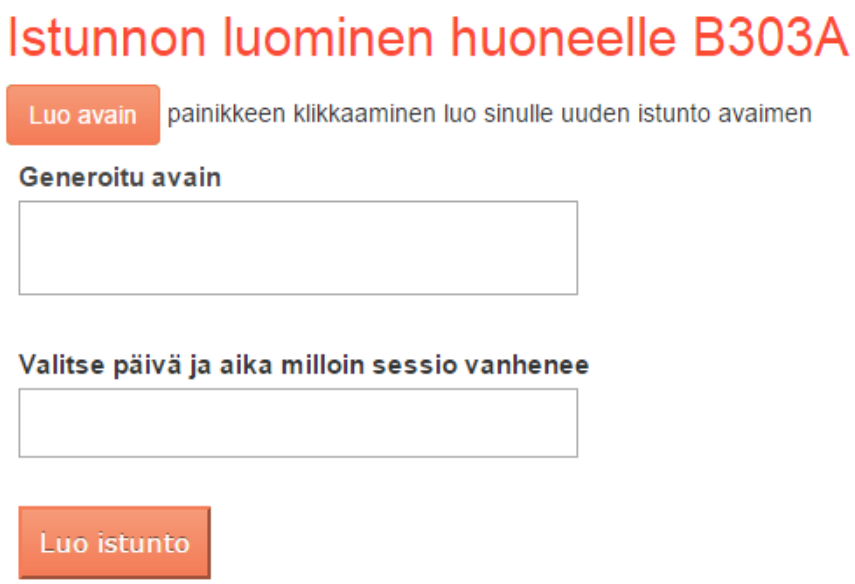
Käyttäjän tunnistamiseen hyödynnetään LDAP-verkkoprotokollaa, joka hakee käyttäjän tiedot Active Directory -hakemistopalvelusta. Ldap-luokka (Liite 2) sisältää *auth()*-funktion, jonka parametreina ovat käyttäjänimi sekä salasana. Sisäänkirjautumisessa käyttäjän lomakkeeseen syöttämät tiedot lähetetään *login()*-funktion kautta ldap-luokalle. Tietojen hakeminen aloitetaan muodostamalla yhteys käyttäjätietokantaan antamalla palvelimen osoite ja portti. *Ldap_bind()*-funktiossa (Liite 2) tarkistetaan, onko annetuilla tiedoilla olemassa vastaavaa käyttäjää hakemistopalvelussa. Mikäli palautettava arvo on tosi, katsotaan kuuluuko objekti sallittuun ryhmään. Tarkistuksessa haetaan yksilöllisen tunnisteen perusteella objekteja hakemistopuusta. Kirjauksen löytyessä siirrytään takaisin käyttäjäien käsittelijän *login()*-funktioon. Pääkäyttäjän kirjautuessa sisään tietojen tarkistaminen suoritetaan MySQL-tietokannan Users-taulussa. Molemmissa tapauksissa hyödynnetään CakePHP:n Auth-komponenttia käyttäjän session luomisessa sekä hallinnassa.

```
public function logout()
{
    $this->Session->setFlash(__('logoutSuccess'), 'flash_success');
    return $this->redirect($this->Auth->logout());
}
```

Esimerkkikoodi 12. Käyttäjän uloskirjaaminen

Sisään kirjautuneena käyttäjällä on pääsy huoneiden istuntojen hallintaan. Pääkäyttäjillä on sessioiden lisäksi oikeus huoneiden ja kameroiden hallintaan liittyviin sivuihin. *Kirjaudu ulos* -painike sijaitsee käyttöliittymän ylätunnisteessa. Sen klikkaaminen ohjaa käsittelijän *logout()*-funktioon (Esimerkkikoodi 12), jossa käyttäjän sessio tuhotaan.

5.6 Istuntojen suojaaminen



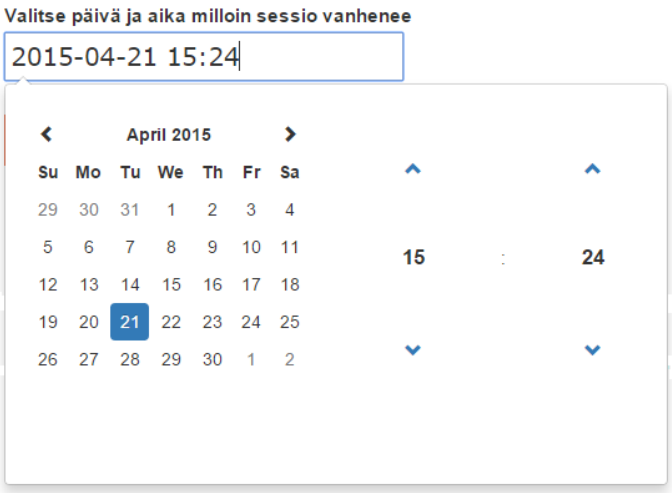
Kuvio 21. Lomake istunnon luomiseen huoneelle

Istuntojen tarkoituksena on suojata käyttöliittymän huoneet ulkopuolisilta katselijoilta harjoitteiden ajaksi. Pääsivulla sijaitseva huoneiden hallintapaneeli ohjaa istunnon luomiselle sekä muokkaamiselle tarkoitettuihin näkymiin. Session tekeminen tapahtuu kuvion 21 sisältämässä lomakkeessa. Lomake sisältää generoidun istuntoavaimen sekä päivämäärä- ja aikakentän. *Luo avain* -painike arpoo satunnaisen viiden merkin pituisen merkkijonon hyödyntäen JavaScriptin matematiikka-oliota (Esimerkkikoodi 13).

```
function generateKey()
{
    var key = "";
    var choices = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
    //append random values into the key variable
    for(var i = 0; i < 5; i++)
    {
        key += choices.charAt(Math.floor(Math.random() * choices.length));
    }
    //set the key variables value to the RoomKey form field
    document.getElementById('RoomKey').value = key;
}
```

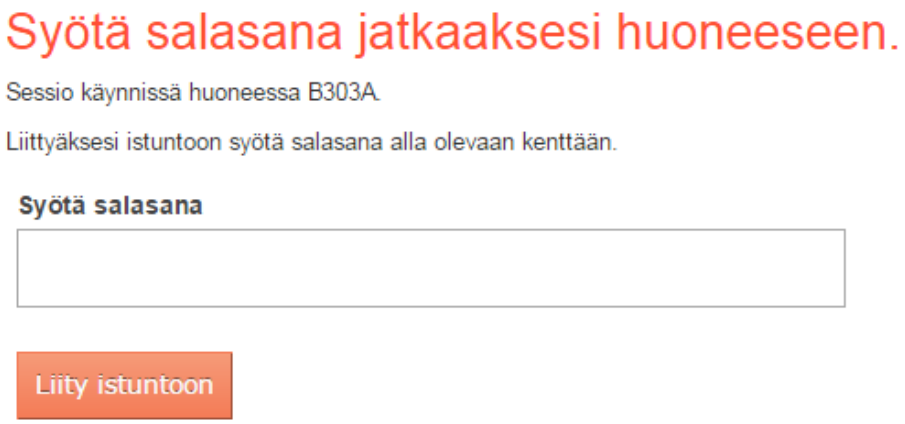
Esimerkkikoodi 13. JavaScript-funktio sessioavaimen generoimiselle

Funktiossa arvotun merkkijonon arvo siirretään lomakkeen *generoitu avain* -kenttään. Seuraavaksi päätetään istunnon päättymisajankohta. Päivämäärän sekä ajan valinnan toteutukseen hyödynnettiin bootstrap-komponenttia (Kuvio 22). Kyseinen komponentti sisältää päivän ja ajan valinnan yhdessä paketissa, mikä nopeuttaa lomakkeen täyttämistä.



Kuvio 22. Päivämäärän ja ajan valintaan käytetty komponentti

Luo istunto -painikkeen klikkaamisen jälkeen avautuu ikkuna, joka sisältää generoidun avaimen sekä voimassaoloajan. Kyseisessä huomautuksessa käyttäjää kehotetaan tallentamaan istuntoavain myöhempää käyttöä varten. Ennen tietokantaan tallentamista istuntoavain täytyy tiivistää sekä päivämääräkomponentin arvo asettaa oikeaan muuttujaan. Kyseiset toimenpiteet suoritetaan huoneen mallin beforeSave()-funktiossa (Liite 3). Avaimen tiivistämiseen käytetään bcrypt-algoritmia, joka on huomattavasti turvallisempi verrattuna esimerkiksi sha-tiivistefunktioihin. Huoneen käsittelijässä (Liite 1) sijaitseva *session()*-funktio hoitaa näkymän tietojen tallennuksen tilan tietokanta merkintään. Aktiivisen istunnon tunnistaa sivuston päänäkymässä olevista vihreistä kamera ikoneista, joiden painaminen ohjaa alla olevan kuvion 23 näkymään.



Kuvio 23. Näkymä sessioavaimen tarkistukseen

Sessioavaimen tarkistaminen tapahtuu käsittelijän *pwdCheck()*-funktiossa (Liite 1). Tarkistuksen yhteydessä lomakkeeseen syötetty salasana tiivistetään, jonka jälkeen sitä verrataan tietokannassa olevaan arvoon. Avainten tiivisteiden vastatessa toisiaan luodaan käyttäjälle kyseiseen huoneeseen aktiivinen istunto, joka takaa pääsyn sessioon. Aktiivisen istunnon tarkistukseen ja säilömiseen hyödynnetään CakePHP:n sessio-komponenttia. Esimerkkikoodissa 14 nähdään, miten avainten vertaaminen sekä aktiivisen istunnon luominen tapahtuu.

```
//Alustetaan istunto muuttujat
$new_sessions = array();
$active_sessions = array();

//Tarkistetaan vastaako syötetty avain kannassa olevaa
//Mikäli vastaa luodaan uusi aktiivinen istunto käyttäjälle
if(password_verify($this->request->data['Room']['key_check'], $room['Room']['session_key']))
{
    $new_sessions[] = $room;

    if($this->Session->check('ActiveSessions'))
    {
        $active_sessions = $this->Session->read('ActiveSessions');
    }
    $active_sessions = array_merge($active_sessions, $new_sessions);
    $this->Session->write('ActiveSessions', $active_sessions);
    $this->redirect(array('controller' => 'rooms', 'action' => 'view', $id));
}
```

Esimerkkikoodi 14. Aktiivisen istunnon luominen käyttäjälle

Käsittelijän *view()*-funktiossa (Liite 1) tarkistetaan, onko huoneella merkittynä tietokantaan istuntoavainta. Mikäli avain löytyy, täytyy käyttäjällä olla aktiivinen istunto tallennettuna sessiokomponenttiin. Kyseisen istunnon luominen tapahtuu

huoneen avaimen tarkistuksen yhteydessä, mikäli syötetty salasana vastaa kannassa olevaa. Käyttäjän yrittäessä päästä katsomaan tilaa ilman komponenttiin tallennettua istuntoa, hänet ohjataan takaisin käyttöliittymän pääsivulle (Esimerkkikoodi 15). Kyseisellä menetelmällä estetään huoneeseen pääsy esimerkiksi suoraan selaimen osoiterivin kautta.

```
//Mikäli huoneessa voimassa oleva istunto tarkistetaan onko käyttäjä
//sessio muuttujaan tallennettu aktiivista sessiota
if($room['Room']['session_key'])
{
    //Mikäli ei aktiivista sessiota, estä pääsy sivulle
    if(!$this->Session->read('ActiveSessions'))
    {
        $this->Session->setFlash(__('accesDenied'), 'flash_failure');
        $this->redirect(array('controller' => 'rooms', 'action' => 'index'));
    }
}
```

Esimerkkikoodi 15. Ote *view()*-funktion aktiivisen session tarkistamisesta

Istunnon vanhetessa huoneelta tyhjennetään tietokannasta sessioavain sekä päättymisajankohta-kentät. Toimenpiteen suorittamiseksi hyödynnetään tietokannan ajastettua tehtävää, joka tarkistaa huonetaulun vanhentuneen tiedon varalta viidentoista minuutin välein. Tarkistuksessa verrataan huoneen päättymisajankohta kenttiä nykyhetkeen.

6 TESTAUS JA JATKOKEHITYS

6.1 Testaus

Ennen sivuston siirtämistä tuotantopalvelimelle suoritetaan sivuston testaus. Sivusto sisältää ominaisuuksia, jotka eivät välttämättä toimi jokaisella selaimella. Käyttöliittymän näkymät ominaisuuksineen käydään läpi yleisimmillä selaimilla: FireFox, Chrome, Internet Explorer ja Safari. Yhteensopivuusongelman ilmetessä toiminto pyritään korjaamaan, jotta sivustoa voidaan käyttää mahdollisimman monella selaimella.

Sivuston käytettävyyden testaaminen on käyttökokemuksen kannalta olennaista. Käytettävyyden testaaminen suoritetaan heuristisen analyysin avulla. Sivusto luovutetaan testihenkilöiden arvioitavaksi. He etsivät käyttöliittymästä mahdollisia puutteita ja epäkohtia. Tarkoituksena on kerätä tietoa sivuston käytettävyydestä, jonka pohjalta käyttöliittymää voidaan parantaa.

Istuntojen turvallisuutta testattaessa pyritään paikallistamaan mahdolliset haavoittuvuudet voimassa olevassa sessiossa. Testattavia asioita ovat mm: 1) voimassaolevaan istuntoon liittymisen mahdollisuus ilman avaimen syöttöä sekä 2) istunnon tuhoaminen, kun sivustolta poistutaan tai istunto menee umpeen. Testien perusteella löydetyt haavoittuvuudet paikataan, mikä takaa entistä turvallisemman ympäristön harjoitteiden seuraamiselle.

Viimeinen vaihe ennen sivuston julkistamista on sosiaali- ja terveysalan henkilökunnan ohjeistaminen palvelun käyttämisessä. Koulutuksen tarkoituksena on rohkaista henkilökuntaa sivuston käyttöönottamisessa sekä mahdollistaa palvelun tarkoituksenmukainen hyödyntäminen opetusta tukevana työvälineenä. Koulutuksen aikana seurataan, kuinka toimenpiteiden suorittaminen onnistuu ja mikä mahdollisesti tuottaa ongelmia. Havainnointien sekä saadun palautteen perusteella suoritetaan sivuston viimeiset parannukset ennen tuotantopalvelimelle siirtämistä.

6.2 Jatkokehitys

Kehitysvaiheen aikana nostettiin esille ominaisuuksia, jotka eivät sisältyneet alkuperäiseen toteutuksen rajaukseen. Kyseiset ominaisuudet tarjoaisivat mm. moniaistillisen käyttökokemuksen sekä suoratoistomateriaalin mahdollisen käyttämisen harjoitteiden suorittamisen jälkeen. Seuraavissa kappaleissa on kuvattu mahdolliset lisäominaisuudet sekä mainittu teknologiat niiden lisäämiseksi sivustolle.

ENVI-tiloissa suoritettujen harjoitteiden tallentaminen ei ole tällä hetkellä mahdollista käyttöliittymän kautta. Halutessa tallentaa videokuvaa tallennus täytyy tällä hetkellä hoitaa kameran oman hallintasivun kautta. Harjoitteiden suoratoisto tallennettaisiin esimerkiksi NAS-verkkotallennus-järjestelmään. Suoratoiston tallentaminen voidaan toteuttaa sivustolle joko manuaalisesti tai ajastetusti. Tallentaminen manuaalisesti tapahtuisi kameroiden katselu-sivulla (rooms/view). Näkymään sijoitettaisiin painike, jonka klikkaaminen aloittaisi huoneen kameroiden suoratoiston tallentamisen. Tallennusominaisuudet olisivat vain sisäänkirjautuneiden käyttäjien ulottuvilla. Ajastetun tallennuksen määrittäminen tapahtuisi istunnon luomisen yhteydessä. Sessio-lomakkeeseen lisättäisiin vaihtoehtoinen lomake, johon määritetään milloin tallennus aloitetaan ja lopetetaan. Suoratoiston tallentaminen mahdollistetaan Vapix-ohjelmointirajapinnan avulla.

Kameroiden suoratoistonäkymissä ei tällä ole hetkellä mahdollista kuunnella kuvattavan tilan ääniä. Näkymässä valitun kameran ääniraidan lisääminen suoratoistoon parantaisi käyttäjäkokemusta. Vapix-ohjelmointirajapinnan avulla olisi mahdollista toteuttaa kyseinen ominaisuus sivustolle.

Tiloissa olevat Axiksen verkkokamerat tarjoavat mahdollisuuden kaksisuuntaiseen kommunikointiin. Harjoitusten johtavalle opettajalle olisi hyvä tarjota mahdollisuus kyseisen ominaisuuden hyödyntämiseen käyttöliittymästä. Tällä tarkoitetaan, että opettaja pystyisi johtamaan harjoitusta pelkästään kameran välityksellä. Käyttöliittymään lisättäisiin *push-to-talk* -painike, joka välittäisi opettajan puheen käyttöliittymästä valitun kameran kautta.

Sivuston ulkoasun responsiivisuus on ominaisuus, joka parantaisi käytettävyyttä muilla laitteilla. Www-sivujen selaaminen mobiililaitteilla on kasvanut räjähdysmäisesti laitteiden kehityksen myötä. Nykyään yhä useampi käyttäjä selaa verkkosivuja mobiililaitteella perinteisen tietokoneen sijaan. Responsiivisen tyylin toteuttaminen on mahdollista bootstrap-sovelluskehyksen avulla.

Sivustolla hyödynnetään Lapin Ammattikorkeakoulussa käytössä olevaa Microsoftin Active Directory -hakemistopalvelua käyttäjien tunnistamisessa. Sen käyttö on rajattu tällä hetkellä pelkästään henkilökunnan sisäänkirjautumisen tarkistamiseen. Sivuston turvallisuutta saataisiin parannettua laajentamalla sisäänkirjautuminen koskemaan koko palvelua. Tällöin sivustolle pääsisivät pelkästään Lapin ammattikorkeakoulun tunnusten haltijat. Toimenpide vaatisi ldap-haun laajentamista koskemaan myös opiskelijoita.

7 YHTEENVETO

Opinnäytetyön tavoitteena oli toteuttaa www-sivusto, jonka kautta tarjotaan mahdollisuus seurata virtuaalisessa oppimisympäristössä suoritettavia harjoitteita. Tavoitteessa onnistuttiin hyvin. Toteutetun sivuston avulla voidaan hallinnoida ENVI-oppimisympäristön verkkokameroiden näkyvyyttä käyttöliittymässä sekä katsoa tarjottujen kameroiden suoratoistoa. Lisäksi tavoitteena oli suojata haluttu tila harjoitteen ajaksi. Tavoite saavutettiin: sivuston huoneet ovat suojattavissa tarpeen vaatiessa. Harjoituksesta vastaava opettaja huolehtii suojaamisesta.

Sivusto on hyödyllinen opetusta tukeva työkalu ENVI-oppimisympäristön käyttäjille. Palvelu mahdollistaa harjoitustilanteiden seuraamisen toisessa huoneessa, mikä vähentää ylimääräistä tungosta itse harjoitustilassa. Sivusto vaatii sosiaali- ja terveysalan henkilökunnalta omatoimisen istuntojen suojaamisen, mikäli tarkoituksenmukaista ei ole tarjota pääsyä harjoitteen seurantaan kaikille. Harjoitteiden katseluoikeuden rajoittaminen on mahdollista erikseen luotavilla istunnoilla, joiden hyödyntäminen on suotavaa.

Opinnäytetyön aikana ongelmaksi muodostui Microsoft Active Directoryn käyttöönotto henkilökunnan sisäänkirjautumisen mahdollistamiseksi. LDAP-protokollan kanssa työskenneltäessä ongelmallista oli virhekirjausten puutteellisuus. Ongelman ratkaiseminen oli huomattavan vaikeaa, sillä lokikirjauksissa ei tarjottu tarpeeksi informaatiota mahdollisesta virheen aiheuttajasta. Kameroiden äänen lisäämistä suoratoistonäkymään yritettiin, mutta se muodostui odotettua monimutkaisemmaksi toimenpiteeksi. Ajanpuutteen vuoksi kyseinen ominaisuus rajattiin toteutetusta versiosta pois.

Opinnäytetyöprosessi oli erittäin mielenkiintoinen ja haastava. Työn tekemisen aikana CakePHP-sovelluskehys tuli entistä tutummaksi. Opinnäytetyö tarjosi mahdollisuuden tutustua uusiin ohjelmointirajapintoihin ja teknologioihin. Verkkokameroiden käyttöönotto sivustolla tarjosi miellyttävää vaihtelua perinteiseen www-sovellusohjelmointiin. Kokonaisuutena aihe oli monipuolinen sekä opettavainen.

LÄHTEET

Axis 2013a. Video Streaming API. Viitattu 29.4.2015 http://www.axis.com/files/manuals/vapix_video_streaming_52937_en_1307.pdf.

- 2013b. General System Settings. Viitattu 29.4.2015 http://www.axis.com/files/manuals/vapix_general_system_settings_52921_en_1307.pdf.

- 2013c. Introduction to VAPIX. Viitattu 29.4.2015 http://www.axis.com/files/manuals/vapix_introduction_52924_en_1307.pdf.

Axis 2014a. Technical guide to network video. Viitattu 10.4.2015 http://www.axis.com/files/brochure/bc_techguide_60870_en_1411_lo.pdf.

- 2014b. Axis M10 Network Camera Series. Viitattu 2.5.2015 http://www.axis.com/files/datasheet/ds_m10_55974_en_1412_lo.pdf.

- 2014c. Axis P3304 Network Camera. Viitattu http://www.axis.com/files/datasheet/ds_p3304_61118_en_1411_lo.pdf.

Axis 2015a. About Axis. Viitattu 10.4.2015 http://www.axis.com/fi/en/about-axis.

- 2015b. Video Management systems. Viitattu 13.4.2015 http://www.axis.com/global/en/learning/web-articles/technical-guide-to-network-video/integrated-systems.

- 2015c. Vapix Version 3. Viitattu 30.4.2015 http://www.axis.com/global/en/support/developer-support/vapix.

Bootstrap 2015. Bootstrap home page. Viitattu 25.4.2015 http:// http://getbootstrap.com/.

CakePHP 2015a. CakePHP Cookbook. Viitattu 28.4.2015 http://book.cakephp.org/2.0/_downloads/en/CakePHPCookbook.pdf.

- 2015b. Core Libraries. Viitattu 27.4.2015 http://book.cakephp.org/2.0/en/core-libraries.html.

- 2015c. General Purpose Libraries. Viitattu 27.4.2015 http://book.cakephp.org/2.0/en/core-libraries/toc-general-purpose.html.

- 2015d. Behaviors. Viitattu 28.4.2015 http://book.cakephp.org/2.0/en/models/behaviors.html.

- 2015e. CakePHP 3.0.0 is here. Viitattu 20.4.2015 http://bakery.cakephp.org/2015/03/22/CakePHP-3-0-0-is-Here.html.

- 2015f. Components. Viitattu 28.4.2015 http://book.cakephp.org/2.0/en/core-libraries/toc-components.html.

- 2015g. Helpers. Viitattu 30.4.2015 http://book.cakephp.org/2.0/en/views/helpers.html.

- 2015h. Utilities. Viitattu 30.4.2015 http://book.cakephp.org/2.0/en/core-libraries/toc-utilities.html.

Heinusuo, R. 2004. PHP ja MySQL. Helsinki: Talentum.

Hopkins, C 2011. Why you should use bcrypt to hash stored passwords. Viitattu 5.5.2015 http://www.sitepoint.com/why-you-should-use-bcrypt-to-hash-stored-passwords/

Lapin AMK 2013. Graafinen ohjeisto. Viitattu 15.4.2015 http://www.lapinamk.fi/loader.aspx?id=0efaeda4-aef1-4809-ae06-8bb08c11be2c.

Lapin AMK 2015a. ENVI – Rovaniemi. Viitattu 4.5.2015 http://www.lapinamk.fi/fi/Tyoelamalle/Kehittamisymparistot/ENVI---Rovaniemi.

- 2015b. pLAB – Ohjelmistotekniikan laboratorio. Viitattu 4.5.2015 http://www.lapinamk.fi/fi/Tyoelamalle/Kehittamisymparistot/pLAB.

Microsoft Windows 2015. FTP FAQ. Viitattu 28.4.2015 http://windows.microsoft.com/en-us/windows-vista/file-transfer-protocol-ftp-frequently-asked-questions.

Peterse, Y. 2011. Use BCrypt Fool. Viitattu 3.5.2015 http://yorickpeterse.com/articles/use-bcrypt-fool/

PHP 2015a. LDAP book. Viitattu 5.5.2015 http://php.net/manual/en/book.ldap.php.

- 2015b. Crypt() function. Viitattu 3.5.2015 http://php.net/manual/en/function.crypt.php.

- 2015c. Password_hash() function. Viitattu 3.5.2015 http://php.net/manual/en/function.password-hash.php.

- 2015d. Password_verify() function. Viitattu 4.5.2015 http:// http://php.net/password_verify.

Ruohonen, M. 2002. Tietoturva. Jyväskylä: Docendo.

Wikipedia 2015. Active Directory. Viitattu 5.5.2015 http://fi.wikipedia.org/wiki/Active_Directory

LIITTEET

Liite 1. RoomsController.php
Liite 2. ldap.php
Liite 3. Room.php

RoomsController.php Liite 1 1(3)

```
//Huoneiden käsitteljiä
class RoomsController extends AppController
{
    public $helpers = array('Html', 'Form', 'Session');
    public $components = array('Session');

    public function beforeFilter() {
        parent::beforeFilter();
        //Näkymät joihin kaikilla kaikilla käyttäjillä
        $this->Auth->allow('index', 'view', 'pwdCheck');
    }

    //Sivuston pääsivu
    public function index() {
        //Haetaan tietokannasta luodut huoneet
        $this->set('rooms', $this->Room->find('all'));
    }

    //Huoneen kameroiden katsominen
    public function view($id = null) {
        //Etsitään valitun huoneen tiedot tietokannasta tunnistaan avulla
        $room = $this->Room->findById($id);

        //Haetaan kameran malli
        $this->loadModel('Camera');
        //Haetaan tietokannasta huoneeseen kuuluvat kamerat huoneen tunnis-
        //teen avulla
        $cameras = $this->Room->Camera->find('all', array('conditions' =>
                        array('room_id' => $id)));
        $this->set('cameras', $cameras);
        if(!$id)
        {
            throw new NotFoundException(__('Invalid room'));
        }
        if(!$room)
        {
            throw new NotFoundException(__('Invalid post'));
        }
        $this->set('room', $room);

        //Mikäli huoneessa voimassa oleva istunto tarkistetaan onko käyttäjän
        //sessio muuttujaan tallennettu aktiivista sessiota
        if($room['Room']['session_key'])
        {
            //Mikäli ei aktiivista sessiota, estä pääsy sivulle
            if(!$this->Session->read('ActiveSessions'))
            {
                $this->Session->setFlash(__('accesDenied'), 'flash_failure');
                $this->redirect(array('controller' => 'rooms',
                                'action' => 'index'));
            }
        }
        //Sivulta poistuessa tuhoa aktiivinen sessio
        $this->Session->delete('ActiveSessions');
    }

    //Sessio avaimen tarkistaminen
    public function pwdCheck($id = null) {
        if(!$id)
        {
            throw new NotFoundException(__('Invalid room'));
        }
        $room = $this->Room->findById($id);
```

```
        $this->set('room', $room);

        if(!$room)
        {
            throw new NotFoundException(__('Invalid room'));
        }

        //Käyttäjä painanut näkymän liity istuntoon painiketta
        if($this->request->is(array('post')))
        {
            //Alustetaan istunto muuttujat
            $new_sessions = array();
            $active_sessions = array();

            //Tarkistetaan vastaako syötetty avain kannassa olevaa
            //Mikäli vastaa luodaan uusi aktiivinen istunto käyttäjälle
            if(password_verify($this->request->data['Room']['key_check'],
                  $room['Room']['session_key'])
            )
            {
                $new_sessions[] = $room;

                if($this->Session->check('ActiveSessions'))
                {
                    $active_sessions = $this->Session->read('ActiveSessions');
                }
                $active_sessions = array_merge($active_sessions, $new_sessions);
                $this->Session->write('ActiveSessions', $active_sessions);
                $this->redirect(array('controller' => 'rooms',
                                      'action' => 'view', $id));
            }
            else
            {
                $this->Session->setFlash(__('sessionKeyFailure'), 'flash_failure');
            }
        }
    }

    //Huoneen lisääminen sivustolle
    public function add() {
        //käyttäjä klikannut tallenna painiketta
        if($this->request->is('post'))
        {
            $this->Room->create();
            //Tallennetaan lomakkeen tiedot tietokantaan
            if($this->Room->save($this->request->data))
            {
                $this->Session->setFlash(__('roomSaved'), 'flash_success');
                return $this->redirect(array('action' => 'index'));
            }
            $this->Session->setFlash(__('roomFailure'), 'flash_failure');
        }
    }

    //Huoneen muokkaaminen tietokannasta
    public function edit($id = null) {
        if (!$id) {
            throw new NotFoundException(__('Invalid room'));
        }

        $room = $this->Room->findById($id);
        if (!$room) {
            throw new NotFoundException(__('Invalid room'));
        }
```

```
        if ($this->request->is(array('post', 'put'))) {
            $this->Room->id = $id;
            if ($this->Room->save($this->request->data)) {
                $this->Session->setFlash(__('roomEditSaved'));
                return $this->redirect(array('action' => 'index'));
            }
            $this->Session->setFlash(__('roomFailure'));
        }
        if (!$this->request->data) {
            $this->request->data = $room;
        }
    }

    //Huoneen poistaminen tietokannasta
    public function delete($id) {
        if($this->request->is('get'))
        {
            throw new MethodNotAllowedException();
        }
        if($this->Room->delete($id))
        {
            $this->Session->setFlash(__('roomDelete'), 'flash_success');
        }
        else
        {
            $this->Session->setFlash(__('roomDeletefail'), 'flash_failure');
        }
        return $this->redirect(array('action' => 'index'));
    }

    //Istunnon luominen huoneelle
    public function session($id = null) {
        if(!$id)
        {
            throw new NotFoundException(__('Invalid room'));
        }

        //Haetaan tietokannasta huoneen tiedot jolle istuntoa ollaan luomassa
        $room = $this->Room->findById($id);
        if(!$room)
        {
            throw new NotFoundException(__('Invalid room'));
        }
        $this->set('room', $room);

        //Käyttäjä klikannut luo istunto painiketta
        if($this->request->is(array('post', 'put')))
        {
            //Päivitetään huoneen tiedot tietokantaan
            $this->Room->id = $id;
            if($this->Room->save($this->request->data))
            {
                $this->Session->setFlash(__('sessionCreated'), 'flash_success');
                return $this->redirect(array('action' => 'index'));
            }
            $this->Session->setFlash(__('sessionFailure'), 'flash_failure');
        }

        if(!$this->request->data)
        {
            $this->request->data = $room;
        }
    }
```

ldap.php Liite 2

```
class ldap
{
    //Käyttäjän tunnistaminen ms ad:ta vasten
    public function auth($user, $password)
    {
        //Käyttäjätietokannan ip-osoite, muutettu todellisesta arvosta
        $ldap = ldap_connect('xxx.xxx.xxx.xxx', 389);
        //rdn muutettu todellisesta arvosta
        $ldaprdn = 'AAA\\' . $user;

        ldap_set_option($ldap, LDAP_OPT_PROTOCOL_VERSION, 3);
        ldap_set_option($ldap, LDAP_OPT_REFERRALS, 0);

        $bind = @ldap_bind($ldap, $ldaprdn, $password);

        if($bind)
        {
            //dn muutettu todellisesta arvosta
            $dn = 'OU=AAAAA,OU=BBBBB,OU=CCCCC,DC=DDDDD,DC=EEEEE';
            $filter = "(&(sn=user)(givenname=".$user."))";
            $justthese = array("givenname");

            $search = ldap_search($ldap, $dn, $filter, $justthese);

            $info = ldap_get_entries($ldap, $search);

            if(count($info) > 0)
            {
                if (($bind = @ldap_bind($ldap, $ldaprdn, $password)))
                {
                    foreach($info as $k => $i)
                    {
                        if(isset($i[' givenname ']))
                        {
                            return true;
                        }
                    }
                }
                else
                {
                    return false;
                }
            }
        }

    }
}
```

Room.php Liite 3

```
App::uses('BlowfishPasswordHasher', 'Controller/Component/Auth');

class Room extends AppModel {
    public $name = 'Room';
    //Relaatio kamera malliin
    public $hasMany = 'Camera';

    //Lomakkeen kenttien validointi
    public $validate = array(
        'name' => array(
            'rule' => 'notEmpty'
        ),
        'positionx' => array(
            'rule' => 'notEmpty'
        ),
        'positiony' => array(
            'rule' => 'notEmpty'
        ),
    );

    //Suoritetaan ennen kuin lomakkeen tiedot tallennetaan tietokantaan
    public function beforeSave($options = array())
    {
        if(isset($this->data[$this->alias]['key']) && isset($this->
                                data[$this->alias]['datetime']))
        {
            //Luo salasana tiivisteen ja tallenntaa sen tietokantaan
            $this->data[$this->alias]['session_key'] = password_hash($this->
                                data[$this->alias]['key'], PASSWORD_BCRYPT);

            //Tallentaa bootstrap päivävalitsimen arvon session_expires kenttään
            //tietokannassa

            $this->data[$this->alias]['session_expire'] = $this->data[$this->
                                                        alias]['datetime'];
        }
        return true;
    }
}
```