

Opinnäytetyö (AMK)

Tietojenkäsittely

Sähköisen liiketoiminnan järjestelmät

2012

Joonas Koskimäki ja Mikael Laaksonen

DYNAAMISEN ILMOITUSTAULUN TOTEUTUS JA TESTAUS MULTIMEDIATUOTANTO- YHTIÖLLE



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Sähköisen liiketoiminnan järjestelmät

Kesäkuu 2012 | 58

Päivi Killström

Joonas Koskimäki ja Mikael Laaksonen

DYNAAMISEN ILMOITUSTAULUN TOTEUTUS JA TESTAUS MULTIMEDIATUOTANTOYHTIÖLLE

Tämän opinnäytetyön tavoitteena oli toteuttaa ja testata dynaaminen ilmoitustaulu multimediatuotantoyhtiö Data Media Gazellelle. Ilmoitustaulu luotiin Data Media Gazellen ja Tietokonekauppa.fi:n käyttämän vanhan ilmoitustaulun tilalle. Uuden ilmoitustaulun tuli olla dynaaminen ja käyttäjäystävällinen. Ilmoitustaulun päätavoitteena oli pystyä lähettämään ilmoituksia, joihin lähettäjällä oli mahdollisuus liittää liitetiedostoja. Lähetettyjen ilmoitusten tuli olla muiden ilmoitustaulun käyttäjien nähtävillä ja käyttäjillä täytyi olla mahdollisuus kommentoida lähetettyjä ilmoituksia.

Käytimme ilmoitustaulun toteutukseen avoimen lähdekoodin PHP-ohjelmointikieltä, MySQL-relaatiotietokantaohjelmistoa, CSS-tyylikieltä ja jQuery -JavaScript kirjastoa. Testaukseen käytimme avoimen lähdekoodin Selenium-testausautomaatiosovellusta ja jMeter-kuormitustestaussovellusta.

Ilmoitustaulua aloimme työstää syksyllä 2011 toisen tekijän luoman ja Data Media Gazellen hyväksymän suunnitelman mukaisesti. Lopputuloksena syntyi ilmoitustaulu, joka täytti alkuperäisen suunnitelman mukaan asetetut tavoitteet. Ilmoitustaulu luotiin Data Media Gazellelle, mutta he eivät ottaneet sitä käyttöön, joten ilmoitustaulu ei tullut tämän opinnäytetyön valmistumisvaiheen jälkeen kenenkään käyttöön.

ASIASANAT:

verkko-ohjelmointi, testaus, PHP, HTML

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | e-Business Systems

June 2012 | 58

Päivi Killström

Joonas Koskimäki and Mikael Laaksonen

IMPLEMENTING AND TESTING A DYNAMIC NOTICE BOARD TO MULTIMEDIA PRODUCTION COMPANY

The main objective of this thesis was to build and test a dynamic notice board for the use of a multimedia production company called Data Media Gazelle. The notice board was created to replace a previous notice board used by Data Media Gazelle and Tietokonekauppa.fi. The new notice board was intended dynamic and user friendly. The main objective of the new notice board was to be able to send notices, in which a sender had a possibility to add attachments. Other users also had to have an opportunity to see posted notices and comment them.

To implement the notice board we used an open source PHP -programming language, MySQL –relational database software, CSS –style sheets and jQuery –JavaScript library. In testing we used an open source test automation tool called Selenium and a load test tool called jMeter.

We started to work on the notice board in autumn 2011 based on the other author's design which was accepted by Data Media Gazelle. As a final result a notice board was born which fulfilled all the demands based on the original design. The notice board was created for the use of Data Media Gazelle, but they did not put it to use, so the notice board was not used by anyone after this thesis was completed.

KEYWORDS:

Web-programming, testing, PHP, HTML

SISÄLTÖ

1 JOHDANTO	7
2 LÄHTÖKOHDAT	8
2.1 Asiakasyritys	9
2.2 Työn vaatimukset	9
2.3 Työnjako	12
2.4 Käyttäjäystävällinen käyttöliittymä	12
2.5 Toteutustekniikat	14
3 PROJEKTIN VAIHEET	16
3.1 Tietojärjestelmän suunnittelu	16
3.1.1 Käyttäjä	16
3.1.2 Ilmoitustaulun selaaminen	17
3.1.3 Ilmoituksen lisääminen	18
3.1.4 Tietomallin rakenne	20
3.1.5 Käyttötapaukset	24
3.2 Tietojärjestelmän toteutus	26
3.2.1 Järjestelmään kirjautuminen	26
3.2.2 Ilmoitustaulun päänäkymä	30
3.2.3 Ilmoituksen lisäys	33
3.2.4 Kommentin lisäys	35
3.2.5 Ilmoitustaulun tapahtumaloki	38
3.2.6 Ilmoitustaulun oikeellisuustarkistus	39
3.2.7 Käyttöliittymän arviointi	44
3.3 Tietojärjestelmän testaus	45
3.3.1 Manuaalinen testaus	46
3.3.2 Testitapaukset	46
3.3.3 Testiautomaation luonti	47
3.3.4 Kuormitustestaus	50
3.3.5 Tietoturvatestaus	52
3.3.6 Toimeksiantajan suorittama testaus	53
4 YHTEENVETO	54

LIITTEET

- Liite 1. Selenium testiraportti
- Liite 2. jMeter testiraportti
- Liite 3. Websecurify testiraportti

KUVAT

Kuva 1. Data Media Gazellen käytössä oleva ilmoitustaulu.	8
Kuva 2. Käyttöliittymän hahmotelma.	11
Kuva 3. Kirjautuminen ilmoitustauluun.	27
Kuva 4. Virheilmoitus kirjautuessa.	28
Kuva 5. Virheilmoitus, pakolliset kentät ovat tyhjiä..	29
Kuva 6. Ilmoitustaulun päänäkyvä ilmoitukset-välilehti avoinna.	31
Kuva 7. Ilmoitustaulun päänäkyvä liitetiedostot-välilehti avoinna.	33
Kuva 8. Ponnahdusikkuna, jossa käyttäjä lisää ilmoituksen.	34
Kuva 9. Ilmoitustaulun ilmoitusnäkyvä lisää kommentti näkyvä aukaistuna.	37
Kuva 10. Ilmoitustaulun tapahtumaloki.	39
Kuva 11. Virheilmoitus väärin merkkien lisäyksestä otsikkoon ja viestiin.	41
Kuva 12. Virheilmoitus liian lyhyestä otsikosta ja tyhjästä pakollisesta kentästä.	42
Kuva 13. Virheilmoitus yrityksestä lähettää tyhjä kommentti.	43
Kuva 14. Virheilmoitus väärin merkkien lisäyksestä kommenttiin	44
Kuva 15. Selenium IDE käyttöliittymä.	49
Kuva 16. Apache jMeter.	51

KUVIOT

Kaavio 1. Käyttäjä ilmoitustaulussa.	16
Kaavio 2. Ilmoitustaulun selaaminen.	17
Kaavio 3. Ilmoituksen lisääminen.	19
Kaavio 4. Järjestelmän tietomalli.	20

TAULUKOT

Taulukko 1. Käyttäjä-taulun rakenne.	21
Taulukko 2. OikeusTaso-taulun rakenne.	21
Taulukko 3. Ilmoitus-taulun rakenne.	22
Taulukko 4. Tiedosto-taulun rakenne.	22

Taulukko 5. Status-tilun rakenne.	23
Taulukko 6. Yritys-tilun rakenne.	24
Taulukko 7. Kommentti-tilun rakenne.	24
Taulukko 8. Ilmoitus-tilun selaus -käyttötapaus.	25
Taulukko 9. Ilmoituksen lisäys -käyttötapaus.	25
Taulukko 10. Ilmoituksen kommentointi -käyttötapaus.	26
Taulukko 11. Ilmoitus-tiluun kirjautumisen kentät	30
Taulukko 12. Ilmoituksen lisäys – ponnahdusikkunan kentät.	35
Taulukko 13. Kommentin lisäys ilmoitus-tiluun – ponnahdusikkunan kentät.	38
Taulukko 14. Ilmoitus-tilun selaus -testitapaus.	46
Taulukko 15. Ilmoituksen lisäys -testitapaus.	47
Taulukko 16. Ilmoituksen kommentointi -testitapaus.	47

1 JOHDANTO

Opinnäytetyömme tavoitteena oli suunnitella ja toteuttaa dynaaminen ilmoitustaulu opinnäytetyömme toimeksiantajalle SH-Group Oy:lle. Dynaaminen ilmoitustaulu tulee SH-Group Oy:n kuuluvien yksiköiden Data Media Gazellen ja Tietokonekauppa.fi:n sisäiseen viestintään.

Tässä opinnäytetyössä kerromme miten suunnittelimme ja toteutimme dynaamisen ilmoitustaulun aloittaen aluksi projektin lähtökohdista kertomalla asiakasyrityksestä ja työn vaatimuksista. Työn vaatimuksista siirrymme työnjakosioon, jossa kerromme miten jaoin vastuun opinnäytetyön työstämisessä. Päätämme lähtökohdat osion kertomalla käyttäjäystävällisen käyttöliittymän toteutuksesta ja toteutustekniikoista, joita käytimme projektin aikana.

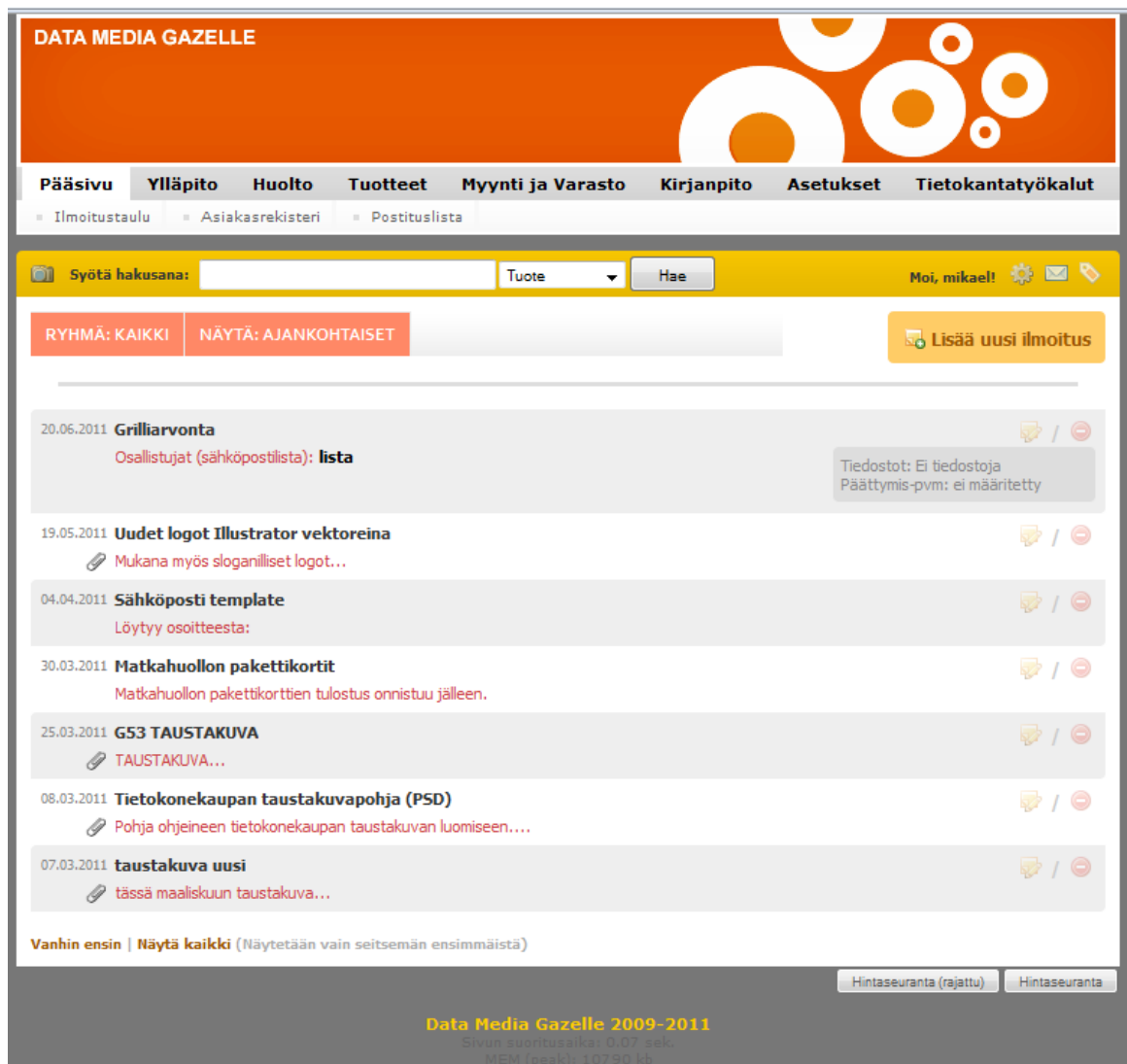
Seuraavaksi siirrymme projektin eri vaiheisiin aloittaen suunnitteluosion, jossa esitämme käyttöliittymän toimintaa kaavioiden, tietomallin ja taulujen avulla, jonka lisäksi esitämme käyttötapauksia, jotka kertovat mitä erityyppisiä asioita ilmoitustaulussa voidaan tehdä. Suunnitteluosuudesta siirrymme projektin toteutusvaiheeseen, jossa kuvailemme ilmoitustaulun toteutusta ja valitsemiamme toteutusmenetelmiä, esittäen kuvia käyttöliittymästä ja kaavioita, joissa esitämme ilmoitustaulun eri elementit ja niiden toiminnat. Toteutusosion lopussa kerromme ilmoitustaulun kenttien oikeellisuustarkistuksien toiminnasta.

Toteutusvaiheesta siirrymme projektin testausvaiheeseen, jossa perehdymme järjestelmän testaukseen eri testimenetelmiä hyödyntäen. Järjestelmää testataan manuaalisesti projektin edetessä sekä lopussa. Loppuvaiheessa luodaan myös testiautomaatio, eli testitapaukset nauhoitetaan ja järjestelmän kuormituskyky testataan.

Viimeisessä kappaleessa esitämme yhteenvedon opinnäytetyöstämme, jossa esitämme arviomme ilmoitustaulun onnistumisista ja epäonnistumisista, sekä kerromme myös mahdollisista jatkokehitysmahdollisuuksista.

2 LÄHTÖKOHDAT

Data Media Gazellella ja Tietokonekauppa.fi:llä on ollut käytössä jo vuosia vanha ilmoitustaulu käyttöliittymä (Kuva 1), jonka käyttö on jäänyt vähäiseksi kummassakin yrityksessä. Opinnäytetyön kartoitusvaiheessa paljastui, että suurin syy ilmoitustaulun vähäiseen käyttöön oli ollut käyttöliittymän käyttäjäystävällisyyden huono taso. Opinnäytetyöaiheen dynaamisesta ilmoitustaulusta saimme Mikael Laaksosen ollessa työharjoittelussa multimediatuotantoyhtiö Data Media Gazellella kesällä 2011.



Kuva 1. Data Media Gazellen käytössä oleva ilmoitustaulu.

2.1 Asiakasyritys

SH-Group Oy on emoyhtiö, johon sekä Date Media Gazelle että Tietokonekauppa.fi kuuluvat. SH-Group Oy on perustettu vuonna 1998. SH-Group Oy:n liikevaihto oli 2 466 000,00 € (12/2010), josta edellisvuoteen nähden oli kasvua 48,20 prosenttia (Yritystele.fi, 2011).

Date Media Gazelle on vuonna 2003 perustettu multimediatuotantoyhtiö, jonka tuotteita ovat verkkosivut, verkkokaupat, bannerit, paino- ja kuvitustuotteet. Yritys sijaitsee Turussa osoitteessa Itäinen Pitkäkatu 39. Yritys koostuu kahdesta vakituisesta työntekijästä ja kahdesta osa-aikaistyöntekijästä. Yrityksen vastuuhenkilönä toimii projektikoordinaattori (Dmgazelle.fi, 2012).

Tietokonekauppa.fi on täyden palvelun IT-alan jälleenmyynti- ja huoltoliike. Yrityksen asiakkaita ovat yritykset, kunnat, valtion laitokset ja yksityisasiakkaat. Yrityksen myymälätilat sijaitsevat osoitteessa Itäinen Pitkäkatu 26. Tietokonekauppa.fi:n myymälästä vastaa myymäläpäällikkö ja verkkosivujen toiminnasta vastaa ohjelmistosuunnittelija, joka työskentelee Date Media Gazellella. Date Media Gazelle huolehtii Tietokonekauppa.fi:n verkkosivujen lisäksi yrityksen markkinoinnista (Tietokonekauppa.fi, 2012).

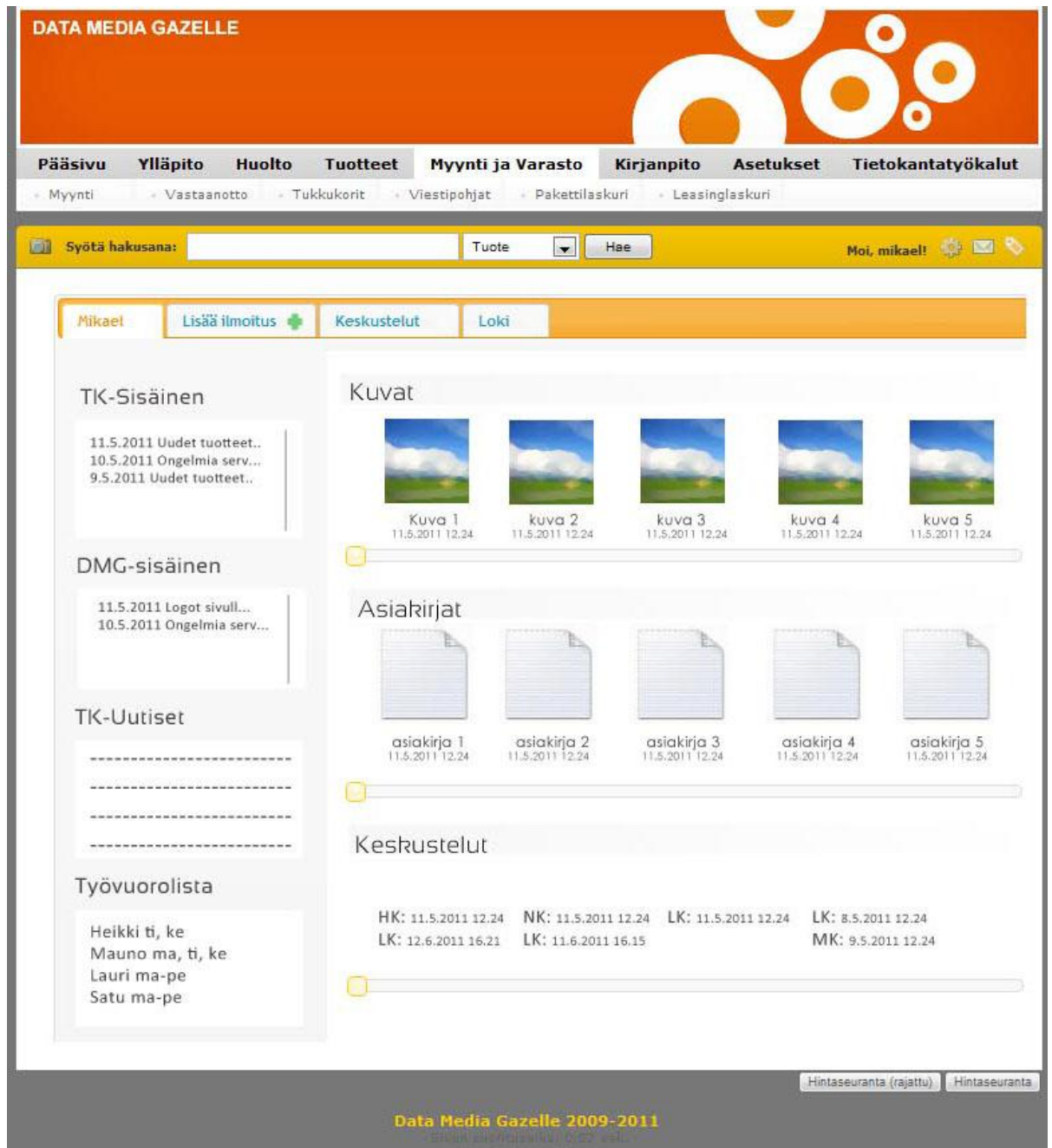
2.2 Työn vaatimukset

Ennen opinnäytetyön aloittamista tiedustelin projektin toimeksiantajalta, sekä yrityksen ohjelmointikoordinaattorilta heidän toiveitaan dynaamisen ilmoitustaulun toteutukseen. Keskustellessa heidän kanssaan kävi ilmi, että yrityksen vanhan ilmoitustaulun suurimpia ongelmia oli sen jäykkä käyttöliittymä, sekä informaation perille saaminen tietyille ryhmille tuotti ongelmia. Dynaamisen ilmoitustaulun käyttöliittymältä toivottiin käyttäjäystävällisyyttä, helppokäyttöisyyttä ja visuaalisesti näyttävää ulkoasua.

Date Media Gazelle ja Tietokonekauppa.fi käyttävät ilmoitustaulua sisäiseen viestintään. Tyypillinen esimerkki ilmoitustaulun käytöstä on: Tietokonekaup-

pa.fi:ssa huomataan virhe yrityksen verkkosivuilla, josta ilmoitetaan Data Media Gazelleen, jotta he korjaisivat havaitun vian. Ilmoitustaulussa viasta pitäisi pystyä ilmoittamaan muille käyttäjille viestinä, jonka he näkevät ja pystyvät tarvittaessa kommentoimaan viestiä. Esimerkiksi työntekijä1 huomaa verkkosivuilla vian, josta hän tekee ilmoituksen ilmoitustaululle: ”Hei, haulla kiintolevyt tulee error-viesti, voisitko katsoa ongelmaa”. Työntekijä2 huomaa työntekijä1:n ilmoituksen, paikantaa sen ja korjaa vian, jonka jälkeen työntekijä2 kommentoi työntekijä1:n vika-ilmoitukseen: ”Vika on korjattu, kiitos ilmoituksesta”.

Ilmoitustaulua käytetään intranetissä, joten ulkopuolisilla ei ole mahdollisuutta käyttää tai nähdä ilmoitustaulua, näin ollen ilmoitustaulun käyttäjämäärä jää vähäiseksi. Projektin tavoitteiden suunnittelu noudatti hyvin pitkälti toimeksiantajan näkemystä dynaamisen ilmoitustaulun toimivuudesta ja ulkoasusta. Toimeksiantajan näkemysten pohjalta loimme Adobe Photoshopilla käyttöliittymän hahmotelman (Kuva 2), jonka toimeksiantaja hyväksyi. Yhteydenpitomme Data Media Gazelleen oli ongelmallista, sillä lukuisten sähköpostiyhteydenottojen jälkeen, alkaen syyskuun lopusta ja päättyen marraskuun loppuun, emme saaneet vastauksia kysymyksiimme, jotka koskivat ilmoitustaulun kehitystä. Ilman jatko-opastusta Data Media Gazellen taholta päätimme, että teemme dynaamisen ilmoitustaulun parhaamme mukaan alkuperäisen suunnitelman mukaan.



Kuva 2. Käyttöliittymän hahmotelma.

Jälkeenpäin tammikuun lopussa toimeksiantaja ilmoitti sähköpostitse tarkemmin toivomuksistaan ilmoitustaulun ulkoasusta ja teknisistä tarpeista, sekä ilmoitti opinnäytetyömme yhteyshenkilönä toimineen ohjelmistosuunnittelijan lopettaneen työskentelyn Data Media Gazellella. Toivomuksissaan toimeksiantaja toivoi, että ilmoitustaulu muistuttaisi oikeaa ilmoitustaulua, jossa ilmoitukset

näyttäisivät ilmoituslapun näköisiä. Toimeksiantaja toivoi, että ilmoitukset kategorioitaisiin eri statuksiin, jonka käyttäjä voi valita ilmoitusta tehdessään. Ilmoitusten eri statukset tulisi näkyä ilmoitustaulussa eri värein, esimerkiksi ilmoitus status ”tiedote” näkyisi vihreänä ilmoitustaulussa. Statuksien määrittely eri värein tulisi helpottaa ilmoitusten erottamista toisistaan. Värit määritetään eri statuksille, joita toimeksiantajan ohjeiden mukaan tulisi olemaan kuusi kappaletta.

2.3 Työnjako

Projektissa, jossa on kyse kahdesta ihmisestä, täytyy ottaa huomioon tasapuolinen työnjako, ettei toinen projektin jäsenistä joudu tekemään kohtuuttomasti enemmän kuin toinen osapuoli. Reilua työnjakoa miettiessämme tulimme siihen lopputulokseen, että jaamme työtehtävät niin että Mikael keskittyy käyttöliittymän graafisen ulkoasun luomiseen ja Joonas taas puolestaan keskittyy tietokantaan ja vastaa järjestelmän testauksesta. Vaikka olimme jakaneet työnjaon eri osa-alueisiin, niin että Mikael oli keskittynyt käyttöliittymän graafiseen osaan ja Joonas tietokantaan oli meillä silti paljon vuorovaikutusta toistemme kanssa auttaen toisiamme mm. ongelmatilanteissa. Myös se, ettemme täysin sokeasti olleet keskittyneet vain omiin osa-alueisiimme auttoi siinä, että pystyimme ymmärtämään käyttöliittymän rakennetta paremmin ja näin ollen pystyimme paremmin yhdistämään graafista osaa tietokantaan ja toisinpäin.

2.4 Käyttäjäystävällinen käyttöliittymä

Näköaisti on normaalisti yksi ihmisen tärkeimmistä aisteista ja tästä syystä näköaistiin perustuvat käyttöliittymät ovat yleisimpiä (Kuutti, W. 2003, 25). Näköaistiin liittyy hahmolakeja, jotka tulkitsevat silmien havaintoja. Hahmolajit ovat aivoissa tapahtuvia tulkintoja, jotka ovat osin ennalta opittuja ja osin synnynnäisiä. Hahmolajit kertovat ihmiselle miten hän mieltää yhteenkuuluviksi näkemäänsä havaintoja. Hyvän käyttöliittymän tulisi noudattaa näitä ”luonnollisia” hahmolakeja (Kuutti, W. 2003, 27). Hahmolajitelmissa saattaa olla tiettyjä poikkeamia eri kulttuureissa, joten käyttöliittymää suunnitellessa täytyy ottaa huomi-

oon kenelle käyttöliittymää suunnitellaan. Esimerkiksi värimaailmassa punainen merkitsee länsimaissa yleensä kuumaa tai vaaraa, mutta muualla sen merkitys voi olla aivan toisenlainen (Kuutti, W. 2003, 45), esimerkiksi Kiinassa punainen symboloi rohkeutta ja kunniaa (Wikipedia, 2011).

Käytettävyyden heuristinen arviointi perustuu heuristiikkoihin, jotka ovat listoja säännöistä ja ohjeista (Kuutti, W. 2003, 49), joita käytettävyydeltään hyvän käyttöliittymän tulisi noudattaa. Käyttöliittymän käytettävyydestä on koottu sääntökokoelma, jota kutsutaan Nielsenin listaksi (Molich & Nielsen 1990).

Nielsenin heuristiset säännöt:

- Vuorovaikutuksen käyttäjän kanssa tulee olla yksinkertaista ja luonnollista.
- Vuorovaikutuksessa tulee käyttää käyttäjän omaa kieltä.
- Käyttäjän muistin kuormitus tulee minimoida.
- Käyttöliittymän tulee olla yhdenmukainen.
- Järjestelmän tulee antaa käyttäjälle kunnollista palautetta reaaliajassa.
- Ohjelmassa ja sen osissa tulee olla selkeät poistumistiet.
- Oikopolkuja ja tehokasta työskentelyä tulisi tukea.
- Virheilmoitusten tulee olla selkeitä ja ymmärrettäviä.
- Virhetilanteisiin joutumista tulisi välttää.
- Käyttöliittymässä tulee olla kunnolliset avustustoiminnot ja dokumentaatio.

Ilmoitustaulun ulkoasua suunnitellessa yritämme käyttää Nielsenin heuristisia sääntöjä, jotta ilmoitustaulun käyttö olisi mahdollisimman helppoa. Käyttäjäystävällisen käyttöliittymän suunnittelusta ja toteutusmenetelmistä kerromme paremmin luvussa 3.2.6 Käyttöliittymän arviointi.

2.5 Toteutustekniikat

Käyttöliittymän suunnitteluun käytimme Violet UML editor -ohjelmaa, jolla loimme kaaviot (Kaavio 1, Kaavio 2, Kaavio 3). Violet UML editor on ilmainen UML-kaavioiden muokkausohjelma, jolla pystyy suunnittelemaan helposti kaavioita tulostamalla ne kuviksi tai paperille tulostimella. Adobe Photoshop -kuvankäsittelyohjelmaa käytettiin käyttöliittymähahmotelmiin (Kuva 2). Notepad++ -tekstinkäsittelyohjelmaa käytimme PHP-, CSS- ja JavaScript-sivujen ohjelmointiin, sekä tiedostojen siirtoon tietokoneelta palvelimelle.

Käytimme toteutuksessa käyttöliittymärakenteiden luomiseen XHTML-kuvauskieltä, sekä CSS-tyylipohjia ja jQuery- JavaScript-kirjastoa määrittämään käyttöliittymän graafisen ilmeen.

XHTML-kuvauskieli on lähes identtinen HTML 4.01:n kanssa, mutta tiukempi ja puhtaampi versio HTML:stä ja se tukee kaikkia tärkeimpiä web-selaimia (W3schools, 2012). CSS-tyylipohja on tekniikka, jolla määritellään ulkoasu HTML-kielellä kuvatulle rakenteelle. Yksinkertaistettuna HTML kuvailee asiakirjan sisällön ja CSS määrittelee sisällön ulkoasun (W3schools, 2012). CSS:stä käytimme versioita CSS2 ja CSS3. jQuery on JavaScript-kirjasto, joka yksinkertaistaa HTML-asiakirjojen liikkumista, tapahtumien käsittelyä, animaatiointia ja Ajax interaktiota nopean web-kehityksen saavuttamiseksi. Käytimme jQuerysta versiota 1.7.2 ja jQuery UI:sta versiota 1.8.20.

Käyttäjäpuolella käytimme JavaScript-komentosarjakieltä. Tietokanta luotiin web-palveluihin suunnatulla MySQL- relaatiotietokantaohjelmistolla, jonka päälle rakennettava ohjelmalogiikka tehtiin web-palveluiden tuottamiseen tarkoitetulla PHP-ohjelmointikielellä. MySQL-tietokantaa hallinnoitiin palvelimella phpMyAdmin hallintatyökalulla.

Pienten kuvien (kuva 6, kuvat numeroissa 1 ja 2) luontiin käytimme Adobe Illustrator -vektorikuvankäsittelyohjelmaa, joka on suunniteltu pienten kuvien luomiseen, kuten ikonien luontiin.

Testauksessa käytimme testiautomaation luontiin Selenium IDE -sovellusta ja kuormitustestauksessa Apache JMeter –sovellusta. Testiautomaatioiden ajamiseen eri web-selaimilla ja raportointiin käytimme Selenium server –sovellusta ja kuormitusraportin luontiin Apache Ant –sovellusta. Nämä kaikki ovat avoimen lähdekoodin sovelluksia. Näistä kerromme lisää kappaleessa 3.3 Tietojärjestelmän testaus.

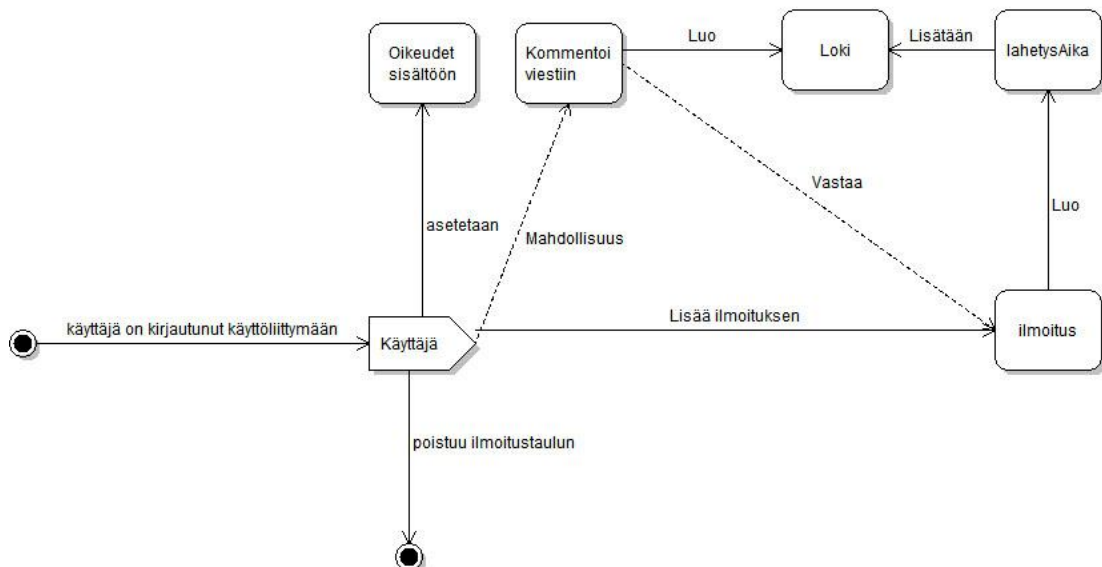
3 PROJEKTIN VAIHEET

3.1 Tietojärjestelmän suunnittelu

Tässä osiossa kuvaamme rakennepohjien pohjalta ilmoitustaulun käyttöliittymän toimintaa. Luvussa 3.1.1 kuvaamme käyttäjän roolia ilmoitustaulussa. Luvussa 3.1.2 kuvailemme ilmoitustaulun käyttöliittymää ja rakennetta. Luvussa 3.1.3 kuvaamme ilmoitusten lähettämistä ilmoitustauluun. Luvussa 3.1.4 kuvaamme tietokannan rakennetta ja luvussa 3.1.5 käyttötapauksia.

3.1.1 Käyttäjä

Tässä osiossa kuvaamme käyttäjän roolia ilmoitustaulussa kirjautumisesta saakka käyttöliittymästä uloskirjautumiseen. Mustat pallot kuvaavat lähtö ja lopputilanteita. Suorat viivat kuvaavat ehdottomia etenemisiä ja katkoviivat ehdollisia etenemisiä.



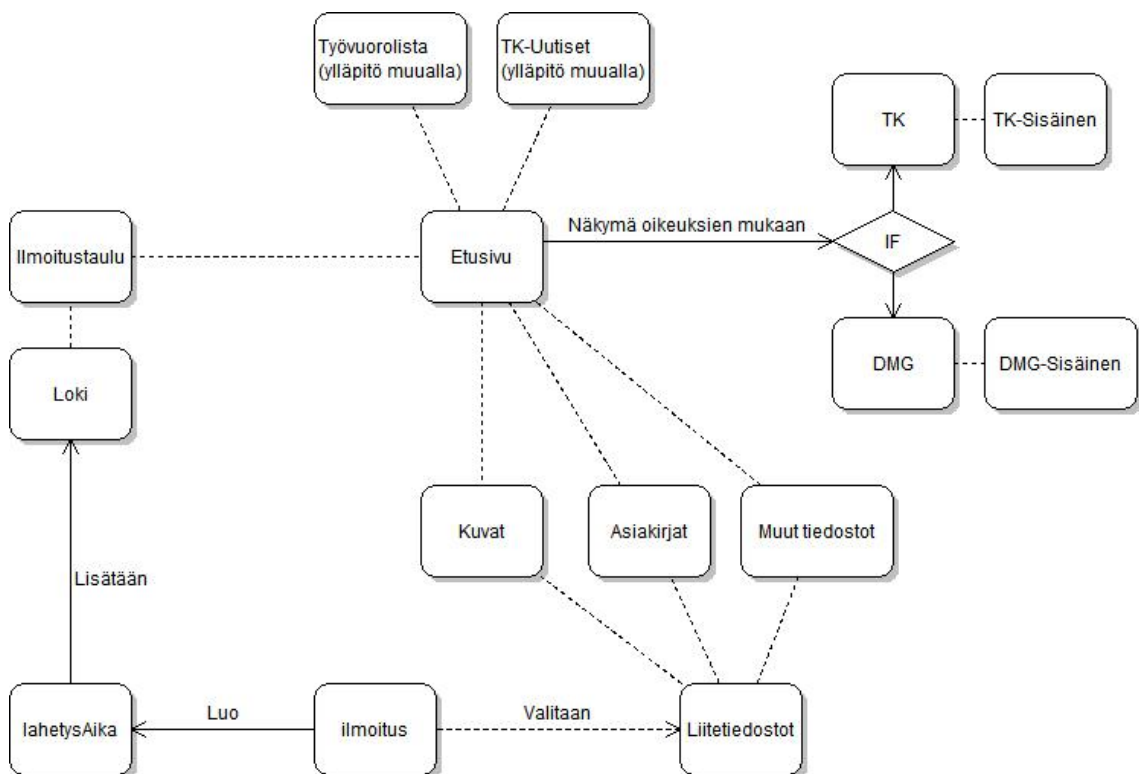
Kaavio 1. Käyttäjä ilmoitustaulussa.

Ilmoitustaulun käyttäjän tulee olla kirjautunut käyttöliittymään, jotta hän pystyy käyttämään ilmoitustaulua. Data Media Gazella kirjautuminen tapahtuu intranettiin kirjautuessa.

Käyttäjän avatessa ilmoitustaulun tarkistetaan hänen käyttäjäoikeutensa ilmoitustaulun sisältöön. Käyttöoikeudet määritetään jokaiselle käyttäjälle erikseen. Jokaisella käyttäjällä on myös nimikirjaimet, jotka koostuvat etu- ja sukunimen ensimmäisistä kirjaimista. Käyttäjällä on oikeus lisätä ilmoitus, sekä vastata muiden käyttäjien ilmoituksiin.

3.1.2 Ilmoitustaulun selaaminen

Tässä osiossa kuvaamme ilmoitusten selaamista. Suorat viivat kuvaavat ehdottomia etenemisiä ja katkoviivat ehdollisia etenemisiä. IF-salmiakki kuvaa ehtorakennetta.



Kaavio 2. Ilmoitustaulun selaaminen.

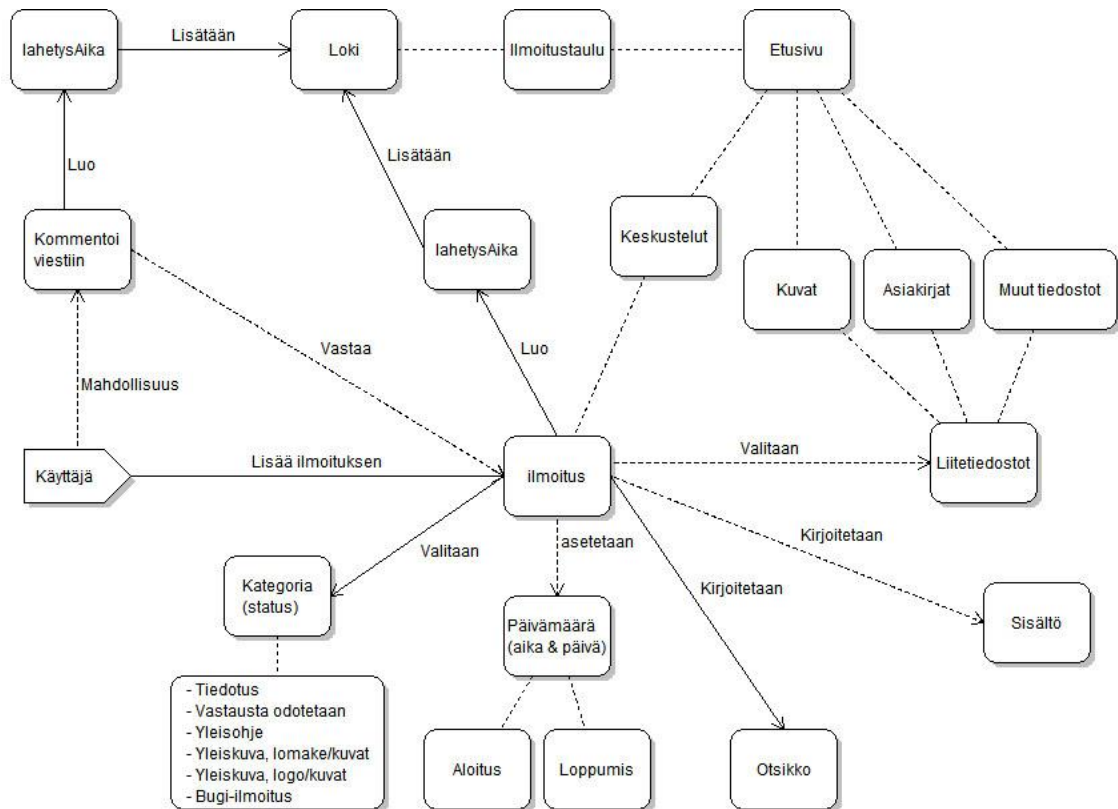
Käyttäjä näkee ilmoitustaulun ja siinä näkyvät ilmoitukset, joihin hänellä on oikeudet. Esimerkkitapaus ilmoitustaulun oikeuksista: käyttäjä työskentelee Tietokonekauppa.fi:ssä ja näin ollen hänellä on oikeudet Tietokonekauppa.fi:n ilmoituksiin.

Ilmoitustaulu koostuu seuraavista välisivuista: päänäkymä, liitetiedostot ja loki. Ilmoitustaululla käyttäjät voivat luoda ilmoituksia, joita toinen käyttäjä voi kommentoida. Nimikirjaimia käytetään keskustelussa käyttäjien identifiointiin. Lokiin kirjataan käyttäjien ilmoitukset. Etusivun näkymä vaihtelee käyttäjäoikeuksien mukaan. Käyttäjän etusivun ilmoitusten näkymä voi olla Tietokonekauppa.fi (TK), Data Media Gazelle (DMG) tai käyttäjän oikeudet voi antaa käyttäjälle oikeuden nähdä molemmat ilmoitukset.

Etusivun näkymä on jaettu osiin, joissa ilmoitukset ja liitetiedostot näkyvät erillisinä alueina. Etusivulle on varattu myös alue Tietokonekaupan viesteille ja työvuorolistalle, jotka ovat molemmat ulkoisia, eli niiden ylläpito tapahtuu muualla, kuin ilmoitustaululla.

3.1.3 Ilmoituksen lisääminen

Tässä osiossa tutustumme ilmoituksen lisäämiseen. Kuvaamme miten käyttäjä lisää ilmoituksen ilmoitustauluun. Suorat viivat kuvaavat ehdottomia etenemisiä ja katkoviivat ehdollisia etenemisiä.



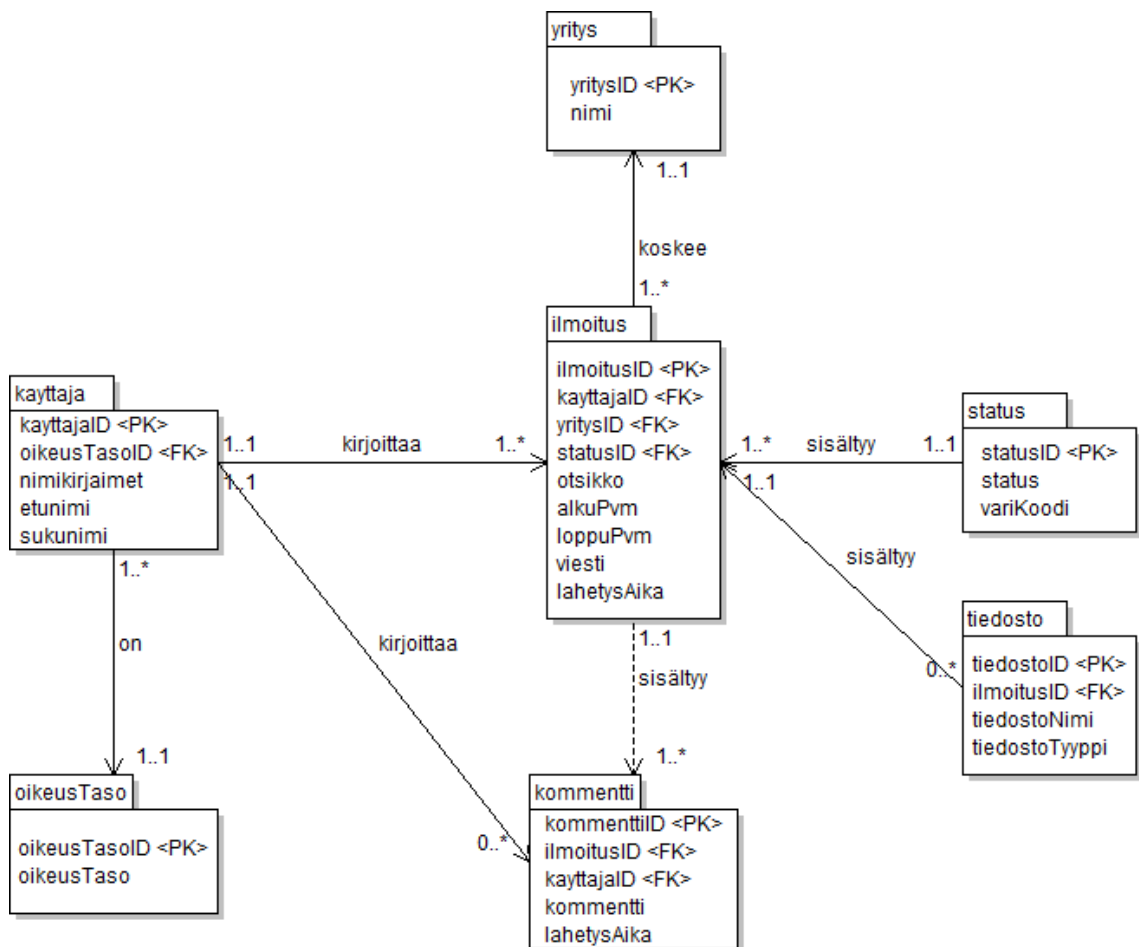
Kaavio 3. Ilmoituksen lisääminen.

Käyttäjä luo ilmoitustauluun ilmoituksen, johon hän määrittää ilmoituksen sisällön. Ilmoitus voi sisältää liitetiedoston, joita voi olla yhdessä ilmoituksessa useampi. Liitetiedostot erotellaan toisistaan etusivulla käyttämällä tiedostojen tyyppiä päätettäväksi esimerkiksi jpg- ja gif-tiedostot näkyvät kuvat-osiossa ja doc- ja pdf-tiedostot asiakirjat-osiossa. Ilmoitukseen asetetaan ilmoituksen aloitus- ja loppumispäivämäärä. Ilmoitukseen kirjoitetaan ilmoituksen otsikko, sekä ilmoituksen sisältö, eli viesti. Sekä ilmoituksen lisäyksestä, että viestiin vastaamisesta muodostuu aikaleima, jota hyödynnetään myöhemmin ilmoitustaulun lokissa.

Ilmoitukset kirjataan lokiin. Ilmoituksen lähetyksen yhteydessä tietokantaan kirjataan lähetyksaika, jota käytetään lokissa.

3.1.4 Tietomallin rakenne

Tässä osiossa kuvaamme tietojärjestelmän tietomallin rakennetta. Jokaisella taululla on attribuutteja ja jokaisella attribuutilla on oma tietotyyppi. Tietotyypit määrittävät, mitä tietoa kenttiin voidaan syöttää. TEXT tai VARCHAR -muotoiseen tietotyyppiin voidaan syöttää vain tekstimuotoista tietoa, kun taas INT tai NUMBER -muotoiseen tietotyyppiin voidaan syöttää vain numeromuotoista tietoa.



Kaavio 4. Järjestelmän tietomalli.

Jokaisella taululla pitää olla pääavain, joka yksilöi taulun tiedot. Käytimme projektissa pääavaimena useimmiten juoksevaa numeroa. Taulut yhdistetään toisiinsa viiteavaimilla. Viite ja pääavaimen pitää olla samaa tietotyyppiä.

Kayttaja-taulu on taulu järjestelmän toteutusta varten, johon tallentuvat kaikki järjestelmää käyttävät käyttäjät. Data Media Gazellella on kuitenkin oma käyttäjäkohtainen kirjautuminen ja tämä sähköinen ilmoitustaulu tulee olemaan vain osa heidän isompaa järjestelmää, joten myöhemmin tätä taulua ei tulla luultavasti tarvitsemaan. Toinen vaihtoehto on, että Kayttaja-taulu integroidaan Data Media Gazellen käyttäjille tarkoitetun taulun kanssa.

	Sarake	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Lisätiedot
<input type="checkbox"/>	kayttajaID	int(10)			Ei		auto_increment
<input type="checkbox"/>	oikeusTasoID	int(1)			Ei		
<input type="checkbox"/>	nimikirjaimet	varchar(3)	utf8_general_ci		Ei		
<input type="checkbox"/>	etunimi	varchar(50)	utf8_general_ci		Ei		
<input type="checkbox"/>	sukunimi	varchar(50)	utf8_general_ci		Ei		
<input type="checkbox"/>	kayttajatunnus	varchar(20)	utf8_general_ci		Ei		
<input type="checkbox"/>	salasana	varchar(20)	utf8_general_ci		Ei		

Taulukko 1. Kayttaja-taulun rakenne.

Kayttaja-taulun yksilöi pääavain kayttajaID, joka on juokseva numero. Kenttä oikeusTasoID on viiteavain taulun oikeusTaso pääavainkentälle oikeusTasoID. Kenttään nimikirjaimet tallentavat käyttäjän nimikirjaimet. Kentät etunimi ja sukunimi ovat käyttäjän nimitiedon tallennusta varten.

OikeusTaso-taulu on tarkoitettu järjestelmän käyttöoikeuksien tallennukseen. Kenttä oikeusTasoID yksilöi kunkin käyttöoikeustason ja kenttä oikeusTaso kertoo mistä käyttöoikeudesta on kyse.

Sarake	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Lisätiedot
oikeusTasoID	int(1)			Ei		
oikeusTaso	varchar(20)	utf8_general_ci		Ei		

Taulukko 2. OikeusTaso-taulun rakenne.

Käyttöoikeustasoja tulee olemaan neljä: pomo dmg, työntekijä dmg, pomo tk, työntekijä tk. Käyttöoikeustasojen idea on, että päälliköllä on enemmän oikeuksia kun työntekijällä.

Ilmoitus-taulu on taulu, johon ilmoitustaulun lähetetyt ilmoitukset tietoisineen tallentuvat. Ilmoitus-taulun yksilöi pääavain ilmoitusID, joka on juokseva numero. Kenttä kayttajaID on viiteavain Kayttaja-taulun kayttajaID pääavainkentälle. Kenttä yritysID on viiteavain Yritys-taulun yritysID pääavainkentälle. Kenttä statusID on viiteavain taulun status pääavainkentälle statusID.

Sarake	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Lisätiedot
<u>ilmoitusID</u>	int(10)			Ei		auto_increment
kayttajaID	int(10)			Ei		
yritysID	int(10)			Ei		
statusID	int(10)			Ei		
otsikko	varchar(50)	utf8_general_ci		Ei		
alkuPvm	text	utf8_general_ci		Kyllä	NULL	
loppuPvm	text	utf8_general_ci		Kyllä	NULL	
viesti	text	utf8_general_ci		Ei		
lahetysAika	text	utf8_general_ci		Ei		

Taulukko 3. Ilmoitus-taulun rakenne.

Kenttään otsikko tallentuu ilmoituksen otsikko. Kenttiin alkuPvm ja loppuPvm tallentuu ilmoituksen voimassaoloaika. Kenttään viesti tallentuu itse kirjoitettu ilmoitus. Kenttä lahetysAika on tarkoitettu jokaisen ilmoituksen aikaleiman tallennusta varten.

Tiedosto-taulu on luotu monien eri tiedostojen tallennusta varten. Yhtä ilmoitusta kohden voi olla monia eri tiedostoja.

Sarake	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Lisätiedot
<u>tiedostoID</u>	int(10)			Ei		auto_increment
ilmoitusID	int(10)			Ei		
tiedostoNimi	text	utf8_general_ci		Ei		
tiedostoTyyppi	varchar(20)	utf8_general_ci		Kyllä	NULL	

Taulukko 4. Tiedosto-taulun rakenne.

Kenttä tiedostoID on taulun pääavain, joka on juokseva numero. Kenttä ilmoitusID on viiteavain Ilmoitus-taulun pääavainkentälle ilmoitusID, eli näin nähdään mihin ilmoitukseen kyseinen tiedosto kuuluu. Kenttään tiedostoNimi tallentuu tiedoston nimi. Itse tiedosto tallennetaan web-käyttöliittymän kautta ja se ei tal-

lennu tietokantaan. Kenttä tiedostoTyyppi kertoo, mikä tiedostopääte tiedostolla on.

Status-taulu on tarkoitettu säilömään viestin eri statukset ja statuksien värikoodit. Eri statuksia ovat tiedotus, vastausta odotetaan, yleisohje, yleiskuva (lomake/kuvat), yleiskuva (logo/kuvat), bugi-ilmoitus. Tiedotus-statuksella tarkoitetaan ilmoitusta, jolla on aloitus ja lopetuspäivämäärä. Ilmoitus arkistoidaan ja se on luettavissa myöhemmin. Vastausta odotetaan –statuksella tarkoitetaan ilmoitusta, joka on tarkoitettu pikaisiin kysymyksiin, joihin odotetaan vastausta. Yleisohje-status tarkoittaa ilmoitusta, joka sisältää yleisohjeistuksia muille käyttäjille. Yleiskuva, lomake/kuvat –statuksella tarkoitetaan ilmoitusta, jonka ensisijaisena tehtävänä on pitää sisällään liitetiedostoja, riippumatta liitetiedostojen formaatista. Yleiskuva, logo/kuvat –statuksella tarkoitetaan ilmoitusta, joka pitää sisällään kuvatiedostoja, mutta myös muita kuvankäsittelyyn liittyviä formaatteja, kuten esim psd ja fla. Bugi-ilmoitus-statuksella tarkoitetaan ilmoitusta, joka pitää sisällään virheilmoituksia, koskien verkkosivuja ja intranetiä.

Sarake	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Lisätiedot
statusID	int(10)			Ei		auto_increment
status	varchar(30)	utf8_general_ci		Kyllä	NULL	
variKoodi	varchar(20)	utf8_general_ci		Kyllä	NULL	

Taulukko 5. Status-taulun rakenne.

Kenttä statusID on taulun pääavain. Jokaisella kuudella statuksella on oma ID. Kenttä status kertoo mistä statuksesta on kyse ja kenttään variKoodi on annettu jokaiselle statukselle oma väri, jota käytetään ilmoitustaulun päänäkymässä värittämään kyseisen statuksen omaava ilmoitus.

Yritys-taulu on tarkoitettu listaamaan ne yritykset joiden käytössä järjestelmä on. Tässä työssä huomioon otetaan vain kaksi yritystä, Data Media Gazelle ja Tietokonekauppa.fi, mutta erillinen yritys-taulu on silti hyvä olla olemassa.

	Sarake	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Lisätiedot
<input type="checkbox"/>	yritysID	int(10)			Ei		
<input type="checkbox"/>	nimi	varchar(100)	utf8_general_ci		Ei		

Taulukko 6. Yritys-taulun rakenne.

Kenttä yritysID on taulun pääavain, joka on lyhenne yrityksen nimestä. Kenttä nimi sisältää yrityksen koko nimen.

Kommentti-taulu on tarkoitettu ilmoitusten mahdollisia kommentteja varten. Yhtä ilmoitusta kohden voi olla monia kommentteja eri käyttäjiltä.

	Sarake	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Lisätiedot
<input type="checkbox"/>	kommenttiID	int(10)			Ei		auto_increment
<input type="checkbox"/>	ilmoitusID	int(10)			Ei		
<input type="checkbox"/>	kayttajaID	int(10)			Ei		
<input type="checkbox"/>	kommentti	text	utf8_general_ci		Ei		
<input type="checkbox"/>	lahetysAika	text	utf8_general_ci		Ei		

Taulukko 7. Kommentti-taulun rakenne.

Kenttä kommenttiID on taulun pääavain ja juokseva numero. Kenttä ilmoitusID on viiteavain taulun Ilmoitus pääavainkentälle ilmoitusID. Kenttä kayttajaID on viiteavain taulun kayttaja pääavainkentälle kayttajaID. Kenttä kommentti on tarkoitettu itse kommentin kirjoitusta varten. Kenttä lahetysAika on tarkoitettu jokaisen kommentin aikaleiman tallennusta varten.

3.1.5 Käyttötapaukset

Käyttötapaukset kertovat, mitä eri asioita järjestelmässä voidaan tehdä. Käyttötapauksen tarkoitus olla hyvin kuvaavia. Seuraavassa on listattuna järjestelmän käyttötapaukset.

Ilmoituksen selaus -käyttötapaus kuvaa ilmoitusten ja liitetiedostojen selausta ilmoitustaululla.

Nimi	Ilmoitustaulun selaus
Suorittajat	Käyttäjä
Esiehdot	Käyttäjä on kirjautunut järjestelmään Ilmoituksia on lisätty ilmoitustauluun.
Kuvaus	1. Käyttäjä näkee ilmoitustaululle lähetetyt ilmoitukset auki olevassa ylemmässä välilehdessä ja pääsee selaamaan niitä avaamalla ilmoituksen ponnahtusikkuna. Käyttäjä näkee kaikki lähetetyt liite-

	tiedostot alemmassa välilehdessä ja pääsee selaamaan niitä.
Poikkeukset	P1: Käyttäjä ei ole kirjautunut järjestelmään oikeilla tunnuksilla. P2: Tiedosto ei avaudu.
Lopputulokset	Käyttäjä pääsee onnistuneesti selaamaan ilmoitustaululle lähetettyjä ilmoituksia sekä liitetiedostoja.

Taulukko 8. Ilmoitustaulun selaus -käyttötapa.

Poikkeuksissa on mainittuna kohta, että käyttäjä ei ole kirjautunut järjestelmään oikeilla tunnuksilla. Järjestelmässä ei ole rekisteröinti-mahdollisuutta, joten työntekijä ei voi itse päättää tunnuksiaan.

Ilmoituksen lisäys -käyttötapa kuvaa uuden ilmoituksen kirjoittamista ja onnistunutta tallennusta tietokantaan.

Nimi	Ilmoituksen lisäys
Suorittajat	Käyttäjä
Esiehdot	Käyttäjä on kirjautunut järjestelmään
Kuvaus	<ol style="list-style-type: none"> Käyttäjä painaa linkistä "Lisää ilmoitus" jolloin ponnahdusikkuna ilmoituksen syötölle avautuu. Käyttäjä täyttää kentät lomakkeeseen: Otsikko, Viestin status (Tiedotus, Vastausta odotetaan, Yleisohje, Yleiskuva, loma-ke/kuvat, Yleiskuva, logo/kuvat, Bugi-ilmoitus) Aloitus pvm (Tiedotus-statuksessa), Lopetus pvm (Tiedotus-statuksessa), Viesti, Tiedosto (ei vaadittu). Käyttäjä lisää ilmoituksen järjestelmään painamalla "Lähetä".
Poikkeukset	P1: Käyttäjä ei ole täyttänyt kaikkia pakollisia kenttiä (otsikko, alkuPvm (Tiedotus-statuksessa), loppuPvm (Tiedotus-statuksessa), viesti) ilmoituksen lisäyslomakkeesta.
Lopputulokset	Käyttäjä lisää ilmoituksen onnistuneesti ilmoitustaululle.

Taulukko 9. Ilmoituksen lisäys -käyttötapa.

Järjestelmä huolehtii myös itse poikkeuskohdasta P1, sillä ilmoituksen syötössä on oikeellisuustarkistukset, joiden avulla ilmoitusta ei pysty lähettämään, jos kaikkia pakollisia kenttiä ei ole täytetty.

Ilmoitukseen kommentointi -käyttötapa kuvaa kommentin kirjoittamista ilmoitukseen ja onnistunutta tallennusta tietokantaan.

Nimi	Ilmoituksen kommentointi
Suorittajat	Käyttäjä
Esiehdot	Käyttäjä on kirjautunut järjestelmään. Ilmoituksia on lisätty ilmoitustauluun.
Kuvaus	<ol style="list-style-type: none"> Käyttäjä painaa ilmoitustaulun auki olevalta ylemmältä välilehdeltä haluttua ilmoitusta, jolloin ilmoitus avautuu kokonaisuudessaan ponnahdusikkunaan. Käyttäjä painaa linkkiä "Lisää kommentti", jolloin avautuu kom-

	mentointilomake. 3. Käyttäjä täyttää kommentointilomakkeen kentät (kommentti). 4. Käyttäjä lisää kommentin ilmoitukseen painamalla "Lähetä"
Poikkeukset	P1: Käyttäjä ei ole täyttänyt kaikki pakollisia kenttiä (kommentti) kommentointilomakkeesta.
Lopputulokset	Käyttäjä lisää kommentin ilmoitukseen onnistuneesti.

Taulukko 10. Ilmoituksen kommentointi -käyttötapa.


Järjestelmä huolehtii myös itse poikkeuskohdasta P1. Kommentin syötössä kommentti-kenttää ei voi jättää tyhjäksi ja tästä huolehtii järjestelmän oikeellisuustarkistus.

3.2 Tietojärjestelmän toteutus

Tässä osiossa kuvaamme ilmoitustaulun toteutusta ja käyttämiämme toteutusmenetelmiä. Luvussa 3.2.1 kuvaamme käyttäjän kirjautumista ilmoitustauluun. Luvussa 3.2.2 kuvailemme ilmoitustaulun päänäkymää, jossa lähetetyt ilmoitukset näkyvät. Luvussa 3.2.3 kuvaamme ilmoitusten lähettämistä ilmoitustauluun. Luvussa 3.2.4 kuvaamme kommentin lisäystä ilmoitukseen. Luvussa 3.2.5 kuvaamme käyttöliittymän loki-osiota ja luvussa 3.2.6 kuvailemme tapoja, joilla pyrimme luomaan käyttäjäystävällisen käyttöliittymän.

3.2.1 Järjestelmään kirjautuminen

Tässä osiossa kuvaamme miten käyttäjän kirjautuminen ilmoitustauluun tapahtuu, sekä käyttöliittymän ulkoasua ja toimivuutta.



Kirjaudu Dynaamiseen ilmoitustauluun !

Käyttäjätunnus

Salasana

Kuva 3. Kirjautuminen ilmoitustauluun.

Ilmoitustauluun sisäänpääsy vaatii käyttäjätunnuksen sekä salasanan, joilla käyttäjä pystyy kirjautumaan ilmoitustauluun kirjautumislomakkeen kautta. Käyttäjä täyttää lomakkeeseen tunnuksensa ja salasanansa, jonka jälkeen hän kirjautuu ilmoitustauluun Kirjaudu-painikkeella. Käyttäjätunnus- ja salasana -lomakekentät sisältävät SQL-injektioilta suojautumiseen tarvittavat mekanismit, jotka suoritetaan kun käyttäjä painaa vihreää Kirjaudu-painiketta.

SQL-injektio on koodi-injektiotekniikka, joka käyttää hyväkseen web-sivun ohjelmiston tietoturvaavaoittuvuuksia.

MySQL injektioilta voidaan suojautua käyttäen lomakekentissä, joihin käyttäjä voi kirjoittaa, funktioita `stripslashes` ja `mysql_real_escape_string`. `Stripslashes` poistaa käyttäjän syötteistä kenoviivat ja `mysql_real_escape_string` poistaa erikoismerkit merkkijonosta, joita voidaan käyttää SQL-lauseissa. Tämä tekee mahdolliseksi esimerkiksi haitallisen käyttäjä-taulun sisällön katsomisen SQL-lauseen kautta. `Mysql_real_escape_string` komento vaikuttaa seuraaviin merkkeihin:

- `\x00`
- `\n`
- `\r`
- `\`

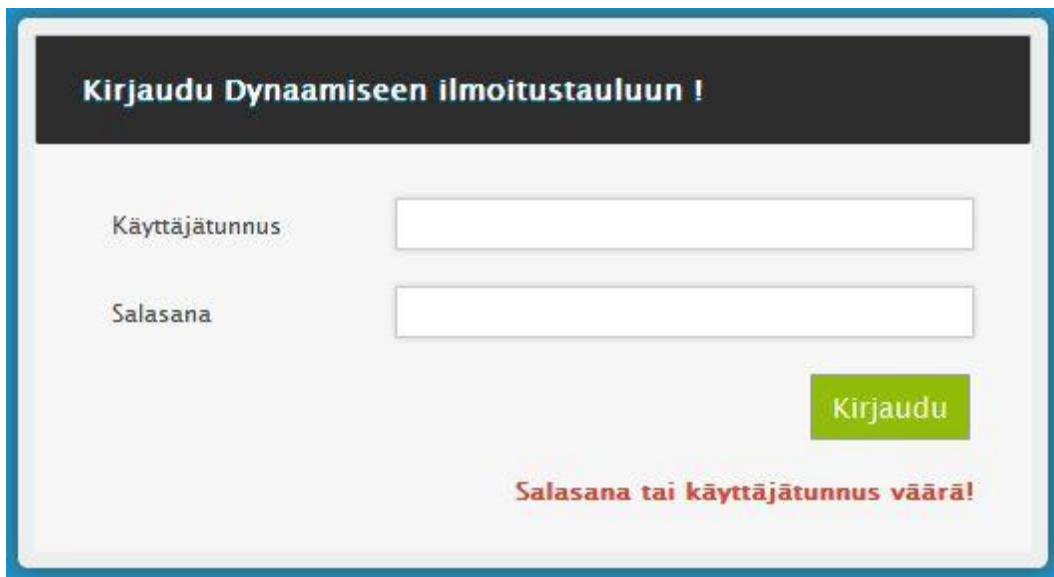
- '
- "
- \x1a

(W3schools 2011.)

MySQL injektioilta suojautumiseen käytimme tässä tapauksessa seuraavanlaisia koodia:

```
$kunnus = stripslashes($kunnus);
$salasana = stripslashes($salasana);
$kunnus = mysql_real_escape_string($kunnus);
$salasana = mysql_real_escape_string($salasana);
```

\$kunnus ja \$salasana ovat muuttujia, jotka on annettu kirjautumis-sivun HTML-lomakekentistä käyttäjätunnukseksi ja salasanalle.



Kirjaudu Dynaamiseen ilmoitustauluun !

Käyttäjätunnus

Salasana

Kirjaudu

Salasana tai käyttäjätunnus väärä!

Kuva 4. Virheilmoitus kirjautuessa.

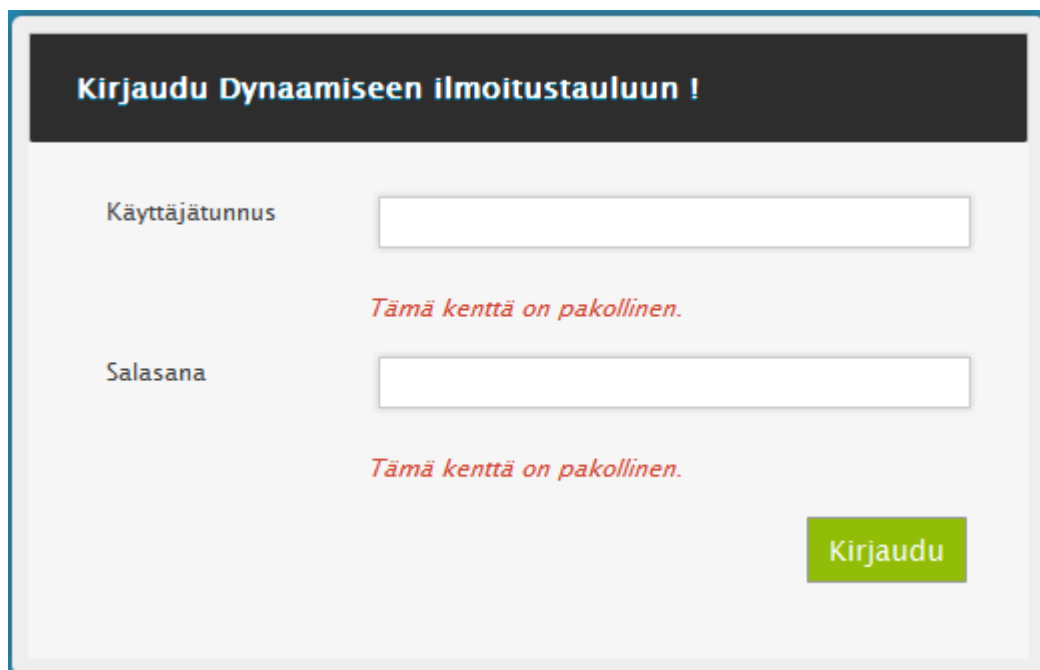
Käyttäjän kirjautuessa ilmoitustauluun on mahdollista, että hän syöttää, joko käyttäjätunnuksensa tai salasanansa väärin. Käyttäjätunnuksen tai salasanan ollessa väärä ilmoitustauluun ilmestyy virheilmoitus ”Salasana tai käyttäjätunnus väärä!”. Virheilmoitus laukeaa Kirjaudu-painikkeen aktivoitessa tarkista_kirjautuminen.php, jossa tapahtuu käyttäjän tunnuksen ja salasanan tarkistus. Tarkistuksen epäonnistuessa päivittyy kirjautuminen.php komennolla:

```
header("Location: kirjautuminen.php?error=1");
```

Koodi muuttaa sivun kirjautuminen.php osoitteen lisäämällä ?error=1-päätteen, joka aktivoi kirjautuminen.php:n koodista php-koodin:

```
if($_GET['error']==1){
    $error = '<span id="error" class="error" style="color:
    #D14836; font-weight: bold;">Salasana tai käyttäjätun-
    nus väärä!</span> ';
}
```

Joka taas tulostaa kirjautumislomakkeessa olevan muuttujan \$error:n, joka näkyy käyttäjälle virheilmoituksena.



The image shows a login form titled "Kirjaudu Dynaamiseen ilmoitustauluun !". It contains two input fields: "Käyttäjätunnus" and "Salasana". Below each field is a red error message: "Tämä kenttä on pakollinen." (This field is required). A green "Kirjaudu" button is located at the bottom right of the form.

Kuva 5. Virheilmoitus, pakolliset kentät ovat tyhjiä..

Käyttäjän yrittäessä kirjautua jättämällä Käyttäjätunnus- tai Salasana -kentän tyhjäksi laukeaa virheilmoitus "Tämä kenttä on pakollinen (Kuva 5)". Virheilmoitus laukeaa käyttämällä jQuery Validation -lisäosasta versiota 1.9.0, joka laukee kirjautuessa seuraavan jQuery koodin:

```
$(document).ready(function(){
    $(".kirjautuminen_form").validate();
});
```

Koodi tekee oikeellisuustarkistuksen lomakkeesta, jossa on luokka nimeltä kirjautuminen_form. Oikeellisuustarkistus laukaisee virheilmoituksen ”kirjautuessa” tekstikentistä, joissa on määritetty luokka nimeltä required, joka on yksi jQuery Validationin ennalta määritetyistä komennoista.

Seuraavassa taulukossa on selitettynä sisäänkirjautumissivun kenttien ja virheilmoitusten merkitykset:

Käyttäjätunnus	Syöttökenttä johon syötetään käyttäjän käyttäjätunnus.
Salasana	Syöttökenttä johon syötetään käyttäjän salasana.
Kirjaudu	Kirjaudu-painike aktivoi käyttötunnuksen ja salasanan tarkistuksen, jolloin käyttäjä joko saa virheilmoituksen tai hän kirjautuu ilmoitustauluun.
Salasana tai käyttäjätunnus väärä!	Virheilmoitus, joka tulee siinä tapauksessa, jos käyttäjätunnus tai salasana on väärin kirjautuessa.
Tämä kenttä on pakollinen.	Virheilmoitus, joka tulee siinä tapauksessa, jos käyttäjä jättää syöte- rivit käyttäjätunnus tai salasana tyhjäksi kirjautuessa.

Taulukko 11. Ilmoitustauluun kirjautumisen kentät

Uloskirjautuminen ilmoitustaulusta tapahtuu ilmoitustaulun päänäkymän oikeasta yläkulmasta sijaitsevasta (Kuva 6, numero 2) ”Kirjaudu ulos”-painikkeesta, jossa on punainen nuoli, joka tulee ulos ovesta. Kuvakkeessa on pyritty ottamaan huomioon käyttäjät käyttämällä tuttua kuvasymboliikkaa. Kirjaudu ulos -painike aktivoi uloskirjautuminen.php:n, jossa se aktivoi istunnon sulkevan komennon:

```
session_destroy();
```

3.2.2 Ilmoitustaulun päänäkymä

Ilmoitustaulun päänäkymää varten on suunniteltu ulkoasu, joka noudattaa luotuja CSS-määrittelyitä. Yläosan painikkeet sekä Ilmoitukset ja Liitetiedostot -osiot on luotu käyttäen hyväksi jQuery-tyylejä. Avattaessa Liitetiedostot-välilehti, Il-

moitukset-välilehti menee piiloon ja sama toimii myös toisinpäin.

The screenshot shows a web application interface. At the top, there is a header with the names 'Matti' and 'Loki'. A green circle with the number '1' highlights a notification icon in the header. On the right side, there is a red circle with the number '2' highlighting a close button. Below the header, there is a section titled 'Työvuorolista' (Work schedule) showing a grid of names for days of the week (Ma, Ti, Ke, To, Pe, La). To the right, there is a section titled 'Ilmoitukset' (Notifications) containing a table with columns: Aihe (Subject), Lähettäjä (Sender), Kommentit (Comments), and Lähetysaika (Send time). Below the table, there is a section titled 'Liitetiedostot' (Attachments).

Aihe	Lähettäjä	Kommentit	Lähetysaika
Lokin toiminta	Jouko Uotinen		06.05.2012, 03:02:10
Muita tiedostoja	Matti Moisio		05.05.2012, 23:19:14
Kuvatiedostoja	Matti Moisio		03.05.2012, 03:10:57
testataan värejä 2	Matti Moisio		03.05.2012, 02:45:44
testataan värejä 1	Matti Moisio		03.05.2012, 02:45:08
Tiedostoja liitteenä	Matti Moisio		23.04.2012, 01:58:49
Ääkköset ei toimi	Matti Moisio		22.04.2012, 17:17:56
Tiedostotesti monella tiedostolla	Matti Moisio		19.04.2012, 04:42:41

Kuva 6. Ilmoitustaulun päänäkymä ilmoitukset-välilehti avoinna.

Ilmoitukset tulostuvat ruudulle niin, että ilmoitustaulun päänäkymässä näkyy vain ilmoituksen aihe, lähettäjän nimi, kommenttien lukumäärä ja lähetysaika. Otsikkoa painamalla näkyviin saadaan koko ilmoitus. Ilmoituksen kokonaisnäkyvässä, ilmoitusta voidaan myös kommentoida. Jokaisen kommentoijan kohdalla näkyy lähettäjän nimi, kommentti ja lähetysaika. Ilmoitukset ovat jaoteltu statuksen värikoodin mukaan.

Ilmoitustaulun päänäkymän vasemmassa ylä laidassa näkyy kirjautuneen käyttäjän etunimi. Käyttäjän kirjautuessa, käyttäjän ID, etunimi ja sukunimi tallennetaan sessioon ja tätä kautta käyttäjän etunimi tulostetaan ruudulle seuraavalla php-koodilla:

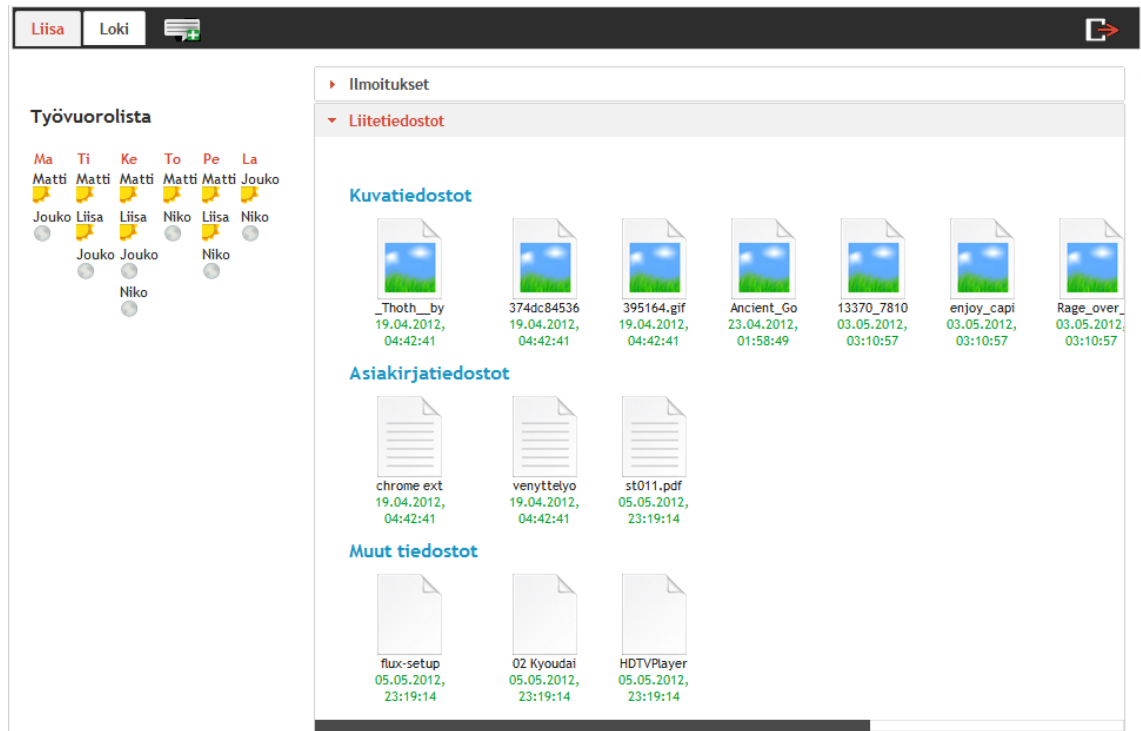
```
echo $_SESSION["etunimi"];
```

Ilmoitusten kohdalla käyttäjän etunimi ja sukunimi on yhdistetty näkyväksi kentäksi Lähettäjä, joka onnistuu käyttäen hyväksi HTML-taulukkoa ja tulostamalla php:n kautta istuntotiedot etunimelle ja sukunimelle.

Ilmoitustaulun päänäkymässä painamalla yksittäisen ilmoituksen otsikkoa, avautuu ilmoitus jQuery ponnahdusikkuna -muodossa. Ponnahdusikkuna sisältää kentät statukselle, ilmoituksen aiheelle, lähettäjälle, viestille, mahdollisille liitetiedostoille ja kommenteille. Ilmoitukset latautuvat ponnahdusikkunaan tietokannasta seuraavalla jQuery koodilla:

```
dialog.load(  
  
url, // url-osoitteeseen ankkuri href-attribuutista  
  
},  
  
function(responseText, textStatus, XMLHttpRequest) {  
// onnistunut callback  
  
$( ".dialog" ).dialog( "open" ); // näytä dia-  
logi  
  
});
```

Edellä mainittu koodi etsii avattavasta ilmoituksesta ilmoituksen url-osoitteen, joka sisältää uniikin tunnistusnumeron, joka toimii tietokannasta pääavaimena ja täten auttaa löytämään viestin sisältämän datan.



Kuva 7. Ilmoitustaulun päänäkymä liitetiedostot-välilehti avoinna.

Ilmoitustaulun liitetiedostot-välilehdellä näkyy ilmoituksien sisältämät liitetiedostot lähetyksjärjestyksessä. Käyttäjä pystyy selaamaan liitetiedostoja vierityspalkilla, halutessaan etsiä vanhempia tiedostoja. Liitetiedostot on jaettu kolmeen osaan: Kuvatiedostot, asiakirjatiedostot ja muut tiedostot. Kuvatiedostot sisältävät kaikki tiedostot, jotka sisältävät kuvia, kuten esim. jpg- ja gif -tiedostot. Asiakirjat sisältävät tekstitiedostoja/dokumentteja, kuten esim. txt- ja doc -tiedostot. Muut tiedostot sisältävät tiedostoja, joita ei ole määritetty kuvatiedostoiksi tai asiakirjatiedostoiksi.

3.2.3 Ilmoituksen lisäys






Ilmoitus lisätään ilmoitustauluun painamalla ilmoitustaulun päänäkymässä Lisää ilmoitus- painiketta (Kuva 6, numero 1), joka aukaisee ponnahtusikkunan (Kuva 8).

Kuva 8. Ponnahdusikkuna, jossa käyttäjä lisää ilmoituksen.

Käyttäjä voi liikuttaa ponnahdusikkunaa yläosan mustasta alueesta sekä suurentaa ponnahdusikkunan kokoa oikeasta alalaidasta.

Käyttäjä täyttää ja valitsee lomakkeesta seuraavat kohdat:

Otsikko	Otsikko kohtaan kirjoitetaan ilmoituksen otsikko, joka kuvaa lyhyesti ilmoituksen viestin sisältöä. Pakollinen kenttä.
Viestin status	Viestin statuksesta ilmenee mihin kategoriaan lähetettävä viesti kuuluu. Käyttäjää huomioiden, jokainen status on määritetty eri värillä, jotta statusien hahmottaminen ilmoitustaulusta olisi mahdollisimman helppoa. Viestien statukset ovat seuraavat: <ul style="list-style-type: none"> • Tiedotus on ilmoitus, jolla on aloitus ja lopetus päivämäärä. Ilmoitus arkistoidaan ja se on luettavissa myöhemmin. Tiedotus status väri: ■ • Vastausta odotetaan on ilmoitus, joka on tarkoitettu pikaisesti ky-

	<p>symyksiin, joihin odotetaan vastausta. Vastausta odotetaan status väri: </p> <ul style="list-style-type: none"> • Yleisohje on ilmoitus, joka sisältää yleisohjeistuksia muille käyttäjille. Yleisohje status väri:  • Yleiskuva, lomake/kuvat on ilmoitus, jonka ensisijainen tehtävä on pitää sisällään liitetiedostoja, riippumatta liitetiedostojen formaatista. Vastausta Yleiskuva, lomake/kuvat status väri:  • Yleiskuva, logo/kuvat on ilmoitus, joka pitää sisällään kuva tiedostoja, mutta myös muita kuvankäsittelyyn liittyviä formaatteja, kuten esim. psd ja fla. Yleiskuva, logo/kuvat status väri:  • Bugi-ilmoitus sisältää virheilmoituksia, koskien verkkosivuja tai intranetiä. Bugi-ilmoitus status väri: 
Aloituspvm	Aloituspvm kohtaan määritetään päivämäärä, jolloin ilmoitusta käsittelevä ”aihe” alkaa. Aloitus pvm:n asetus on mahdollinen vain tiedotusstatuksessa, muutoin se on piilossa ilmoituksen lisäys -lomakkeesta. Aloitus päivämäärän voi lisätä, joko manuaalisesti tai valita jQuery-kalenterista, joka tulee esiin painamalla Aloitus pvm -tekstikenttää. Valittaessa Tiedotusstatus tämä on pakollinen kenttä.
Lopetus pvm:	Lopetus pvm kohtaan määritetään päivämäärä, jolloin ilmoitus loppuu, eli milloin ilmoitus ei ole enää ajankohtainen. Lopetus pvm:n asetus on mahdollinen vain tiedotusstatuksessa, muutoin se on piilossa ilmoituksen lisäyslomakkeesta. Lopetus päivämäärän voi lisätä joko manuaalisesti tai valita jQuery-kalenterista, joka tulee esiin painamalla Lopetus pvm -tekstikenttää. Valittaessa Tiedotusstatus tämä on pakollinen kenttä.
Viesti:	Viesti kohtaan kirjoitetaan viesti, jossa kerrotaan mitä ilmoituksella halutaan ilmoittaa.
Tiedosto:	Tiedosto kohtaan käyttäjä voi lisätä halutessaan liitetiedostoja. Liitetiedostot näkyvät listattuna ja ne voi poistaa painamalla punaista X-kuvaa.
Lähetä	Lähetä-painiketta painamalla käyttäjä lähettää ilmoituksen nähtäväksi muille käyttäjille. Lähetä-painikkeen väriksi valittiin vihreä, joka yhdistetään vapaa kulku -merkkiin liikenteessä (Color Wheel Pro, 2012). Lähetä/kirjautu sisään -painikkeiden värit ovat yhdenmukaisia käyttöliittymässä.
Peruuta	Peruuta-painike sulkee ponnahdusikkunan ja samalla peruttaa ilmoituksen. Peruuta-painikkeen väriksi valittiin punainen, joka yhdistetään stop-merkkiin liikenteessä (Color Wheel Pro, 2012). Peruuta/sulje/tyhjennä-painikkeiden värit ovat yhdenmukaisia käyttöliittymässä.
X	Punainen X-painike ponnahdusikkunan oikeassa yläkulmassa sulkee ponnahdusikkunan ja samalla peruttaa ilmoituksen.

Taulukko 12. Ilmoituksen lisäys – ponnahdusikkunan kentät.

Käyttäjän painettua lähetä-painiketta ponnahdusikkuna sulkeutuu ja ilmoitustaulun päänäkyvä latautuu uudestaan ja käyttäjän lähettämä ilmoitus näkyy listalla ensimmäisenä.

3.2.4 Kommentin lisäys

Kommentit ovat listattuna ilmoituksen alapuolella ja itse kommentissa näkyy aina lähettäjän nimi, kommentti ja lähetysaika. Painamalla kohdasta ”Lisää

kommentti”, ponnahtusikkunan koko kasvaa pystysuunnassa ja esille aukeaa tekstikenttä kommentin syöttämiselle.

Ilmoitus ✖

Status: Bugi-ilmoitus

Aihe: Ääkköset ei toimi

Lähtettäjä: Matti Moisio

Viesti:

AlkuPvm: 22.04.2012

LoppuPvm:

Tiedostot:

Kommentit:

Lähtettäjä: Liisa Mäki
Kommentti: Kiitos bugin korjauksesta
Lähetysaika: 14.05.2012, 17:55:31

Lähtettäjä: Matti Moisio
Kommentti: Nyt toimii
Lähetysaika: 22.04.2012, 17:18:15

+ Lisää kommentti

Nimi: Liisa Mäki

Kommentti:

Lähetä **Tyhjennä**

Sulje

Kuva 9. Ilmoitustaulun ilmoitusnäkö lisää kommentti näkö aukaistuna.

Käyttäjä voi liikuttaa ponnahdusikkunaa yläosan mustasta alueesta sekä suurentaa ponnahdusikkunan kokoa oikeasta alalaidasta.

Kommentin lähettäjän nimi tulee automaattisesti sen perusteella, kuka on kirjautunut järjestelmään. Kommentin pystyy lähettämään painamalla vihreää Lähetä-painiketta, jonka jälkeen ponnahdusikkuna sulkeutuu.

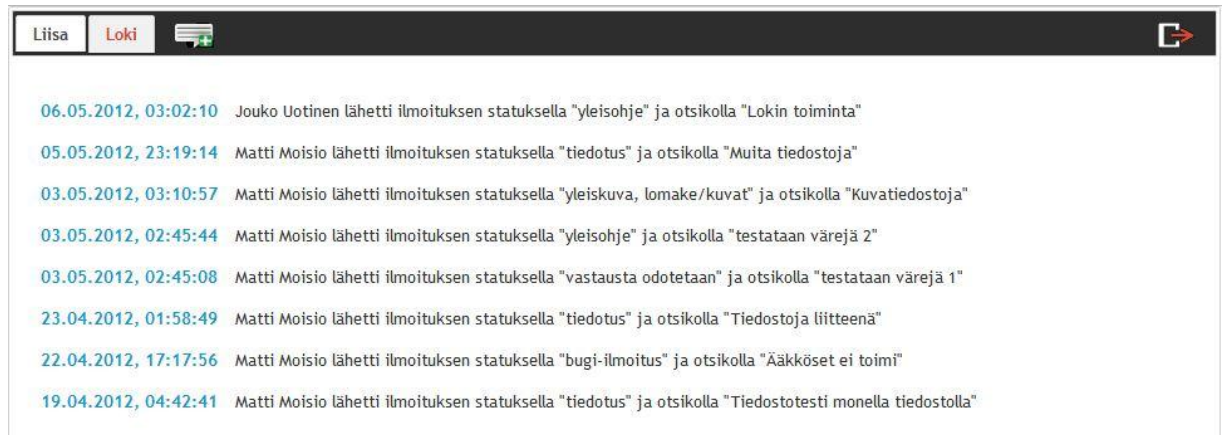
Lisää kommentti	Lisää kommentti avaa piilotetun alueen, joka aukaisee kommentin lisäys -alueen. Painikkeessa on plus-merkki vihreällä pohjalla, jotta painikkeen hahmottaminen olisi käyttäjälle helpompaa.
Nimi	Käyttäjän nimi, joka on kirjautunut ilmoitustauluun.
Kommentti	Tekstikenttä, johon kirjoitetaan kommentti. Pakollinen kenttä.
Lähetä	Lähetä-painiketta painamalla käyttäjä lähettää kommentin ilmoitukseen. Lähetä painikkeen väriksi valittiin punainen, joka yhdistetään vapaa kulku -merkkiin liikenteessä (Color Wheel Pro, 2012). Lähetä/kirjautu sisään -painikkeiden värit ovat yhdenmukaisia käyttöliittymässä.
Tyhjennä	Tyhjennä-painikkeen väriksi valittiin punainen, joka yhdistetään stop-merkkiin liikenteessä (Color Wheel Pro, 2012). Peruuta/sulje/tyhjennä-painikkeiden värit ovat yhdenmukaisia käyttöliittymässä.
Sulje	Sulje-painikkeen väriksi valittiin punainen, joka yhdistetään stop-merkkiin liikenteessä (Color Wheel Pro, 2012). Peruuta/sulje/tyhjennä-painikkeiden värit ovat yhdenmukaisia käyttöliittymässä.

Taulukko 13. Kommentin lisäys ilmoitustauluun – ponnahdusikkunan kentät.

Ilmoitus-ponnahdusikkunassa kommenttialueen korkeus on määritetty 350 pikseliin. Jos ponnahdusikkunan korkeus ylittää 350 pikseliä, vierityspalkki ilmestyy, jolla käyttäjä voi selata ilmoituksen materiaalia ponnahdusikkunasta.

3.2.5 Ilmoitustaulun tapahtumaloki

Ilmoitustauluun kuuluu loki, josta nähdään lähetetyt ilmoitukset sekä minä päivänä ja mihin kellonaikaan ilmoitus on lähetetty ja kuka sen on lähettänyt. Lokiin päästään painamalla kohdasta Loki, kirjautuneen käyttäjän vierestä.



Kuva 10. Ilmoitustaulun tapahtumaloki.

Lokissa näkyy sinisellä värillä päivämäärä ja kellonaika, koska tietty tapahtuma on tapahtunut ja mustalla värillä itse tapahtumakuvaus. Tapahtumakuvaus muodostuu ilmoituksen lähettäneestä käyttäjästä, toimenpiteestä, ilmoituksen statuksesta ja ilmoituksen otsikosta.

3.2.6 Ilmoitustaulun oikeellisuustarkistus

Kävimme luvussa 3.2.1 Järjestelmään kirjautuminen läpi oikeellisuustarkistusta kirjautuessa ilmoitustauluun. Tässä luvussa käymme läpi oikeellisuustarkistuksesta ilmoitustaulussa. Käytimme oikeellisuustarkistukseen jQuery Validation -lisäosasta versiota 1.9.0, johon muokkasimme oman oikeellisuustarkistuksen, joka hyväksyy vain kirjaimia, numeroita, pilkkuja ja pisteitä. Käyttämämme oikeellisuustarkistus jQuery-koodi on seuraava:

```
jQuery.validator.addMethod("merkkiValidointi", function(value, element) {
    return this.optional(element) || /^[a-z0-9-.,\s]+$/i.test(value);
}, "<br /> Käytä vain kirjaimia, numeroita, pilkkuja ja pisteitä");
```

Virheilmoitus laukeaa ilmoitusta lisätessä jQuery Dialog -pönnahdusikkunan sisällä olevasta määrittelystä, jossa tarkistus on sidottu "Lähetä"-painikkeeseen:

```
"Lähetä": function(){  
  
    if ($("#ilmoitus").validate().form() == true) { // tarkistaa onko vaaditut  
        kentät täytetty  
  
        document.ilmoitus.submit();  
  
        $( this ).dialog( "close" );  
  
    },
```

Virheilmoitus näkyy syötekentissä, joissa on määritetty luokka nimeltä merkki-Validointi. Käyttäjän lisätessä syötekenttiin virheellisiä merkkejä, aktivoituu virheilmoitus *"Käytä vain kirjaimia, numeroita, pilkkuja ja pisteitä"* lähetyksen yhteydessä (Kuva 11). jQuery Validation -lisäosa tarkistaa myös pakolliset kentät, jotka ovat jääneet tyhjiksi.

Lisää ilmoitus
✖

Otsikko:

Käytä vain kirjaimia, numeroita, pilkkuja ja pisteitä

Viestin status:
 ▼

Aloituspvm:

Lopetus pvm:

Viesti:

Käytä vain kirjaimia, numeroita, pilkkuja ja pisteitä

Tiedosto:
 No file chosen

Kuva 11. Virheilmoitus väärin merkkien lisäyksestä otsikkoon ja viestiin.

Pakolliset kentät tarkistetaan lisäämällä syötekenttään luokka nimeltä required. Pakollisten kenttien tarkistus laukeaa myös "Lähetä"-painikkeesta antamalla virheilmoituksen "Tämä kenttä on pakollinen." (Kuva 12). Pakolliset kentät ovat ilmoitusta lisättäessä otsikko, aloitus pvm ja lopetus pvm. Aloitus- ja lopetus päivämäärää kysytään vain valittaessa tiedotus-status, muutoin kentät ovat piilotettuina ja aktivoitu pois käytöstä, joten virheilmoitukset ilmestyvät vain valittaessa tiedotus-status.

Lisää ilmoitus ✕

Otsikko:

Kirjoita vähintään 3 merkkiä.

Viestin status:
 ▼

Aloituspvm:

Lopetus pvm:

Tämä kenttä on pakollinen.

Viesti:

Tiedosto:
 No file chosen

Kuva 12. Virheilmoitus liian lyhyestä otsikosta ja tyhjästä pakollisesta kentästä.

Ilmoitusta lisättäessä tarkastetaan myös otsikon pituus, joka on määritetty vähintään kolmen merkin pituiseksi. Otsikon pituus tarkistetaan käyttäjän painaessa "Lähetä"-painiketta ja otsikon pituuden ollessa alle kolme merkkiä aktivoituu virheilmoitus *"Kirjoita vähintään 3 merkkiä."*

Ilmoituksen lisäyksen oikeellisuustarkistuksesta siirrymme ilmoituksen kommentoinnin oikeellisuustarkistukseen, joka aktivoituu jQuery-koodilla:

```
$(document).ready(function(){
```

```
$(".lisaa_kommentti_form").validate();  
  
});
```

Koodi tarkistaa luokan nimeltä `lisaa_kommentti_form`, joka löytyy kommentti lomakkeesta.

Nimi: Matti Moisio

Kommentti:

Tämä kenttä on pakollinen.

Lähetä

Tyhjennä

Kuva 13. Virheilmoitus yrityksestä lähettää tyhjä kommentti.

Kommenttia lisätessä ilmoitukseen, käyttäjä ei voi lähettää tyhjää kommenttia, vaan "Lähetä"-painikkeesta aktivoituu oikeellisuustarkistus, joka antaa virheilmoituksen "Tämä kenttä on pakollinen" (Kuva 13) syötekentästä, johon on määritetty luokka nimeltä `required`.

Nimi: Matti Moisio

Kommentti:

Käytä vain kirjaimia, numeroita, pilkkuja ja pisteitä

Lähetä

Tyhjennä

Kuva 14. Virheilmoitus värierojen lisäyksestä kommenttiin

Kommentin lisäyksessä käytetään myös syötettyjen merkkien tarkistusta käyttämällä tekstikentässä luokkaa nimeltä merkkiValidointi, joka aktivoituu ”Lähetä”-painikkeesta antamalla virheilmoituksen ”*Käytä vain kirjaimia, numeroita, pilkkuja ja pisteitä*” (Kuva 14).

3.2.7 Käyttöliittymän arviointi

Käyttöliittymää suunnitellessa otimme huomioon Nielsenin heuristiset säännöt, johon viittasimme luvussa 2.4 Käyttäjätavallinen käyttöliittymä. Nielsenin heurististen sääntöjen pohjalta pyrimme luomaan mahdollisimman yksinkertaisen käyttöliittymän, jossa käyttäjän kanssakäyminen käyttöliittymän kanssa olisi mahdollisimman helppoa ja miellyttävää.

Käyttäjän vuorovaikutuksessa käyttöliittymään pyrimme käyttämään värejä, jotka herättäisivät mielikuvia, kuten esimerkiksi lomakkeiden painikkeissa, jossa käytimme vihreätä ja punaista. Vihreää käytimme lähetä ja kirjaudu -painikkeissa, joka yhdistetään liikenteen vapaaseen kulku-merkkiin (Color Wheel Pro, 2012). Peruuta-painike sulkee ponnahdusikkunan ja samalla peruttaa ilmoituksen. Peruuta, sulje ja tyhjennä -painikkeiden väriksi valitsimme taas

punaisen, joka yhdistetään liikenteen stop-merkkiin (Color Wheel Pro, 2012). Punaista käytimme myös virheilmoituksissa. Kuvat yritimme pitää mahdollisimman selkeinä ja mielikuvia herättävinä, kuten esimerkiksi ”kirjautu ulos” -painike (Kuva 6, numero 2), jossa käytimme symbolina ovea, josta lähtee punainen nuoli ulos.

Käyttäjän muistin kuormituksen minimoimiseksi käytimme yhtenäisiä värejä lomakkeiden painikkeissa ja ilmoitustaulun virheilmoituksissa, lomakkeissa ja kirjautumisessa. Käyttöliittymän väreissä pyrimme pitämään selkeän minimaalisen värimaailman, jossa pääväreinä käytimme mustaa, valkoista ja harmaata, sekä punaista, vihreää ja sinistä toissijaisina väreinä. Toimenpiteet käyttöliittymässä suunnittelimme mahdollisimman yksinkertaisiksi, jotta käyttäjän olisi mahdollisimman helppo hahmottaa eri käyttöliittymän osa-alueet ja näin oppia käyttämään ilmoitustaulua nopeasti.

Käyttöliittymän reaaliaikaisuus näkyy esimerkiksi ilmoituksen lisäyksessä, jossa lähetetty ilmoitus näkyy ilmoitustaululla ensimmäisenä heti käyttäjän lähetettyä sen. Palautetta käyttäjä taas saa tehdessään esimerkiksi virheitä täyttäessään ilmoitusta, näin voi käydä esimerkiksi lähettämällä ilmoitus ilman otsikkoa, josta tulee virheilmoitus.

3.3 Tietojärjestelmän testaus

Tässä osiossa käymme läpi rakennetun tietojärjestelmän testausta ja kerromme miksi päädyimme juuri kyseisiin testimenetelmiin. Luvussa 3.3.2 esitämme eri testitapaukset. Luvussa 3.3.1 käymme läpi manuaalista testausta ja kerromme Scrum:sta. Luvussa 3.3.3 kuvaamme testiautomaation luontia. Luvussa 3.3.4 käymme läpi kuormitustestausta ja luvussa 3.3.5 tietoturvatestausta.

3.3.1 Manuaalinen testaus

Manuaalinen testaus tapahtuu jo järjestelmän kehitysvaiheessa yhteistyössä toteutuksen kanssa. Aina kun yksi järjestelmän moduuli valmistuu, se testataan ja löytyneet bugit korjataan, jonka jälkeen testataan uudestaan ja tätä jatketaan kunnes bugeja ei enää löydy. Lopussa järjestelmä testataan kokonaisuudessaan.

3.3.2 Testitapaukset

Seuraavassa on listattuna järjestelmän testitapaukset.

Ilmoituksen selaus -testitapaus kuvaa testattavat asiat koskien ilmoitusten selausta ilmoitustaululla.

Nimi	Ilmoitustaulun selaus
Suorittajat	Käyttäjä
Askeleet	<ol style="list-style-type: none"> 1. Testataan että käyttäjä ei pysty selaamaan ilmoitustaulua ilman että hän on kirjautunut järjestelmään. 2. Testataan että ilmoituksia pääsee selaamaan kokonaisuudessaan.
Lopputulos	Käyttäjä pääsee onnistuneesti selaamaan ilmoitustaululle lähetettyjä ilmoituksia sekä liitetiedostoja.

Taulukko 14. Ilmoitustaulun selaus -testitapaus.

Istunnon ollessa suljettuna, ei käyttäjällä tulisi olla oikeuksia päästä selaamaan ilmoitustaulua ilman kirjautumista.

Ilmoituksen lisäys -testitapaus kuvaa testattavat asiat koskien uuden ilmoituksen kirjoittamista ja onnistunutta tallennusta tietokantaan.

Nimi	Ilmoituksen lisäys
Suorittajat	Käyttäjä
Askeleet	<ol style="list-style-type: none"> 1. Testataan että ilmoituksen syöttölomakkeen kenttiin ei pysty kirjoittamaan haitallista koodia. 2. Testataan että ilmoituksen syöttölomakkeen kenttiä ei pysty jättämään tyhjäksi. 3. Testataan että ilmoituksen syöttölomakkeen päivämääräkenttiin ei pysty kirjoittamaan muuta kun päivämääriä. 4. Testataan että käyttäjä ei pysty lähettämään vääränmuotoisia liitetiedostoja. 5. Testataan että lähetetty ilmoitus ilmestyy järjestelmään.
Lopputulos	Käyttäjä lisää ilmoituksen onnistuneesti ilmoitustaululle.

Taulukko 15. Ilmoituksen lisäys -testitapaus.

Järjestelmän oikeellisuustarkistus pitää huolen, että tyhjiä kenttiä ei pysty jättämään, samoin kun haitallista koodia ei pysty suorittamaan, mutta nämä pitää testata käytännössä.

Ilmoitukseen kommentointi -testitapaus kuvaa testattavat asiat koskien kommentin kirjoittamista ilmoitukseen ja onnistunutta tallennusta tietokantaan.

Nimi	Ilmoituksen kommentointi
Suorittajat	Käyttäjä
Askeleet	<ol style="list-style-type: none"> 1. Testataan että ilmoituksen kommentointilomakkeen kenttiin ei pysty kirjoittamaan haitallista koodia. 2. Testataan että ilmoituksen kommentointilomakkeen kenttiä ei pysty jättämään tyhjäksi. 3. Testataan että ilmoituksen kommentti ilmestyy järjestelmään ja näkyy varsinaisen ilmoituksen alla.
Lopputulokset	Käyttäjä lisää kommentin ilmoitukseen onnistuneesti.

Taulukko 16. Ilmoituksen kommentointi -testitapaus.

Kommentoinnin osalta myös järjestelmän oikeellisuustarkistus pitää huolen, että tyhjiä kenttiä ei pysty jättämään ja haitallista koodia ei pysty suorittamaan, mutta nämä on testattava käytännössä.

3.3.3 Testiautomaation luonti

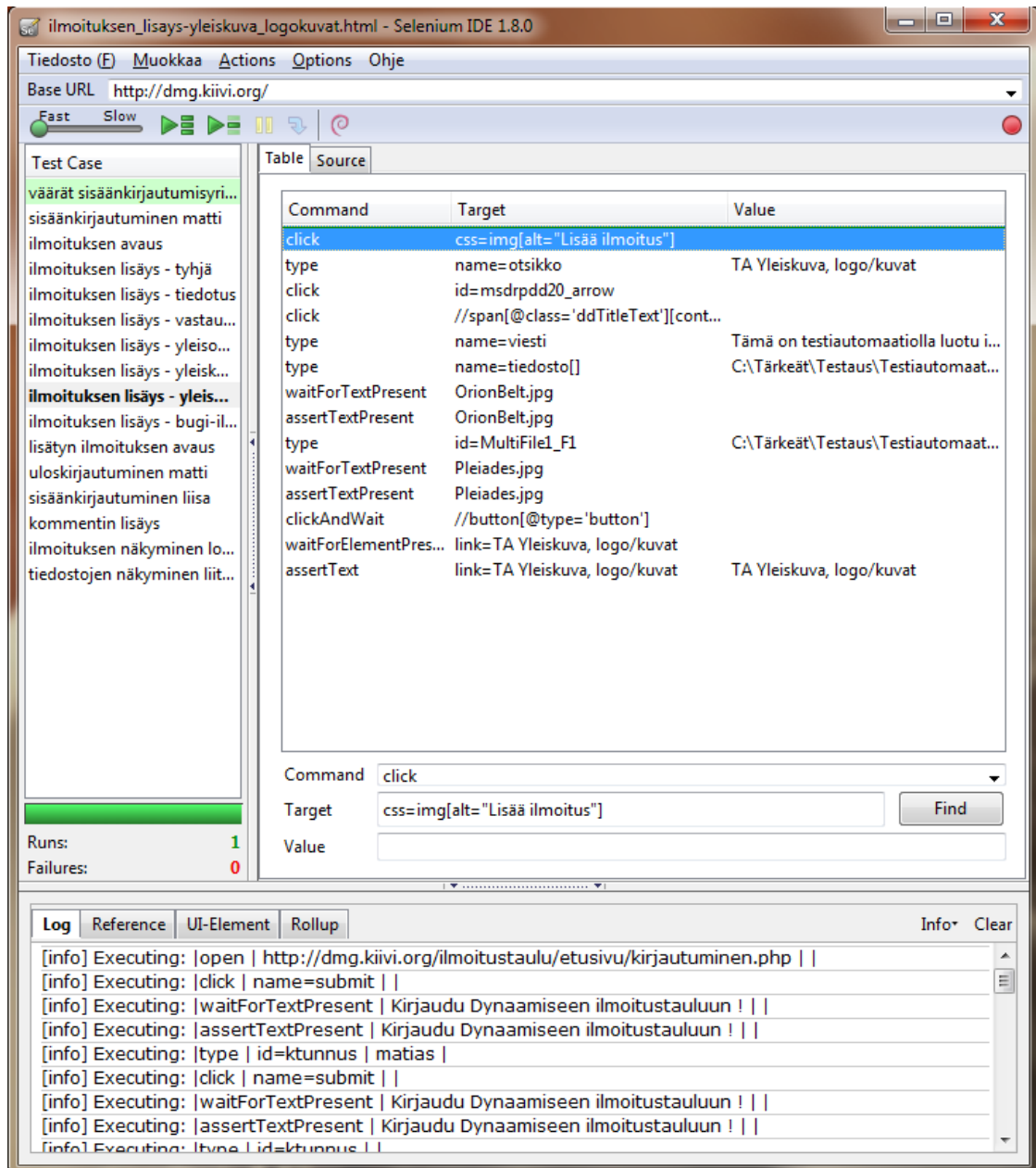
Testiautomaation tarkoitus ei ole löytää lisää bugeja, vaan saada toistettua regressioita, eli samoja asioita ei tarvitse joka vaiheessa testata aina uudelleen ja uudelleen manuaalisesti. Isoissa projekteissa pitää tarkkaan määritellä, mitä automatisoidaan ja mitä ei, koska itse prosessi on hyvin aikaa vievää ja aika on rahaa. Vaikka itse testiautomaation rakentaminen ei veisikään rutkasti aikaa, niin pitää huomata, että testiautomaatiota pitää päivittää jokaisessa vaiheessa kun järjestelmään tulee muutoksia. (Zallar 2000.)

Testiautomaatiosovellusta valittaessa olisi hyvä asia, jos sovelluksella pystyisi nauhoittamaan monet asiat suoraan, koska tämä säästää aikaa. Myös testiau-

tomaatiosovelluksen luoma koodi ei saisi olla hankalaa luettavaa, jolloin siihen onnistuu tekemään helposti manuaalisiakin muokkauksia. (Zallar 2000.)

Päädyimme testiautomaatiota luotaessa Selenium nimiseen ilmaisen lähdekoodin testiautomaatiosovellukseen. Käytimme Selenium-paketista Selenium IDE -sovellusta testitapausten nauhoittamiseen ja Selenium Server -sovellusta nauhoitettujen testitapausten ajamiseen eri web-selaimilla, sekä raportoinnin luontiin.

Selenium IDE on Mozilla Firefox -liitännäinen, jolla web-sovelluksen toimintaa voidaan nauhoittaa. Usein kuitenkin suorat nauhoitukset eivät toimi, sillä sovelluksessa voi olla käytössä elementeille dynaamiset ID:t ja tällöin automaatiot pitää kirjoittaa manuaalisesti käyttämällä hyväksi XPathia elementtien sijaintien löytämiseen.



Kuva 15. Selenium IDE käyttöliittymä.

Selenium IDE käyttöliittymässä vasemmalla sivulla näkyvät auki olevan testisarjan testitapaukset. Oikealla sivulla näkyy auki oleva testitapaus komentoineen. Alhaalla on loki, josta nähdään kokoajan mitä ollaan suorittamassa sekä Reference-välilehdeltä saadaan apua tietyn komennon kirjoittamiseen.

Rakensimme Selenium IDE:llä testiautomaatiot väärille sisäänkirjautumisyriyksille, jossa yritetään kirjautua sisälle tyhjällä tunnuksella ja salasanalla sekä

väärällä tunnuksella ja väärällä salasanalla. Testihenkilö Matin sisäänkirjautumiselle on oma testitapaus, samoin kuin jo olemassa olevan ilmoituksen avaukselle. Loimme automaation, jossa yritetään lähettää tyhjää ilmoitusta sekä automaatiot ilmoituksen lähettämiseksi kaikilla statuksilla ja yhden lähetetyn ilmoituksen avaamiselle. Yleiskuva, logo/kuvat ja yleiskuva, lomake/kuvat – ilmoituksissa lähetetään myös liitetiedostoja ilmoituksen mukana. Testihenkilö Matin uloskirjautumiselle on myös oma testitapaus, samoin kuin toisen testihenkilö Liisan sisäänkirjautumiselle. Liisa kirjoittaa yhtenä testitapauksena Matin lähettämään ilmoitukseen kommentin. Ilmoituksen näkymiselle lokissa on myös testitapaus, samoin kun lähetettyjen liitetiedostojen näkymiselle liitetiedostot-osiossa.

Suoritimme Selenium IDE:llä luodun testiautomaation Selenium Server-ohjelmalla Internet Explorer 9, Mozilla Firefox 12 ja Google Chrome 18 –web-selaimilla. Liitteenä (Liite 1) on raportti suorituksesta Mozilla Firefox 12 -web-selaimella. Raportista huomasimme, että tyhjän ilmoituksen lähetys -testitapaus oli epäonnistunut. Ilmoitustaulun ilmoitus-ponnahdusikkunassa oli alunperin javascript –oikeellisuustarkistukset, jotta tyhjää ilmoitusta ei pitäisi pystyä lähettämään, mutta ilmeisesti jQuery-ponnahdusikkuna ei tue normaalia javascript validointia, vaan sille on rakennettava jQuery-pohjaiset validoinnit. Tämän jälkeen rakensimme jQuery-pohjaiset validoinnit, joten tyhjää ilmoitusta ei pysty enää lähettämään.

3.3.4 Kuormitustestaus

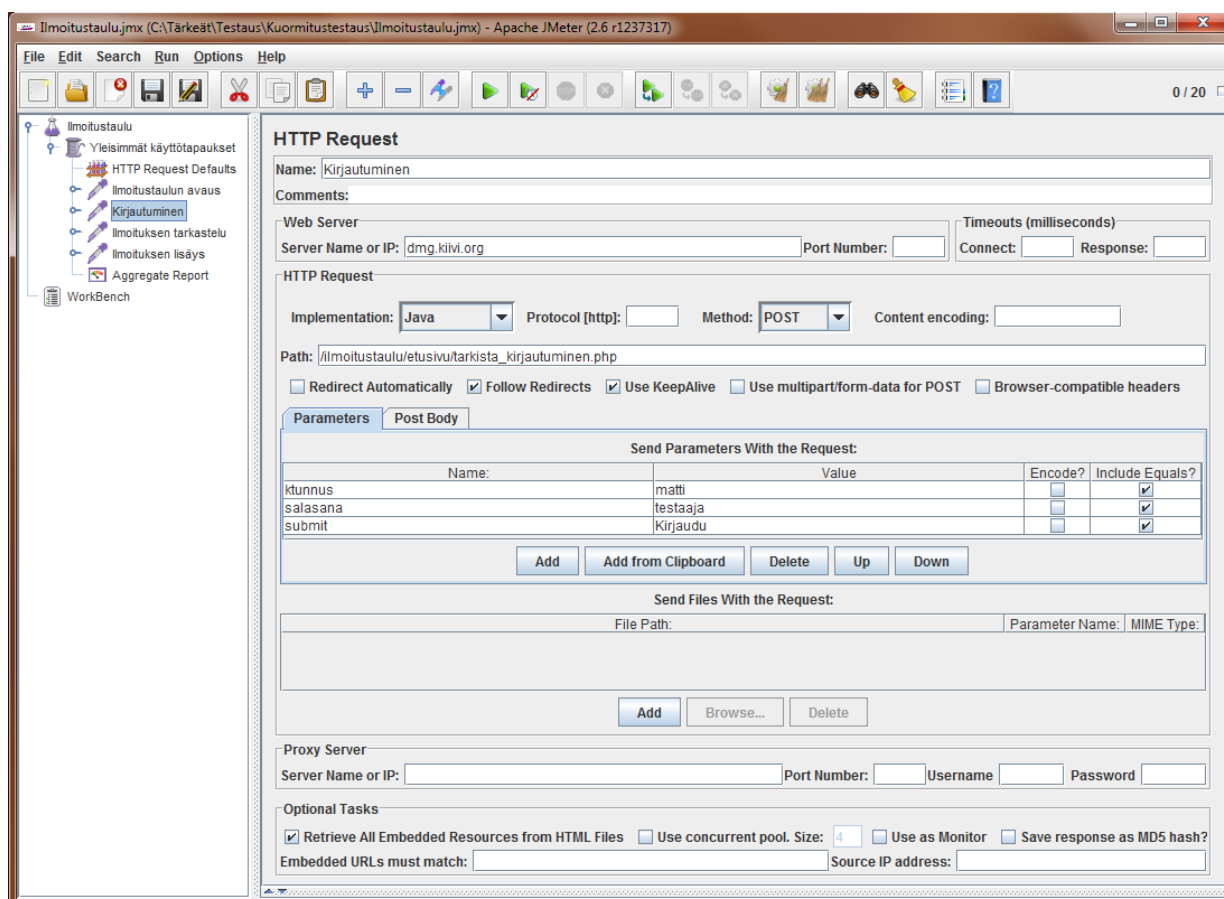
Kuormitustestauksen päätavoitteena on varmistaa, että järjestelmä toimii niin kuin sen on suunniteltu toimivan. Toinen tavoite on, että kaikki käyttäjät pystyvät käyttämään järjestelmää vasteaikojen ollessa kohtuullisia kaikista niistä käyttöliittymistä käsin, joista järjestelmän tulee toimia. (Mansikkamäki 2001.)

Kuormitustestauksella pääasiassa selvitetään, millä käyttäjämäärällä järjestelmä kuormittuu niin paljon, että se ei enää ole käytettävissä hyväksytyllä tasolla ja miten kauan järjestelmän käyttäjä joutuu odottamaan sivulatausten välillä se-

kä miten useampien käyttäjien aikaansaama kuorma vaikuttaa järjestelmän suorituskykyyn ja tietokantaan. (Mansikkamäki 2001.)

Tapauksessamme kuormitustestauksen rooli ei ole niin iso, koska käyttäjiä ovat vain työntekijät ja työntekijöitä ei ole niin paljoa, mutta on hyvä ottaa huomioon asioita, kuten työntekijöiden määrän kasvu.

Kuormitustestausta luotaessa päädyimme Apache JMeter nimiseen avoimen lähdekoodin kuormitustestaussovellukseen. Käytimme Apache JMeteriä käyttötapausten nauhoittamiseen ja ajoimme testit määrätyillä käyttäjämäärillä. Kuormitustestausraportin generoimiseen käytimme Apache Ant nimistä ohjelmaa, jonka integroimme toimimaan Apache JMeter ohjelman testien raportointisovelluksena.



Kuva 16. Apache jMeter.

Apache jMeter –ohjelman käyttöliittymässä vasemmalla sivulla ylimpänä näkyy testisuunnitelman nimi, jonka alle testitapaukset jaotellaan säiejoukkoihin (thread group). Säiejoukkojen hyöty on siinä, että niille saadaan määritettyä jokaiselle omat käyttäjien määrät ja suorituskerrat, samoin kun ramp-up period. Ramp-up period kertoo miten kauan kestää liikkua ylös täyteen käyttäjämäärään, esimerkiksi jos käyttäjien määrä on 10 ja ramp-up period on 100, eli $100/10 = 10$, joka tarkoittaa että joka kymmenen sekunnin välein yksi pyyntö lisää kohdistuu palvelimeen. Itse testitapauksia kutsutaan sillä nimellä mihin kohteeseen pyyntö kohdistuu, esimerkiksi HTTP Request kohdistuu web-sivuihin ja sovelluksiin ja FTP Request tiedostonsiirtopalvelimeen. Käyttöliittymän oikealla puolella näkyy auki olevan komponentin asetukset.

Testasimme kuormitusta 20:llä käyttäjällä, laittaen ramp-up periodin kahteen, jolloin heti alussa ei tule turhaa kuormitusta ja sitä kautta virheitä kun kaikki käyttäjät yrittävät tehdä kaiken samalla sekunnin murto-osalla. Kuormitustesti meni läpi ilman virheitä. Loimme kuormitustestistä raportin käyttäen Apache Ant-sovellusta (Liite 2).

3.3.5 Tietoturvatestaus

Käytimme ilmoitustaulun tietoturvan testaamiseen sovellusta Websecurify. Websecurify käy läpi automaattisesti web-sivun tai web-sovelluksen mahdollisia tietoturva-aukkoja silmällä pitäen. Websecurify sisältää tarkistukset ainakin tärkeimmistä OWASP TOP 10 –haavoittuvuuksista. OWASP TOP 10:een (2007) kuuluu seuraavat haavoittuvuudet: verkkosivun rakenne ei säily (XSS), taustajärjestelmäkyselyn rakenne ei säily, haitallisen tiedoston suoritus, turvaton suora viittaus tietoaalkioon, puutteellinen pyynnön alkuperän tarkistus (CSRF), tiedon vuotaminen ja puutteellinen virheenkäsittely, puutteellinen tunnistusmenetely ja istunnonhallinta, puutteellinen tietojen salaaminen, turvattomat tietoliikennetytydet, rajoittamaton URL-tason pääsy. (OWASP 2007.)

Websecurifyn luomasta raportista (Liite 2) kävi ilmi että ilmoitustaulussa on kolme haavoittuvuutta. Ilmoitus.php-tiedostossa, eli ilmoitus-ponnahdusikkunassa

on XSS-haavoittuvuus, joten kaikki käyttäjän lähettämät tiedot pitää saada tallentumaan tietokantaan turvallisemmassa muodossa. Toisena haavoittuvuutena ilmoitustaulu paljastaa käytössä olevan palvelimen ylläpitosovelluksen ja PHP:n version. Kolmantena haavoittuvuutena tiedostonsiirrossa on XSS-haavoittuvuus. Tätä haavoittuvuutta hyväksikäyttäen ilmoitustauluun on mahdollista lähettää haitallista koodia tiedostoina, koska tiedoston sisältöä ei tarkisteta ilmoituksen lähetysvaiheessa.

3.3.6 Toimeksiantajan suorittama testaus

Data Media Gazelle kävi läpi ja testasi manuaalisesti myös ilmoitustaulun toimintaa, jonka perusteella he tekivät päätöksen ottavatko he sitä käyttöön. Testauksessa he saivat kaadettua ilmoitustaulun lähettämällä ilmoituksen sisältönä haitallista koodia. Haitallisen koodin ja tyhjien ilmoitusten syöttömahdollisuus oli meillä tiedossa, koska emme tässä vaiheessa olleet vielä saaneet jQuery-oikeellisuustarkistuksia toimimaan. Opinnäytetyön valmistumisvaiheessa järjestelmään ei kuitenkaan enää pystynyt lähettämään tyhjiä ilmoituksia, eikä haitallista koodia.

4 YHTEENVETO

Tämän opinnäytetyön aiheen saimme toimeksiantona ja tarkoitus oli rakentaa uusi ilmoitustaulu mediayhtiö Data Media Gazellelle. Uuden ilmoitustaulun tarkoituksena oli korvata heidän käytössään ollut vanha, käyttöliittymän käytettävyydeltään huonolla tasolla ollut ilmoitustaulu. Etenimme Mikael Laaksosen työharjoittelussa tekemän ja Data Media Gazellen hyväksymän suunnitelman mukaisesti ja aloitimme työn syksyllä 2011. Ilmoitustaulun päätavoitteina oli voida lähettää ilmoituksia muiden työntekijöiden nähtäville ja ilmoituksiin piti olla mahdollista lisätä myös liitetiedostoja.

Tammikuun lopussa Data Media Gazelle selkeytti ja lisäsi ilmoitustauluun kohdistuvia vaatimuksia, mutta näitä seuraamalla ilmoitustaulu olisi näyttänyt täysin toisen näköiseltä, joten päätimme jatkaa jo hyvässä vaiheessa edennyttä projektia seuraten vanhoja vaatimuksia.

Ilmoitustaulu onnistui mielestämme suhteellisen hyvin ja noudatti alkuperäistä suunnitelmaa 95 prosenttisesti. Ilmoitustauluun kirjautuminen, oikeellisuustarkistukset sekä ilmoitusten lähettäminen eri statuksilla ja niiden näkyminen ilmoitustaululla onnistuivat, sekä myös liitetiedostojen lähetys ja lataus saatiin toimimaan. Ilmoitustaulun loki onnistui osittain: ilmoituksen näkyvät siinä, mutta myös liitetiedostojen ja ehkä myös kirjautumisten näkyminen siinä olisi ollut hyvä asia. Yritimme luoda pudotusvalikkoja, josta olisi voinut valita statuksen mukaan näytettävät ilmoitukset tulostumaan ilmoitustaululle tässä epäonnistuen, mutta syynä saattoi olla tämän osalta myös PHP:n ja jQuery:n yhteensovittamisongelma. Liitetiedostoille saimme tehtyä oman välilehden ja liitetiedostojen jaon kolmeen eri kategoriaan.

Valitettavasti Data Media Gazelle ei ottanut järjestelmää käyttöön, koska se ei vastannut heidän tarpeitaan.

Jatkoa ajatellen edellisessä kappaleessa mainittu pudotusvalikko olisi hyvä saada tehdyksi, sillä muuten yhdeltä sivulta ilmoitusten haku voi olla aikaa vie-

vää. Liitetiedostoihin olisi hyvä saada myös samanlainen pudotusvalikko hake-
maan esimerkiksi tietyinä päivinä lähetetyt liitetiedostot. Yksi isompi jatkokehi-
tysidea olisi tehdä ilmoitustaulusta älypuhelimille ja tablet-tietokoneille optimoitu
versio.

LÄHTEET

Color Wheel Pro. Viitattu 14.5.2012. -> Red. Saatavilla www-muodossa <http://www.color-wheel-pro.com/color-meaning.html>

Color Wheel Pro. Viitattu 14.5.2012. -> Green. Saatavilla www-muodossa <http://www.color-wheel-pro.com/color-meaning.html>

Dmgazelle.com. Viitattu 26.5.2012 -> Yritys. Saatavilla www-muodossa <http://dmgazelle.com/live/>

Wille Kuutti 2003, Käytettävyys, suunnittelu ja arviointi, Talentum

Mansikkamäki, R. 2001. Kuormitustestaus businessvaatimusten varmistajana. Systemityö 1/01.

OWASP. 2007. OWASP Top 10 for 2010. Viitattu 14.5.2012. Saatavilla www-muodossa https://www.owasp.org/index.php/Top_10_2007_Finnish

Tietokonekauppa.fi. Viitattu 26.5.2012 -> Yritystiedot. Saatavilla www-muodossa <http://www.tietokonekauppa.fi/?page=yhteystiedot>

W3schools. Viitattu 15.12.2011 -> haku: stripslashes(). Saatavilla www-muodossa http://www.w3schools.com/php/func_mysql_real_escape_string.asp

W3schools. Viitattu 15.12.2011 http://www.w3schools.com/php/func_string_stripslashes.asp

W3schools. Viitattu 26.5.2012 -> haku: css -> CSS Introduction. Saatavilla www-muodossa http://www.w3schools.com/css/css_intro.asp

W3schools. Viitattu 26.5.2012 -> haku: xhtml -> XHTML Introduction. Saatavilla www-muodossa http://www.w3schools.com/css/css_intro.asp

Wikipedia. Viitattu 26.10.2011 haku: Red -> symbolism. Saatavilla www-muodossa http://en.wikipedia.org/wiki/Red#cite_note-53

Yritystele.fi. Viitattu 26.10.201 -> haku: SH-Group Oy. Saatavilla www-muodossa. <http://www.yritystele.fi/node/78671>

Zallar, K. 2000. Practical Experience in Automated Testing, Automated Software Testing - A Perspective. Viitattu 26.10.2011 <http://www.methodsandtools.com/archive/archive.php?id=33>

Liite 1 Selenium testiraportti

Test suite results

result:	failed
totalTime:	61
numTestTotal:	16
numTestPasses:	15
numTestFailures:	1
numCommandPasses:	41
numCommandFailures:	1
numCommandErrors:	1
Selenium Version:	2.21
Selenium Revision:	.0
Test Suite	
väärät sisäänkirjautumisyritykset	
sisäänkirjautuminen matti	
ilmoituksen avaus	
ilmoituksen lisäys - tyhjä	
ilmoituksen lisäys - tiedotus	
ilmoituksen lisäys - vastausta odotetaan	
ilmoituksen lisäys - yleisohje	
ilmoituksen lisäys - yleiskuva, lomake/kuvat	
ilmoituksen lisäys - yleiskuva, logo/kuvat	
ilmoituksen lisäys - bugi-ilmoitus	
lisätyn ilmoituksen avaus	
uloskirjautuminen matti	
sisäänkirjautuminen liisa	
kommentin lisäys	
ilmoituksen näkyminen lokissa	
tiedostojen näkyminen liitetiedostot-välilehdellä	

Liite 2 jMeter testiraportti





Kuormitustestin tulokset - 20 käyttäjää

Suunniteltu käytettäväksi ohjelmilla [JMeter](#) ja [Ant](#).

Yhteenveto

Testien lukumäärä	Virheet	Onnistumisprosentti	Keskiaika	Minimiaika	Maksimiaika
80	0	100.00%	15446 ms	3752 ms	27102 ms

Testit

Testin nimi	Testien lkm	Virheet	Onnistumisprosentti	Keskiaika	Minimiaika	Maksimiaika	
Ilmoitustaulun avaus	20	0	100.00%	7537 ms	3752 ms	9925 ms	
Kirjautuminen	20	0	100.00%	22320 ms	19460 ms	27102 ms	
Ilmoituksen tarkastelu	20	0	100.00%	9604 ms	7610 ms	12943 ms	
Ilmoituksen lisäys	20	0	100.00%	22322 ms	17980 ms	25917 ms	

Liite 3 Websecurify testiraportti

Cross-site Scripting

Cross-site Scripting (XSS) is a type of web application security vulnerability which allows code injection by malicious web users into the web pages viewed by other users. An exploited XSS vulnerability could be used by attackers to bypass access controls, steal data, craft phishing attacks and launch targeted attacks using browser exploits.

solution: Sanitize all user-supplied data before using it as part of dynamically generated pages and data.

request:

```
GET http://dmg.kiivi.org/ilmoitustaulu/etusivu/php/ilmoitus.php?id=%22'%3CoNHH1%3E HTTP/1.1
```

Banner Disclosure

The server or application disclosed its type and version. This information could be used by attackers to make an educated guess about the application environment and any inherited weaknesses that may come with it.

solution: It is recommended to prevent the application from disclosing its type and version.

banner: X-Powered-By: PHP/5.2.17, PleskLin**request:**

```
GET http://dmg.kiivi.org/ilmoitustaulu/etusivu/index.php HTTP/1.1
```

banner: X-Powered-By: PleskLin**request:**

```
GET http://dmg.kiivi.org/ilmoitustaulu/etusivu/this.href%3B HTTP/1.1
```

File Upload

File upload facilities are usually considered dangerous as if mishandled, they could lead to persistent Cross-site Scripting (XSS) attacks or remote code execution.

solution: Ensure that the file upload form has the appropriate access controls and is resilient to common attacks.

url: http://dmg.kiivi.org/ilmoitustaulu/etusivu/index.php**form:** <form method="POST" action="http://dmg.kiivi.org/ilmoitustaulu/etusivu/php/lisaa_ilmoitus.php" enctype="multipart/form-data" autocomplete="on"> ... </form>