

Opinnäytetyö (AMK)

Tietotekniikan koulutusohjelma

Ohjelmistotekniikka

2010

Jori Jokinen

TYÖKALU INTERNETSIVUJEN ARVIOINTIIN



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma | Ohjelmistotekniikka

Kesäkuu | 32

Ohjaaja: TkL Jari-Pekka Paalassalo

Jori Jokinen

TYÖKALU INTERNETSIVUJEN ARVIOINTIIN

Työn tavoite oli luoda internetsivuja tekeväille Sofokus Oy:lle uudenlainen sovellus, jolla kommentoidaan heidän tekemiään internetsivuja. Sovellus suunniteltiin parantamaan yrityksen ja asiakkaan välistä kommunikointia.

Työssä suunniteltiin ja toteutettiin tähän tarkoitukseen internetpohjainen sovellus. Internetsivuista otetaan kuvankaappaus, johon voidaan tämän jälkeen lisätä kommentteja haluttuihin kohtiin. Näin saadaan asiakkaalta konkreettista palautetta helpommin sekä ymmärrettään asiakkaan tarpeet paremmin.

Sovelluksessa on erityisesti otettu huomioon helppokäyttöisyys, jotta kaikki asiakkaat osaisivat käyttää sovellusta hyödykseen. Tästä syystä sovelluksesta tehtiin mahdollisimman yksinkertaisen näköinen ja keskityttiin saamaan sovelluksesta mahdollisimman helposti käytettävä.

Työn tuloksena saatiin toimiva sovellus, jota yritys käyttää hyödyksi internetsivujen kehitystyössään. Sovelluksessa on hyvät jatkokehitysmahdollisuudet ja sille varmasti löytyisi kaupallisia tarpeitakin, koska monet alan yritykset kärsivät samasta ongelmasta.

ASIASANAT:

verkko-ohjelmointi, JSP, JavaServlet, JavaScript

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Information Technology | Software Engineering

June | 32

Instructor: Jari-Pekka Paalassalo, Lic.Tech, Principal Lecturer.

Jori Jokinen

TOOL FOR EVALUATING WEB PAGES

The main objective of this thesis was to develop a web-based software application for commenting on web pages for Sofokus Ltd. The application was designed to enhance communication between company and customer.

For this purpose a web-based software application was developed. With this software it is possible to comment on web pages that the company has made. A user takes a screenshot in which comments can be attached to desired places. In this way the company receives concrete feedback more easily and understands the customer's needs better.

Ease of use was especially one of the main goals of this application because every customer should be able to use it. By courtesy of that, the application was made to look simple, also focusing on making it as easy to use as possible.

As a result of this thesis, a functional application that the company takes advantage of when producing web pages was developed. The application has great further development possibilities for being even commercialized because many companies of the same industry suffer from the same problem.

KEYWORDS:

web-programming, JSP, JavaServlet, JavaScript

SISÄLTÖ

1 JOHDANTO	1
2 TYÖN ALKUASETELMAT JA TOTEUTUSTEKNIIKAT	2
2.1 Lähtökohdat ja tavoitteet	2
2.2 Sovelluksen toiminta	2
2.3 Työn kartoitus	3
2.4 Työn suunnittelu	3
2.5 Toteutustekniikat	4
2.5.1 Ohjelmointikieli	4
2.5.2 Tietokanta	7
2.5.3 Sovelluspalvelin	7
2.6 Sovelluksessa käytetyt ohjelmointirajapinnat ja liitännäiset	8
2.6.1 Hibernate Framework	9
2.6.2 Apache Tiles	9
2.6.3 jQuery	9
2.6.4 Jcrop	10
2.6.5 Easy Slider	10
3 SOVELLUKSEN TOTEUTUS	10
3.1 Järjestelmän arkkitehtuuri	11
3.2 Kehitysympäristön asennus	11
3.3 Tietokannan luonti	12
3.4 Hibernate Frameworkin asennus	13
3.5 Apache Tilesin asennus	14
3.6 Lokalisoinnin asennus	15
3.7 Sovelluksen ohjelmointi	16
3.7.1 Sovelluksen runko	16
3.7.2 Perustoiminnallisuudet	17
3.7.3 Interaktiivisten toiminnallisuuksien toteutus	18
4 SOVELLUKSEN TOIMINTA JA HYÖTY KÄYTTÄJILLE	21
5 YHTEENVETO	24
LÄHTEET	25
LIITTEET	26

1 JOHDANTO

Internetsivuja kehittävien yritysten ongelmana on saada kehitettävistä sivuista hyvää ja konkreettista palautetta asiakkailta kehitystyön aikana. Puhelin- ja sähköpostikeskustelut ovat huonoja vaihtoehtoja tässä tapauksessa, sillä asiakkaasta ei aina saa näillä keinoin selvää mitä hän todella haluaa. Tapaamiset asiakkaan kanssa ovat erittäin hyvä vaihtoehto, mutta ovat taas aikaavieviä, joten niitä ei jokaisen pienen asian takia kannata järjestää.

Tähän tarkoitukseen kehitettiin Sofokus Oy:n toimeksiannosta internetpohjainen sovellus, jolla voidaan kommentoida yrityksen kehittämiä internetsivuja. Internetsivuista otetaan kuvankaappaus, johon voidaan liittää kommentteja haluttuihin kohtiin. Sovelluksen tarkoituksena on parantaa yrityksen ja asiakkaan välistä kommunikaatiota, jolloin kehitystyö nopeutuu ja helpottuu. Visuaalisuuden kautta ohjelmistokehittäjä pystyy huomattavasti nopeammin ja paremmin ymmärtämään asiakkaan tarpeet, joten etu on molemminpuolinen.

Päätavoitteena oli suunnitella sovelluksesta mahdollisimman helppokäyttöinen, jotta jokainen asiakas osaisi sitä käyttää. Ulkoasusta luotiin mahdollisimman yksinkertainen ja pelkistetty, jotta ohjelmasta saataisiin mahdollisimman selkeä. Työnkehityksessä käytettiin pohjana toimeksiantajan ja kehittäjän omia näkökulmia, joiden pohjalta valmistettiin toimiva sovellus.

2 TYÖN ALKUASETTELMAT JA TOTEUTUSTEKNIIKAT

2.1 Lähtökohdat ja tavoitteet

Tässä opinnäytetyössä kehitetään internetsivujen kommentointiohjelma. Idea lähti liikkeelle Sofokus Oy:n tarpeesta parantaa asiakkaan ja yrityksen välistä kommunikointia.

Kyseisellä yrityksellä, kuten varmasti muillakin internetsovelluksia kehittäville yrityksillä, on ongelmana saada riittävän hyvää konkreettista palautetta asiakkaalta helposti. Visuaalisen palautteen avulla yrityksen työntekijä pystyy huomattavasti helpommin ja nopeammin saamaan selville mitä asiakas todella haluaa.

Hyvä tapa tähän asti on ollut järjestää asiakkaan kanssa tapaaminen, jossa käsitellään kaikki asiat niin, että molemmat ovat kaikista asioista selvillä. Kuitenkin sovellusta kehitettäessä kaikista asioista ei ole kannattavaa järjestää tapaamista, vaan asiat olisi hyvä hoitaa jollakin muulla tavalla.

Tähän tarkoitukseen kehitettiin sovellus, jolla pystyy kommentoimaan internetsivuja. Kyseinen sovellus helpottaa huomattavasti yrityksen ja asiakkaan välistä kommunikaatiota, koska sovelluksen avulla asiakkaalta saadaan konkreettista visuaalista palautetta internetsivuista. Tavoitteena on näin helpottaa ja nopeuttaa sovellusten kehitystä, koska asiakkaiden kanssa kommunikointi helpottuu uuden sovelluksen myötä.

Tarkoituksena on rakentaa toimeksiantajan ideoista toimiva sovellus, joka asennetaan yrityksen omalla palvelimelle toimintakuntoon. Sovellus otetaan ensin pelkästään yrityksen omaan käyttöön, josta mahdollisuuksien mukaan sovellusta voidaan vielä jatkokehittää ja parannella mahdollisesti ilmitulleiden asioiden pohjalta.

2.2 Sovelluksen toiminta

Sovelluksen pääasiallinen toiminto on itse internetsivun kommentointi, mutta kuitenkin sovellus vaatii huomattavan määrän muutakin, jotta siitä saadaan toimiva. Käyttäjän tulee pystyä luomaan ensin uusi projekti, johon tämän jälkeen voidaan ladata kuvia. Kuvat on jaettu projekteihin, jotta käytettävyys on helpompaa asiakkailta, joilla on

paljon kuvia. Sovelluksen avulla ladataan kuvankaappaus halutusta sivusta ohjelmaan. Tämän jälkeen voidaan siirtyä itse kuvan kommentointiin.

Asiakas tai yritys voi ladata sovellukseen kuvia, joita molemmat voivat käydä katsomassa ja lisäämässä haluamalleen alueelle omia kommentteja ja muokata myös toisen tekemiä valmiita kommentteja. Näin yrityksen ja asiakkaan välinen kommunikointi paranee huomattavasti, kun kommunikoinnissa on apuna visuaalisuutta.

2.3 Työn kartoitus

Projektin suunnittelua lähdettiin tekemään kartoittamalla ensin, mitä kaikkea ominaisuuksia ohjelmassa tulisi olla. Kartoitus toteutettiin keskustelemalla yrityksen yhteyshenkilön kanssa. Projektin kartoituksesta pyrittiin tekemään mahdollisimman tarkka, jotta vaatimukset olisivat mahdollisimman hyvin selvillä, ennen kuin projektia lähdetään toteuttamaan.

Kartoituksessa määriteltiin mahdollisimman tarkasti kaikki ohjelmaan halutut ominaisuudet ja muut erityisvaatimukset. Tämän hyvin tehdyn kartoituksen pohjalta on hyvä lähteä tekemään itse suunnittelua. Kartoituksesta tehtiin kunnollinen dokumentointi, jotta kaikki asiat olisivat tallessa, eikä mitään unohtuisi projektia tehdessä.

Usein projektit saattavat laajeta edetessään, joten kartoituksella pyritään pitämään huolta myös siitä, että projekti pysyy annetuissa rajoissa. Ilman kunnollista kartoitusta työnantaja saattaa vaatia aina vain lisää ominaisuuksia sovellukseen sitä mukaa, kun niitä tulee mieleen ohjelmaa tehtäessä.

Vaikka kartoitus tehtäisiin kuinka tarkasti, usein molempien osapuolten on joustettava jonkin verran, jotta sovelluksesta saataisiin kehitettyä mahdollisimman hyvä. Tässäkään tapauksessa projektin edetessä ei voitu välttää projektin paisumista, sillä sovelluksen kehitysideoita tuli melko paljon. Kuitenkaan mitään kovin suuria muutoksia ei tullut ja hyvän kartoituksen ansioista projekti saatiin pidettyä hyvin koossa.

2.4 Työn suunnittelu

Jo heti projektin kartoituksen alussa tarkennettiin, että pääpaino sovellusta kehitettäessä tulee olla käytettävyydessä. Täten mitään näyttäviä grafiikoita tai animaatioita ei tarvitse tehdä. Tästä syystä projektin suunnittelussa keskityttiin

erityisesti saamaan sovelluksesta mahdollisimman yksinkertainen ja helppokäyttöinen. Sovelluksen helppo ja sujuva käytettävyys on erittäin tärkeää, koska sovellusta tullaan käyttämään yrityksen asiakkaiden toimesta ja jokaisen asiakkaan tulisi osata käyttää sitä. Lisäksi sovelluksen tulisi antaa hyvä kuva yrityksestä asiakkaille, jotta asiakkailla olisi mahdollisimman positiivinen mielikuva yrityksestä.

Suunnittelussa käytiin läpi kaikki sivut ja toiminnallisuudet, jotka sovellukseen tulee. Suunnittelu pyrittiin tekemään mahdollisimman tarkasti jokaisen komponentin osalta, jotta sovelluksen toteutusvaiheessa ei tulisi enää ongelmia, jotka ovat johtuneet puutteellisesta suunnittelusta.

Yrityksen yhteyshenkilö antoi erittäin vapaat kädet sovelluksen suunnittelussa. Tarkkaa kuvausta siitä, millaiset sivut tarkalleen tuli olla, ei annettu. Sivujen hierarkiastakaan ei annettu mitään ohjetta, vaan se tuli myös itse suunnitella vastaamaan sovelluksen tarpeita. Toteutustekniikoissa suositeltiin käyttämään samaa, kuin yritys käyttää, jotta mahdollisissa ongelmatilanteissa apua saisi työtovereilta. Kuitenkin suunnittelun aikana kartoitettiin kaikki vaihtoehdot, jotta löytyisi paras toteutustekniikka juuri tälle työlle.

2.5 Toteutustekniikat

Projektin kartoitusvaiheessa yrityksen toimesta määrättiin, että ohjelmasta tulee internetpohjainen. Tämä päätös tehtiin pitkälti sen takia, että ohjelmaa olisi helpompi päästä käyttämään, koska sitä ei tarvitse asentaa koneelle. Asiakkailla on huomattavasti helpompaa, kun ei tarvitse asentaa uutta ohjelmaa kyseistä palvelua varten. Internetpohjainen sovellus on myös muuten hyvä kyseiselle yritykselle, koska he itsekin valmistavat internetsivuja, joten sivujen ylläpito ja kehitys on helppoa. Toimiva sovellus antaa myös asiakkaalle positiivisen mielikuvan ja luottamusta kyseistä yritystä kohtaan.

Toteutustekniikat ovat niin ikään erittäin tärkeää suunnitella, ennen kuin aloittaa itse projektin toteuttamisen. Hyvällä suunnittelulla varmistetaan, että sovellus on ylipäätään mahdollista tehdä käytetyllä toteutustekniikalla ja että sovelluksen toteutuksessa käytettäisiin mahdollisimman hyvin siihen sopivia tekniikoita.

2.5.1 Ohjelmointikieli

Ensimmäisenä vaiheena toteutustekniikoiden suunnittelussa oli pohtia, mitä ohjelmointikieltä projektissa käytettäisiin. Esikartoituksen jälkeen vaihtoehtoja

käytettäväksi ohjelmointikieleksi valikoitui kolme, jotka eroavat melko paljon toisistaan: Java, Microsoft.NET ja PHP. Suunnittelussa perehdyttiin jokaisen kielen osalta siihen, mikä soveltuisi parhaiten juuri kyseisen sovelluksen toteutukseen. Suunnittelu on tärkeää tehdä tässä vaiheessa hyvin, sillä se helpottaa erittäin paljon toteutusta ja näin varmistetaan, että ei tule isoja kompastuskiviä projektin toteutusvaiheessa.

Java

Java on Sun Microsystemsin kehittämä ohjelmointikieli. Javan vahvuutena on sen turvallisuus, koska sitä ajetaan virtuaalikoneessa. Javalle löytyy myös valmiiksi paljon erilaisia valmiita ohjelmaosia, joita voi käyttää hyödyksi ohjelmassa. Kielen heikkous on sen raskaus. Java vaatii virtuaalikoneensa takia enemmän resursseja muihin kieliin verrattuna.[1]

Javalla tehtävissä web-pohjaisissa sovelluksissa käytetään tekniikkaa: Java Servlet + JavaServer Pages(JSP). Näillä yhdessä pystytään luomaan dynaamisia internetsivuja.[2] Tässä sovelluksessa lisäksi käytetään JavaScriptiä, jonka avulla pystytään tekemään itse kuvan kommentointisivulla tarvittavat lisätoiminnallisuudet.

Java Servlet Technology on rajapinta, jonka on kehittänyt Sun Microsystems. Rajapinnan avulla voidaan luoda www-ohjelmia Javalla. Servleteillä pystytään tuottamaan dynaamisia vastauksia lähetettyjen pyyntöjen perusteella. Servletien avulla sovellus pystyy hakemaan esimerkiksi JDBC-rajapintaa hyväksikäyttäen tietokannasta haluttavia tietoja riippuen pyynnöstä.[3]

JavaServer Pagesin avulla pystytään luomaan dynaamisia internetsivuja yksinkertaisemmin ja helpommin. JSP-sivulla voidaan käyttää hyödyksi JavaServer Pages Standard Tag Libarya (JSTL) ja Expression Languagea (EL), joilla dynaamisen sisällön käsittely sivulla on erittäin helppoa.[4]

Yhdessä näiden kaikkien avulla saadaan helposti luotua lopullinen dynaaminen internetsivu. Kyseinen tekniikka on erittäin suosittu internetsivujen kehittäjien keskuudessa sen nopeuden ja helppouden ansiosta.[2]

Microsoft.NET

Microsoftin kehittämä ohjelmistokomponenttikirjasto .NET Framework mahdollistaa ohjelmien turvallisen ja nopean kehittämisen. .NET tukee n. 20 ohjelmointikieltä, näistä käytetyimpiä ovat C# ja VB.NET. Suurimpana heikkoutena voidaan pitää työkalujen hintaa.[5]

PHP

PHP on erittäin paljon käytetty kieli dynaamisissa internetsivuissa. PHP on komentosarjakieli, joka käännetään ymmärrettävään muotoon vasta internetsivua kutsuttaessa. PHP:n suurimpia etuja ovat sen keveys sekä MySql-tietokantojen käytettävyys. PHP-koodi sulautetaan HTML:n joukkoon.[6]

Valittu ohjelmointikieli

Taustatutkimusta tehtäessä valinta alkoi kallistua Javan suuntaan. Javalle on olemassa paljon valmiita toiminnallisuuksia, joita löytyi myös käytettäväksi tässä ohjelmassa. Nämä helpottavat huomattavasti ohjelman tekoa, koska aivan kaikkea ei tarvitse itse tehdä.

Erittäin suuri vaikutus kielen valintaan oli myös se että toimeksiantajayritys käyttää itse pääasiassa Javaa ohjelmointikielenä. Ongelmien tullessa saisi helpommin apua, kun on kokeneita ohjelmistokehittäjiä tukena.

Kuten tässä luvussa aiemmin jo mainittiin, niin Javalla tehtäessä internetsivuja käytetään hyödyksi Java Servleteja ja JavaServer Pageseja. Näiden avulla pystytään helposti ja nopeasti luomaan dynaamisia internetsivuja.[2] Tätä tekniikkaa käytettiin myös tässä projektissa.

Kuvan kommentointisivu vaatii kuitenkin erittäin monimutkaisia ja interaktiivisia toimintoja, joita ei yksinään näillä tekniikoilla pysty toteuttamaan. Tähän asiaan avuksi otettiin JavaScript. JavaScript mahdollistaa näiden kaikkien interaktiivisten toimintojen teon. JavaScriptin avulla voidaan muun muassa liittää dynaamista tekstiä sivuille, ottaa vastaan käyttäjän komentoja, sekä lukea ja kirjoittaa Html:a.[7]

2.5.2 Tietokanta

Tehtävä sovellus vaatii myös toimiakseen tietokannan. Käyttäjien lisäämät tiedot säilytetään tietokannassa, josta ohjelma voi käyttää tietoja hyödykseen haluamallaan tavalla. Tietokannan valinta oli myös melko suoraviivainen. Jo projektin kartoitusvaiheessa suositeltiin käyttämään tietokantana MySQL:ää. Suositus oli pitkälti sitä varten, että yritys itse käyttää tietokantana MySQL:ää palvelimella. Täten palvelimelle on valmiiksi MySQL asennettuna. Muitakin vaihtoehtoja tosin pohdittiin, mutta kuitenkin tultiin tulokseen, että käytettävä tietokanta on MySQL. Yksi suurimmista vaikuttavista asioista oli myös MySQL:n ilmaisuus, koska sovellus tuli tehdä käyttäen vapaan lähdekoodin tuotteita.

MySQL on maailman suosituin avoimen lähdekoodin tietokanta. MySQL:ää on ladattu yli 100 miljoonaa kappaletta sen historian aikana. MySQL:n suosion takana on sen nopeus, luotettavuus, helppokäyttöisyys ja tietenkin ilmaisuus. MySQL:n hyvydestä kertoo jotain myös, että sitä käyttävät monet suuret yritykset kuten esimerkiksi Google, YouTube ja suomalainen Nokia.[8]

Tietokannan ja sovelluksen välille päätettiin asentaa myös Hibernate Framework, jolla tietokannan ja sovelluksen välinen kommunikointi sujuu huomattavasti helpommin ja paremmin.

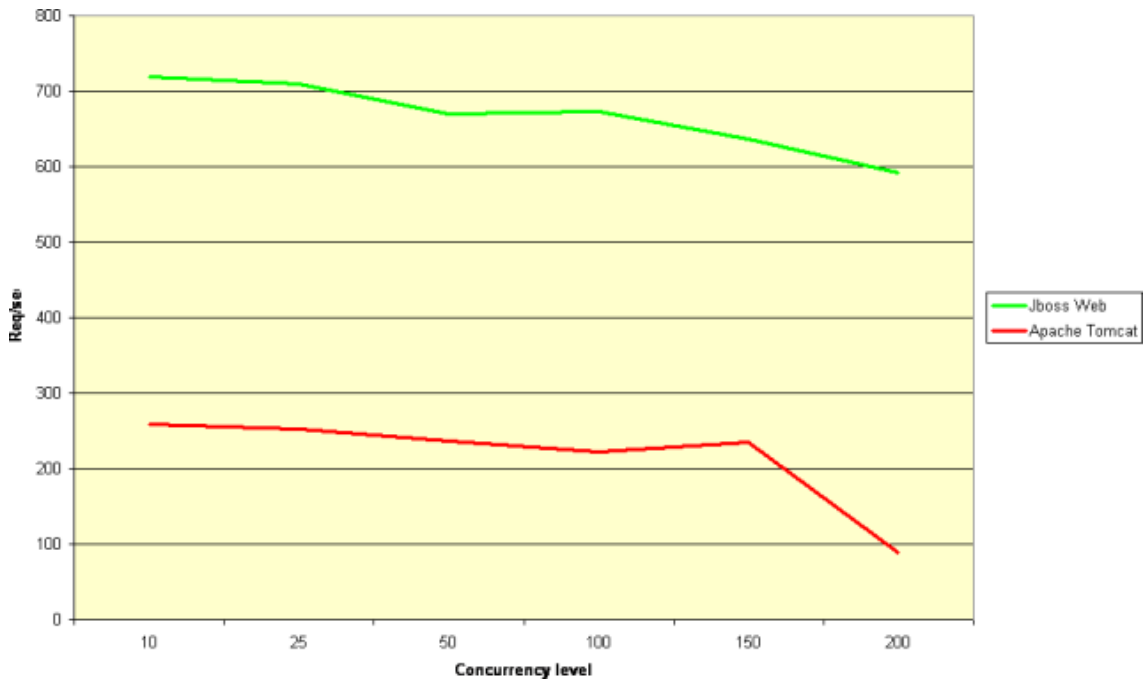
2.5.3 Sovelluspalvelin

Java vaati myös erillisen sovelluspalvelimen, jolla itse sovellus suoritetaan erillisessä virtuaalikoneessa. Vaihtoehtoja sovelluspalvelimeksi oli muutamia, joista piti valita projektiin parhaiten sopiva.

Sovelluspalvelimeksi valittiin JBoss 4.0.2, koska yritys käytti tätä jo valmiiksi omalla palvelimellaan. Ohjelma on näin ollen helppo julkaista palvelimelle ja se toimii varmasti niin kuin on suunniteltu. Sovelluksen tekovaiheessa JBossista on saatavilla jo uudempikin versio, mutta sovelluksen kehityksessä päätettiin kuitenkin käyttää vanhaa versiota varmuudenvuoks, jotta säästyttyäisiin mahdollisilta yhteensopivuusongelmilta. Uusi JBossin versio kun ei kuitenkaan ollut tuonut mitään uusia asioita, joista projektia tehdessä olisi ollut hyötyä.

JBoss on ilmainen open source lisenssillä saatava tuote. Se rakentuu Apache Tomcat web-palvelimen varaan ja tuo tähän vielä päälle monia asioita, jotka helpottavat

ohjelman tekoa. JBossin suuri valttikortti on myös sen nopeus verrattuna Apache Tomcatiin, kuten kuvassa 1 voidaan huomata.[9] JBoss on lisäksi erittäin helppo asentaa, lataat vain tiedostot koneelle ja olet valmis kehittämään sovelluksia sen päälle.



Kuva 1. JBossin ja Tomcatin nopeusvertailu.

2.6 Sovelluksessa käytetyt ohjelmointirajapinnat ja liitännäiset

Ohjelmointirajapinta on valmis ohjelmistokomponentti, jota voidaan käyttää hyödyksi sovelluksen osana ja näin ollen helpottaa omaa ohjelmointityötään. Ohjelmointirajapinnat ovat erittäin käteviä ja aikaa säästäviä, kun kaikkea ei tarvitse kehittää alusta asti uudelleen, vaan voi käyttää hyödyksi jonkun aikaisemmin tekemää ohjelmointirajapintaa.[2]

Tämän projektin suunnitteluvaiheessa etsittiin ja tutustuttiin mahdollisiin ohjelmointirajapintoihin, joita voitaisiin käyttää hyväksi tässä projektissa. Taustatutkimus tuotti tulosta ja erittäin hyviä rajapintoja löytyi muutama. Seuraavassa esitellään muutama tässä sovelluksessa käytetty rajapinta ja liitännäinen.

2.6.1 Hibernate Framework

Hibernate Framework on ilmainen tietokannan ja ohjelman välillä käytettävä sovelluskehys, joka helpottaa erittäin paljon tietokannan käyttöä ohjelmassa. Hibernaten suurin hyöty on tietokannan toiminnan peittäminen olio-suuntautuneelta maailmalta. Tietokannan taulut ja suhteet kuvataan xml-tiedostoilla. Hibernaten avulla tietojen hakeminen ja lisääminen tietokantaan on huomattavasti helpompaa ja nopeampaa. Kyselyt tehdään käyttämällä kuitenkin sql-kieltä, joten monimutkaisemmatkin kyselyt onnistuvat myös Hibernaten avulla. [3]

2.6.2 Apache Tiles

Internetsivuja kehitettäessä unohtuu usein ohjelmistokehittäjiltä ohjelmistokoodin uudelleenkäyttö. Samaa asiaa ei pitäisi tehdä monessa kohdassa, vaan tehdä yksi koodi, jota voi käyttää jokaisessa paikassa. Tähän tarkoitukseen on kehitetty Apache Tiles, jolla voidaan sovelluksen ulkoasu tehdä vain kerran ja käyttää valmiita komponentteja jokaisella sivulla tarpeiden mukaan. Tiles on erinomainen työkalu tehtäessä www-sovelluksia. Se helpottaa paljon käyttöliittymän kehittämistä. Tilesin avulla saadaan aikaan yhtenäinen ulkoasu koko sovellukselle. Tilesillä tehdyistä ”tiilistä” kootaan kokonainen sivu. Tämä helpottaa erittäin paljon uusien sivujen tekemistä yhtenevällä ulkoasulla. Sovelluksen kehittäminen helpottuu huomattavasti, kun on olemassa valmis sivupohja, johon vain laitetaan haluttu tieto sisälle.[2]

2.6.3 jQuery

Projektin aikana tehtävä sovellus poikkeaa hyvin paljon perinteisestä internetisivusta. Tämän sovelluksen pääsivulla on erittäin paljon interaktiivisia toiminnallisuuksia. Näiden toiminnallisuuksien tekoon valittiin jQuery, jolla kyseisten asioiden teko on huomattavasti helpompaa ja nopeampaa, kuin käyttämällä pelkästään JavaScriptiä.

jQuery on uudenlainen JavaScript-kirjasto, joka helpottaa erittäin paljon JavaScript-ohjelmointia. jQuery hoitaa asioita puolestasi, joten ohjelmointi on huomattavasti nopeampaa ja helpompaa sen avulla. jQueryn suosion taustalla on sen sopiminen erittäin monen asian tekoon. jQueryn avulla voidaan muun muassa muokata sivun ulkoasua, animoida dokumentin muutoksia ja hakea AJAXilla tietoa serveriltä ilman sivun uudelleenlatausta.[10] Tässä lueteltuna vain pääasiat, joita tässäkin sovelluksessa on käytetty hyödyksi.

jQueryn tullessa erittäin suosituksi internet-sovellusten kehittäjien keskuudessa, monet ovat kehittäneet siihen myös erinomaisia liitännäisiä. Liitännäisten avulla kaikkea ohjelmointityötä ei tarvitse tehdä itse, vaan voidaan käyttää hyväksi näitä liitännäisiä. Projektin suunnittelun aikana etsittiin jo mahdollisia liitännäisiä, joita voisi käyttää hyödyksi sovelluksessa. Kyseisiä liitännäisiä löytyikin muutama ja näitä pyrittiin käyttämään mahdollisimman hyvin hyödyksi sovelluksessa.

2.6.4 Jcrop

Jcrop on jQueryn liitännäinen, jonka avulla voidaan kuvasta valita jokin osa. Se on alun perin suunniteltu alueen valitsemiseen kuvasta ja tekemään tästä uusi kuva. [11] Tässä sovelluksessa ei kuitenkaan tarvittu kaikkea toiminnallisuutta mitä Jcrop tarjoaa. Jcropilla havainnollistetaan kuvasta alue, jolla liitetty kommentti on. Liitännäistä jouduttiin kuitenkin muokkaamaan melko paljon, jotta se soveltui mahdollisimman hyvin kyseiseen tarkoitukseen. Jcrop on helposti muokattavissa soveltumaan myös erittäin moneen muuhunkin asiaa.

2.6.5 Easy Slider

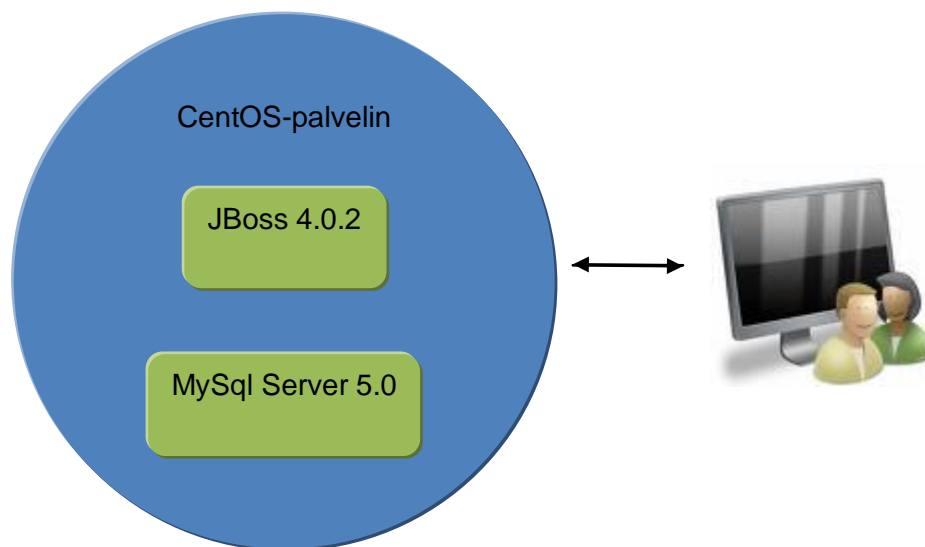
Easy Slider on niin ikään jQueryn liitännäinen. Easy Slider on suunniteltu käytettäväksi liu'uttamaan mitä tahansa elementtejä internetsivuilla, jonka johdosta se onkin erittäin monikäyttöinen liitännäinen.[12] Erittäin monilla internetsivuilla on toteutettu muun muassa animoitu mainostenvaihto kyseisellä liitännäisellä.

3 SOVELLUKSEN TOTEUTUS

Projektin huolellisen suunnittelun jälkeen on hyvä lähteä tekemään itse sovellusta. Huolellinen suunnittelu auttaa välttämään suurimmilta kompastuskiviltä. Projektin toteutuksessa lähdetään liikkeelle kehitysympäristön asentamisesta, jonka jälkeen aloitetaan sovelluksen tekeminen. Projektin aikaakuluttavin osa on itse sovelluksen ohjelmointi, johon menee huomattavan paljon aikaa. Kun sovellus on saatu valmiiksi, asennetaan tämä yrityksen palvelimelle testaukseen. Testauksen jälkeen suoritetaan mahdolliset virheenkorojaukset, jos niitä ilmenee.

3.1 Järjestelmän arkkitehtuuri

Valmis sovellus julkaistaan yrityksen omalla palvelimella, jossa se otetaan yrityksen omaan testauskäyttöön. Kyseisen palvelimen käyttöjärjestelmänä toimii Linuxin CentOS. Erillistä tietokantapalvelinta ei ole, vaan kyseiselle palvelimelle on asennettu sekä JBoss 4.0.2, että MySql Server 5. Kuva 2 havainnollistaa käytettyä arkkitehtuuria.



Kuva 2. Palvelimen arkkitehtuuri

3.2 Kehitysympäristön asennus

Kehitysympäristö on rakennettu Microsoft Windows 7:ää käyttävälle koneelle. Ympäristön asennus aloitettiin asentamalla Eclipse. Se on ilmainen ohjelmointiympäristö, joka soveltuu erittäin hyvin Java-sovellusten kehitykseen. Eclipsen asennus on erittäin helppoa, lataat vain tiedostot koneelle ja olet valmis käyttämään ohjelmaa.

Internet-pohjaisten ohjelmien kehitys Javalla vaatii myös sovelluspalvelimen, johon ohjelma julkaistaan ohjelman käyttöä varten. Suunnitteluvaiheessa päädyttiin käyttämään JBossin versiota 4.0.2. JBossin asennus on erittäin helppoa. Ladataan tiedostot vain koneelle ja olet valmis käyttämään palvelinta lisäämällä sen Eclipsessä julkaisupalvelimeksi.

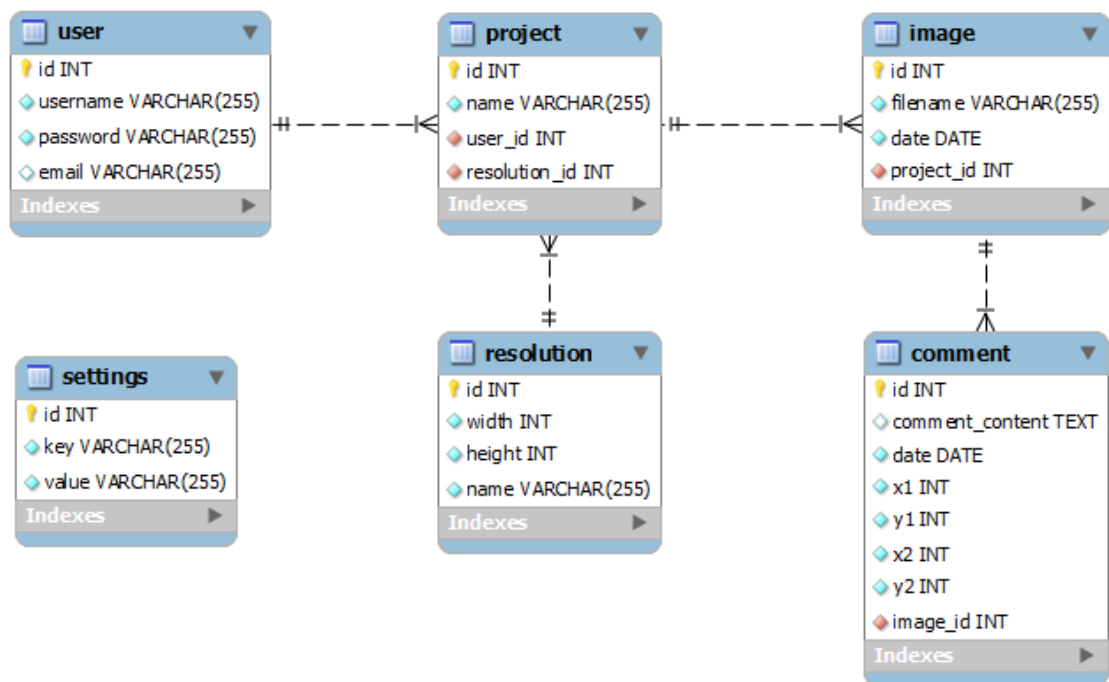
Viimeisenä vaiheena kehitysympäristön asennuksessa on tietokannan asennus. Tietokanta tarvitsee tietokanta serverin, johon itse tietokanta laitetaan. Koska tietokantana käytetään MySql-kantaa, niin tietokoneelle asennettiin MySql Server 5.1. Asennus onnistuu helposti seuraamalla asennusvelhoa.

Kun kehitysympäristö on saatu asennettua koneelle kokonaan, on seuraavaksi vuorossa luonnollisesti itse sovelluksen toteutus. Tässä lähdettiin liikkeelle suunnittelemalla ja toteuttamalla tietokanta alkuperäisen projektisuunnitelman pohjalta.

3.3 Tietokannan luonti

Tietokannan luomisessa lähdettiin liikkeelle suunnittelemalla tietokannan taulut ja rivit projektisuunnitelman pohjalta. Tietokannan suunnitteleminen kannattaa tehdä huolellisesti, jotta välttyttäisiin mahdollisilta muokkauksilta jälkeenpäin. Näiltä ei kuitenkaan voi aina välttyä, kun jokin ohjelman osa vaatiikin jotain erilaista, kuin on osannut etukäteen suunnitella. Tässäkään projektissa muokkaamiselta ei välttytty huolellisesta suunnittelemisesta huolimatta. Yksi projektin osa muuttui projektin loppuvaiheella erilaiseksi, koska tuli parempi toteutusidea vastaan. Tässä vaiheessa myös tietokantaa jouduttiin muokkaamaan vastaamaan uusia tarpeita. Onneksi tietokannan muokkaaminen on erittäin yksinkertaista hyvillä ohjelmilla tehtynä.

Tietokannan luonnissa käytettiin ilmaista MySql Workbenchiä, jolla on erittäin nopeaa ja helppoa luoda tietokantoja. Ohjelmalla luotiin kaikki halutut taulut, rivit ja suhteet taulujen välille. Tämän jälkeen tietokanta laitettiin serverille kyseisellä ohjelmalla, jonka jälkeen tietokanta on käytettävissä.



Kuva 3. Tietokannan malli

Tietokantamallissa (Kuva 3) nähdään, millaisen tietokannan tehty sovellus vaatii. Tietokannan osalta sovellus on erittäin yksinkertainen. Kaikki tarvittava tieto saadaan laitettua kuuteen tauluun.

3.4 Hibernate Frameworkin asennus

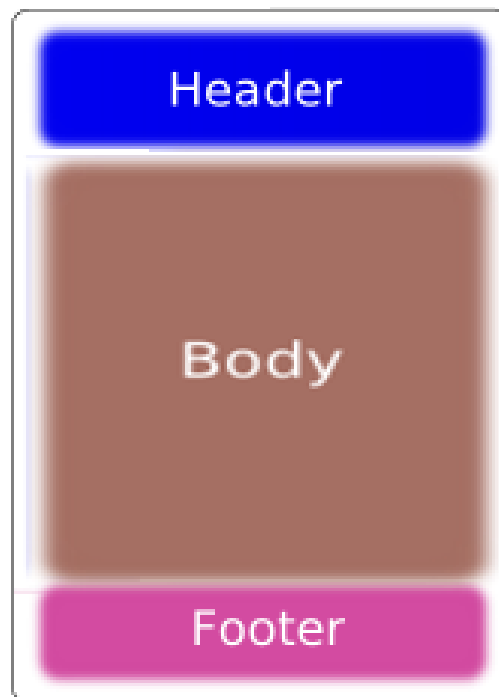
Hibernaten asentaminen projektiin vaatii jonkin verran perehtymistä asiaan, mutta lopulta kaikki on sen vaivan arvoista. Hibernate vaatii muutaman jar-tiedoston lisäämisen projektiin. Tämän lisäksi tietokannan tauluista täytyy tehdä vastaavat xml-tiedostot, joilla kuvataan tietokanta Hibernatelle ja annetaan tarvittavat asetukset tietokannan käyttöä varten. Lisäksi tarvitsee tehdä luokat, joiden kautta tiedot haetaan Hibernatella. Tämän jälkeen Hibernaten käyttö on helppoa ja vaivatonta. [3]

Hibernaten asennuksessa kannattaa perehtyä tarkasti ohjeisiin jo ennen kuin aloittaa asennusta, jotta välttyä mahdollisilta ongelmilta. Hibernaten asennus on melko hankala, mikäli sitä ei ole koskaan ennen tehnyt. Hibernate vaatii näin ollen paljon tutustumista asiaan ennen kuin asennus onnistuu. Liitteet 1...3 ovat esimerkki-tiedostot Hibernaten vaatimista tiedostoista. Esimerkkinä käytetty tietokannan user-taulua.

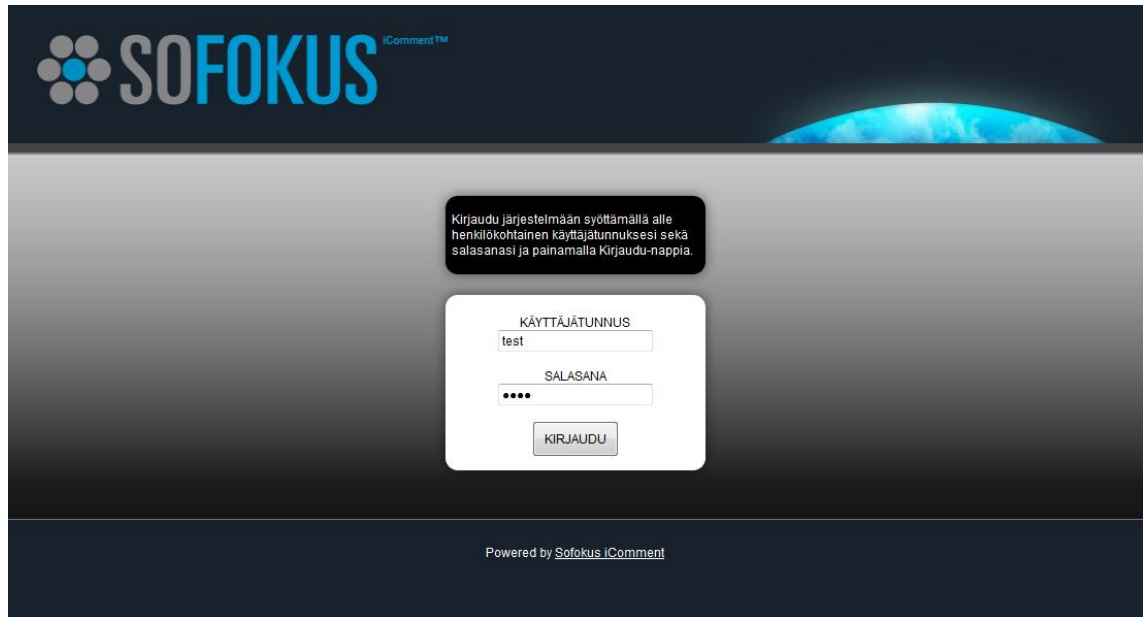
3.5 Apache Tilesin asennus

Yksi projektin vaatimuksista oli saada sovellus vastaamaan yrityksen yleistä ulkoasua. Tämä päätettiin toteuttaa valitsemalla sivupohjaksi yrityksen aiemmissa sovelluksissa käyttämä sivupohja. Aiemman sovelluksen pohjaa voitiin joltain osin käyttää hyödyksi, mutta vaati kuitenkin paljon muokkausta ennen kuin pohja saatiin vastaamaan tämän projektin tarpeita. Vanhoista grafiikoista ja muista oli kuitenkin erittäin paljon hyötyä, kun niitä ei joutunut tekemään kaikkia uudelleen.

Tässä sovelluksessa käytetty ulkoasu on käytettävyyden takia erittäin yksinkertaistettu. Sivun on jaettu kolmeen osaan: otsikkopalkki, sivun sisältö ja alaviite. Otsikkopalkki sekä alaviite ovat kaikilla sivuilla samanlaiset, kun taas sisältöön tulee aina kunkin sivun sisältämä informaatio. Kuva 4 esittää havainnollisemmin sivun Tiles-komponentit.



Kuva 4. Sovelluksessa käytetty sivupohja



Kuva 5. Sisäänkirjautumissivun ulkoasu

Sovellusta tehtäessä päätavoitteena oli saada siitä mahdollisimman helposti käytettävä. Tämän takia ulkoasusta pyrittiin tekemään mahdollisimman yksinkertaisen näköinen, jotta sovellusta olisi mahdollisimman helppo käyttää ja kaikki toiminnallisuudet löytyisivät hyvin. Kuten sovelluksen sisäänkirjautumissivusta (Kuva 5) huomaa, ulkoasu on melko pelkistetty, mutta kuitenkin näyttävännäköinen.

Tilesin asennus on melko yksinkertaista. Projektiin tulee lisätä muutama jar-tiedosto, jonka jälkeen voidaan käyttää Tilesia sivujen kehittämisessä. Tilesin käyttö ensimmäistä kertaa vaatii kuitenkin jonkin verran perehtymistä asiaan, jotta sovelluskehitys onnistuu sujuvasti.

3.6 Lokalisoinnin asennus

Sovelluksen yhtenä vaatimuksena oli käyttää sovelluksessa lokalisointia. Tämä tarkoittaa sitä, että kaikki tekstit jotka sivulla esitetään, ovat erillisessä tiedostossa. Kullekin kielelle on omat tiedostot, josta sovellus poimii tekstit valitun kielen mukaan. Tässä sovelluksessa ei kuitenkaan otettu käyttöön kuin suomenkieli, mutta lokalisointi tuli asentaa mahdollista tulevaisuuden tarvetta silmällä pitäen.

Lokalisoinnin asennus sovellukseen on erittäin yksinkertaista ja nopeaa. Myös tekstien käyttö sivuilla on tehty erittäin helpoksi, joten lokalisointi kannattaa lisätä sovellukseen, vaikka sitä ei välttämättä tarvitsisikaan sillä hetkellä.

3.7 Sovelluksen ohjelmointi

Kun kaikki oli saatu asennettu, aloitettiin itse ohjelmointityö. Tämä osio oli kaikista aikaa vievintä työtä, koska sovelluksen interaktiiviset toiminnallisuudet ovat erittäin monimutkaisia ja vaativat erittäin paljon ohjelmointityötä. Etenkin itse kommentointi-sivun ohjelmointi oli erittäin haasteellista.

3.7.1 Sovelluksen runko

Ohjelmoinnissa lähdettiin liikkeelle toteuttamalla sovelluksen runko. Sovelluksen runkona toimii 4 sivua: kirjautumis-, projekti-, esikatselu- ja kommentointi-sivu. Runkoa luodessa tehtiin erillinen ControllerAbstract-luokka (Ohjelma 1), joka tarkistaa aina session voimassaolon yritettäessä siirtyä johonkin sovelluksen sivulle. Mikäli voimassa olevaa sessiota ei ole, sovellus ohjaa käyttäjän sisäänkirjautumissivulle.

Ohjelma 1. ControllerAbstract-luokan lähdekoodi – ControllerAbstract.java

```
package com.sofokus.imagecomment.controller;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.sofokus.imagecomment.beans.SessionBean;

/*
 * Tarkistaa onko voimassa olevaa sessiota.
 *
 * @author Jori Jokinen
 */

public abstract class ControllerAbstract extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Tarkista onko voimassaolevaa sessiota
        SessionBean sessionBean = initSession(request);
        // Jos sessiota ei ole, ohjaa sisäänkirjautumissivulle
        if(sessionBean == null){
            response.sendRedirect(request.getContextPath() +
"/login/");
        }else{
            doGet(request, response, sessionBean);
        }
    }
}
```

```

public void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    // Tarkista onko voimassaolevaa sessiota
    SessionBean sessionBean = initSession(request);
    // Jos sessiota ei ole, ohjaa sisäänkirjautumissivulle
    if(sessionBean == null){
        response.sendRedirect(request.getContextPath() +
"/login/");
    }else{
        doPost(request, response, sessionBean);
    }
}

public abstract void doPost(HttpServletRequest request,
HttpServletResponse response, SessionBean session) throws
ServletException, IOException ;

public abstract void doGet(HttpServletRequest request,
HttpServletResponse response, SessionBean session) throws
ServletException, IOException ;

private SessionBean initSession(HttpServletRequest request){
    SessionBean sessionBean =
(SessionBean)request.getSession().getAttribute("sessionBean
");

    if(sessionBean != null){
        request.setAttribute("baseUrl",
request.getContextPath());
        return sessionBean;
    }else{
        return null;
    }
}
}
}

```

3.7.2 Perustoiminnallisuudet

Sovelluksen toteutuksessa tuli edetä järjestyksessä sivu kerrallaan eteenpäin, jotta pystyttiin testaamaan aina jokaista sivua. Tämän takia ohjelmoitiin seuraavaksi sisäänkirjautumissivun toiminnallisuudet.

Ohjelmointi aloitettiin toteuttamalla sisäänkirjautumissivulle kirjautumislaatikko JSP:llä. Tämän jälkeen tuli luoda Servlet, joka tarkistaa tietokannasta, onko kirjautumisessa annetut käyttäjätunnus ja salasana oikein. Salasana on tietoturvasyistä koodattu MD5:llä, jotta salasanaa ei pysty kukaan muu saamaan selville lähetettäessä se palvelimelle.

Jokaisen sivun pääperiaatteena toimii Servletin ja JSP:n yhdistäminen. Sivulle siirryttäessä Servletillä haetaan aina haluttu käyttäjäkohtainen informaatio

tietokannasta. JSP:n avulla nämä haetut tiedot lisätään sivulle halutulla tavalla. Näin saadaan aikaan dynaaminen sivu, jossa tiedot vaihtelevat aina käyttäjän mukaan. Liitteissä 4...5 havainnollistetaan prosessia esimerkkinä projekti-sivun Servlet- ja JSP-tiedostot.

3.7.3 Interaktiivisten toiminnallisuuksien toteutus

Suurin osa ajasta sovellusta toteuttaessa meni interaktiivisten toiminnallisuuksien ohjelmointiin, koska etenkin kommentointi-sivun toiminnallisuudet ovat erittäin monimutkaisia. Nämä toiminnallisuudet ohjelmoitiin käyttäen JavaScriptiä. JavaScriptillä ohjelmoitaessa käytettiin vielä hyödyksi jQuery-kirjastoa. jQueryn ansiosta ohjelmointityö oli helpompaa, kuin jos olisi joutunut tekemään pelkällä JavaScriptillä.

JavaScriptiä käytettiin hyödyksi myös muilla sivuilla joidenkin asioiden toteutukseen. Tästä esimerkkinä esikatselu-sivun dynaaminen kuvien näyttö. Esikatselu-sivulla voidaan muuttaa pudotusvalikosta, kuinka monta kuvaa näytöllä näytetään. Vaihdettaessa asetuksia, sovellus hakee AJAXin avulla halutut kuvat tietokannasta.

Koska JavaScriptiä tuli sovellukseen paljon, päätettiin se jakaa kahteen tiedostoon, jotta selkeys säilyy. Toisessa on pelkästään kommentointi-sivulla vaaditut toiminnallisuudet ja toisessa muilla sivuilla käytetyt toiminnallisuudet. Näin asioita saatiin selkeytettyä huomattavasti.

Projektin suunnittelun aikana löydettiin myös hyviä jQueryn liitännäisiä, joita päätettiin käyttää mahdollisuuksien mukaan hyödyksi sovelluksessa. Jcrop lisättiin sovelluksen kommentointi-sivulle havainnollistamaan kommentoitua aluetta. Jcrop on tehty erittäin helpoksi ottaa käyttöön, sillä se ei tarvitse välttämättä kuin yhden rivin koodia ja jar-tiedoston lisäämisen projektiin. Ohjelma 2 havainnollistaa, kuinka yksinkertaista Jcropin liittäminen helpoimmillaan on. Tässä sovelluksessa tosin Jcropia jouduttiin muokkaamaan erittäin paljon, jotta se sopi juuri tähän tarkoitukseen.

Ohjelma 2. Jcropin liittäminen kuvaan.

```
// Lisää Jcropin kuvaan, jonka id="cropbox"  
JcropApi = $.Jcrop('#cropbox');
```

SOFOKUS ETUSIVU RATKAISU PALVELUT JA TUOTTEET ASIAKKAAT YRITYS REKRYTOINTI AJANKOHTAISTA YHTEYSTIEDOT

19.10.2009 Valtakunnallinen pakinto Yrittäjän tukiverkollle
9.9.2009 Räkkaat verkkosovellukset tuntemattomia V-S:n yrityksille
17.8.2009 Alueellinen Tuottava idea -pakinto Yrittäjän tukiverkollle

KEHITÄ KETTERÄSTI, TUPLAA TUOTTAVUUTESI

Sofokus on suomalainen, innovatiivisiin ja vaativiin sähköisiin liiketoimintaratkaisuihin erikoistunut asiantuntijayritys. Suunnittelemme, toteutamme ja ylläpidämme erilaisiin tarpeisiin räätälöityjä verkkopalveluja ja verkkosovelluksia, jotka tehostavat yritysten ja yhteisöjen toimintaa.

Liiketoimintamme perustuu asiakaslähtöisyyteen ja tavoitteemme on olla ensisijainen yhteistyökumppaninne. Erityisosaamistamme ovat sähköisen liiketoiminnan asiantuntemus sekä teknologinen osaaminen.

Microsoft CERTIFIED Partner **redhat** **SUOMEN VAHVIMMAT** 2009

KOMMENTTI
MITEN VERKKOPALVELUSI OPTIMOINTI KASVATTAA MYYNTIÄSI VÄLITTÖMÄSTI?

SÄHKÖISET LIIKETOIMINTA-RATKAISUT
Etsitkö keinoja parantaa tuottavuutta sisällönhallinnan ratkaisulla?
Liiketoimintaratkaisuja ovat mm. yritysten julkiset verkkopalvelut, extranet/intranetit, portaalit, ja erilaiset ryhmätyö- ja dokumentinhallintaratkaisut.
Ratkaisutarjontamme perustuu alan johtavien toimijoiden kuten Microsoftin tuotepohjaisiin ratkaisuihin.
[Lue lisää](#)

ASIAKASKOHTAISET VERKKOSOVELLUKSET
Asiakaskohtaiset verkkosovellukset integroidaan olemassaoleviin tietojärjestelmiin ja ne tyypillisesti liittyvät toiminnanohjaukseen, verkkokauppaan, tilauksien tai varausten käsittelyyn sekä asiakkuuksien hallintaan.
Verkkosovellukset rakennetaan usein Sofokus iSuite™:n päälle, joka on ketterä, skaalautuva ja monikanavaisuutta tukeva menetelmäpatentoitu verkkopalvelualusta.

KONSULTOINTI- JA KOULUTUSPALVELUT
Tarjoamme konsultointipalveluita, mm. tarvekartoitus ja vaatimusmäärittely, jotka tukevat ja vauhdittavat asiakkaitemme verkkopalveluprojekteja.
Koulutuspalvelumme vahvistavat ja ylläpitävät asiakkaidemme kaupallista ja teknistä osaamista sähköisissä liiketoimintaratkaisuisissa.
[Lue lisää](#)

Nimi:
Yritys:
Sähköposti:
Puhelin:

Kuva 6. Jcropilla näytetään kommentoitu alue kuvasta

Kuvassa 6 on esimerkki sivulla olevasta kommentista. Vihreällä olevat viivat on tehty Jcropilla havainnollistamaan aluetta, jolla kommentti on. Jcropin ansiosta alueen valitseminen on erittäin helppoa, alueen valitseminen toimii juuri samalla tavalla kuin kuvankäsittelyohjelmissä.

Toinen tässä sovelluksessa käytetyistä jQueryn liitännäisistä on Easy Slider. Easy Slider on pitkälti yhtä helposti sovellukseen liitettävä kuin Jcrop. Easy Slider vaatii jonkin verran enemmän ohjelmointia riippuen siitä, mitä sen sivuille laitetaan.



Kuva 7. Kommentti näytettynä Easy Sliderilla

Easy Slideria käytetään tässä sovelluksessa kommenttien näyttämiseen (Kuva 7). Jokaisella kommentilla on oma sivu Easy Sliderissä, joita selataan Easy Sliderin avulla. Kommenttien selaaminen tapahtuu painamalla kuvasta kommentin numeroa tai painamalla Easy Sliderin omasta napista halutun kommentin numeroa. Easy Slider soveltuu erittäin hyvin tässä sovelluksessa kommenttien näyttämiseen.

Sovelluksen käytettävyyden kannalta erittäin suuri parannus oli myös lisätä kommentti vaihtumaan, kun painetaan kommentin numeroa kuvasta. Vaativaksi osoittautui näiden kahden liitännäisen liittäminen yhteen, jotta kommenttien selaaminen olisi helppoa. Asiaa hankaloitti edellisestään myös vielä se että Jcropin tulee havainnollistaa samalla valittuna oleva alue. Ongelmaan kuitenkin keksittiin ratkaisu ja kommenttien selaaminen onnistuu erittäin helposti.

Sovellusta kehitettäessä tuli vastaan, että jQuery sisältää ominaisuuden, jolla voidaan tehdä erittäin helposti Easy Sliderilla tehdystä laatikosta liikuteltava. Tämä osaltaan paransi myös huomattavasti sovelluksen käytettävyyttä, koska laatikon voi aina siirtää haluamaansa kohtaan.

4 SOVELLUKSEN TOIMINTA JA HYÖTY KÄYTTÄJILLE

Tuloksena kehitystyöstä syntyi sovellus, joka parantaa huomattavasti internetsivuja tekevän yrityksen kommunikointia asiakkaan kanssa. Yritys saa tiedon asiakkaan tarpeista huomattavasti helpommin ja nopeammin, kuin perinteisillä kommunikointitavoilla. Kehitystyö paransi myös huomattavasti tekijän taitoa tehdä kyseisiä sovelluksia, sekä antoi paljon hyödyllistä kokemusta sovellusten tekemisestä alusta loppuun.

Sovellukseen kirjaututtaessa sisään avautuu sivu, jossa voidaan lisätä ja muokata projekteja. Jos asiakkaalla on esimerkiksi useampi palvelu tilattuna yritykseltä, niin jokaisesta voidaan tehdä erillinen projekti. Näin saadaan selkeytettyä sovellusta, mikäli yhdelle asiakkaalle tulee paljon kommentoitavia kuvia.



Kuva 8. Esimerkki projekti-sivusta.

Kuvassa 8 näkyy kolme esimerkkiprojektia ja niiden tiedot. Sivun napeista pystytään lisäämään, muokkaamaan ja poistamaan projekteja tarpeen mukaan.

Projekti-sivulta valittaessa jokin projekti, siirrytään siitä seuraavalle sivulle, jossa näytetään kaikki projektiin lisätyt kuvat. Tällä esikatselu-sivulla voidaan lisätä kuvia projektiin, sekä on myös mahdollista poistaa tarpeettomat kuvat. Esikatselu-sivulla kuvien näyttö on tehty dynaamisesti, kuvia näytetään aina tietty määrä riippuen

projektiin valitusta resoluutiosta. Näin käyttäjä saa määriteltyä halutut asetukset itselleen, riippuen esimerkiksi kuinka iso näyttö on käytössä.



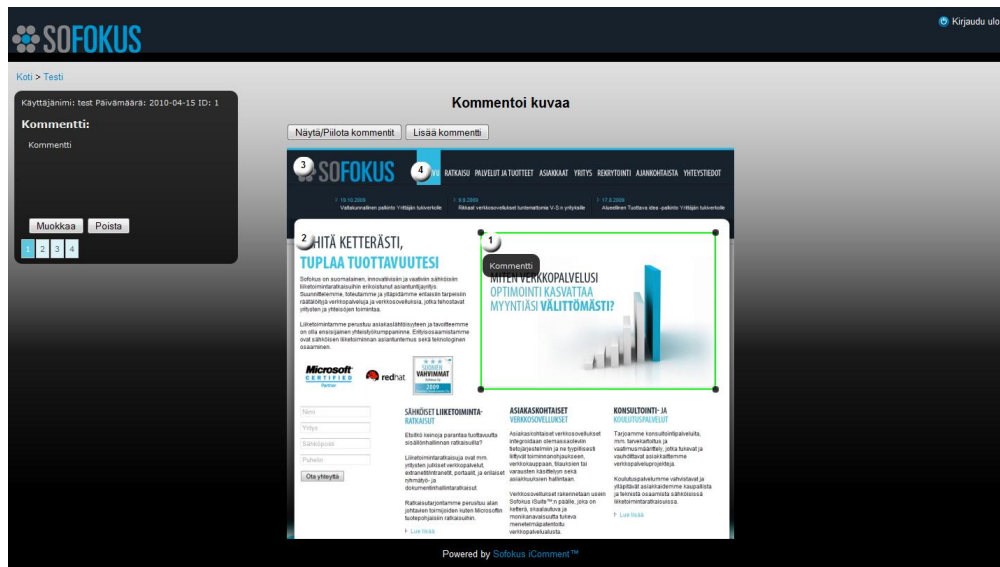
Kuva 9. Esimerkki esikatselu-sivusta

Kuvassa 9 on esimerkki esikatselu-sivusta. Valitun resoluution mukaan sovellus näyttää aina tietyn määrän kuvia esikatselussa. Pudotusvalikosta voidaan valita näytettäväksi myös kaikki kuvat. Mikäli kaikki kuvat eivät mahdu yhdelle sivulle, voidaan kuvia selata kuvien alla olevista napeista. Myös kuvien lisäys ja poisto onnistuu tällä sivulla ylälaudassa olevilla napeilla.

Esikatselu-sivulta valittaessa jokin kuva siirrytään varsinaiselle kommentointi-sivulle. Kommenttien lisäys, muokkaus ja poisto on tehty erittäin yksinkertaiseksi, jotta sovellusta olisi mahdollisimman helppo käyttää. Myös kommenttien selaus toimii erittäin helposti, joko vain painelemalla kuvassa olevia kommenttien numeroita tai selaamalla kommentteja liikkuvan työkalupalkin avulla.

Sovelluksen pääsivuna toimii sivu, jossa kuvan kommentointi tapahtuu. Projektin päätavoitteena oli saada kommentoinnista mahdollisimman helppoa ja yksinkertaista,

jotta sovellusta olisi mukava käyttää. Sivun käytettävyydestä saatiin erittäin hyvä ja kommenttien lisäys kuvaan on erittäin yksinkertaista. Projektin alussa sivusta suunniteltiin jonkin verran erilainen, mutta ensimmäisen version valmistuttua testien aikana tuli paljon parannusideoita mieleen. Sivua uudistettiin aika paljon, jonka jälkeen sovelluksen käytöstä tulikin huomattavasti edellistä versiota helpompaa ja käytännöllisempää.



Kuva 10. Esimerkki kommentointi-sivusta

Kuten kuvasta 10 nähdään, sivu on erittäin pelkistetty, jotta käytettävyys olisi mahdollisimman hyvä. Käytettävyyden kannalta suurin parannus lopulliseen versioon oli tehdä vasemmalla olevasta kommenttipaneelista liikuteltava. Käyttäjä voi siirtää paneelin aina sinne mihin sen itse haluaa. Etenkin isoja kuvia kommentoitaessa tämä on erittäin hyvä ominaisuus. Käyttäjän ei tarvitse aina selata ikkunassa sinne missä paneeli sijaitsee, vaan paneeli voi olla aivan kommentoitavan alueen vieressä.

Kaikki kommentointiin liittyvät asiat integroitiin kyseiseen paneeliin, jotta käytettävyys olisi mahdollisimman hyvä, eikä käyttäjän tarvitse selailta ympäri sivua löytääkseen kunkin toiminnallisuuden.

5 YHTEENVETO

Tässä työssä suunniteltiin ja toteutettiin internetsivuja kehittävä Sofokus Oy:lle internetpohjainen sovellus, jolla voidaan kommentoidaan yrityksen valmistamia internetsivuja. Sovellus parantaa huomattavasti yrityksen ja asiakkaiden välistä kommunikaatiota, tuoden siihen visuaalisen puolen.

Lopputuotoksena saatiin erittäin hyvin toimiva sovellus, josta on suurta hyötyä yritykselle ja sen asiakkaille. Sovelluksen avulla ohjelmistokehittäjä saa huomattavasti nopeammin rakentavaa ja hyvää palautetta kehitettävistä sivuista. Sovelluksen tuodessa kommunikaatioon visuaalisen puolen ohjelmistokehittäjä myös ymmärtää helpommin asiakkaan tarpeet.

Monilla yrityksillä on edessään samankaltaisia ongelmia, joten ohjelman jatkokehityksellä ja kaupallistamisella olisi hyvät näkymät. Kaupalliseen tarkoitukseen sovellusta tulisi tosin kehittää huomattavasti, jotta se soveltuisi mahdollisimman moneen asiaan, eikä vain internetsivujen kommentointiin. Näin saataisiin markkinoitua sovellusta mahdollisimman moneen asiaan ja asiakaskunta olisi mahdollisimman suuri.

Sovellus on jo ensimmäisessä versiossaan erittäin monikäyttöinen. Internetsivujen lisäksi siitä olisi varmasti myös erittäin paljon hyötyä mainostoimistoille. Mainostoimisto voisi laittaa luonnosversion esimerkiksi mainoksesta kyseiseen palveluun, johon asiakas sitten voisi käydä laittamassa parannusehdotuksia ja palautetta. Jatkokehityksellä sovelluksesta saisi erittäin monipuolisen ja melkein vain taivas on rajana mietittäessä, mihin kaikkeen sovellusta voisi käyttää.

LÄHTEET

- [1] Holzner, Steven. "Java". Paraglyph Press. 2001.
- [2] Chopra, Vivek. Eaves, Jon. Jones, Rupert. "Beginning JavaServer Pages". John Wiley & Sons, Incorporated. 2005.
- [3] "The Java Servlet API White Paper". [www-dokumentti]. Saatavilla: <http://java.sun.com/products/servlet/whitepaper.html> (Luettu: 12.4.2010)
- [4] "JavaServer Pages Technology". [www-dokumentti]. Saatavilla: <http://java.sun.com/products/jsp/?intcmp=3184> (Luettu: 12.4.2010)
- [5] ".NET Framework Overview". [www-dokumentti]. Saatavilla: <http://www.microsoft.com/net/overview.aspx> (Luettu: 12.4.2010)
- [6] "PHP Manual". [www-dokumentti]. Saatavilla: <http://fi2.php.net/manual/en/index.php>. (Luettu: 12.4.2010)
- [7] "JavaScript Introduction". [www-dokumentti]. Saatavilla: http://www.w3schools.com/js/js_intro.asp (Luettu: 18.4.2010)
- [8] "About MySQL". [www-dokumentti]. Saatavilla: <http://www.mysql.com/about/> (Luettu: 20.4.2010)
- [9] "JBoss Web - Introduction". [www-dokumentti]. Saatavilla: <http://www.jboss.org/jbossweb/introduction.html> (Luettu: 20.4.2010)
- [10] Chaffer, Jonathan. Swedberg, Karl. "Learning jQuery 1.3". Packt Publishing Ltd. 2009
- [11] "Jcrop - Manual". [www-dokumentti]. Saatavilla: http://deepliquid.com/content/Jcrop_Manual.html (Luettu: 1.5.2010)
- [12] "jQuery plugin – Easy Image or Content Slider". [www-dokumentti]. Saatavilla: <http://cssglobe.com/post/3783/jquery-plugin-easy-image-or-content-slider> (Luettu: 5.5.2010)

LIITTEET

- Liite 1 User-luokan tapahtumakuvaus Java-tiedostolla - User.java
- Liite 2 User-luokan kuvaus Hibernatelle - User.hbm.xml
- Liite 3 Hibernaten asetustiedosto - hibernate.properties
- Liite 4 Esimerkki tietokantahausta Java Servletillä - ProjectServlet.java
- Liite 5 Esimerkki sivun koostamisesta JSP:lla – project.jsp

User-luokan tapahtumakuvaus Java-tiedostolla

```
package com.sofokus.imagecomment.beans;

/**
 * User-luokan tapahtumakuvaus.
 *
 * @author Jori Jokinen
 */

public class User {
    private int id;
    private String username;
    private String password;
    private String email;

    public User() {}

    public int getId() {
        return id;
    }
    private void setId(int id) {
        this.id = id;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
}
}
```

User-luokan kuvaus Hibernatelle

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
  "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>

  <class name="com.sofokus.imagecomment.beans.User" table="user">
    <id name="id" column="id">
      <generator class="native"/>
    </id>
    <property name="username"/>
    <property name="password"/>
    <property name="email"/>
  </class>

</hibernate-mapping>
```


Hibernaten asetustiedosto

hibernate.connection.driver_class = com.mysql.jdbc.Driver

hibernate.connection.url = jdbc:mysql://localhost:3306/comment_image

hibernate.connection.username = xxx

hibernate.connection.password = xxx

hibernate.dialect = org.hibernate.dialect.MySQLDialect

hibernate.current_session_context_class = thread

Esimerkki tietokantahausta Java Servletillä

```
package com.sofokus.imagecomment.project;

import org.hibernate.Query;
import org.hibernate.Session;

import com.sofokus.imagecomment.utils.*;
import com.sofokus.imagecomment.beans.*;
import com.sofokus.imagecomment.controller.ControllerAbstract;

import java.util.*;
import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

/*
 * Haetaan käyttäjäkohtaiset projektit tietokannasta.
 *
 * @author Jori Jokinen
 */

public class ProjectServlet extends ControllerAbstract {
    @Override
    public void doGet(HttpServletRequest request,
        HttpServletResponse response, SessionBean session) throws
        ServletException, IOException {

        // Hae userId sessiosta
        Integer userId = session.getUser().getId();

        // Avaa hibernaten sessio
        Session hibernateSession =
            HibernateUtil.getSessionFactory().getCurrentSession();
        hibernateSession.beginTransaction();

        // Hae kaikki käyttäjän projektit tietokannasta
        Query q = hibernateSession.createQuery(" from Project where
            user_id = ?");
        q.setInteger(0, userId);
        List<Project> projects = new ArrayList<Project>(q.list());

        // Lisää listaan resolution nimi
        for (Project project : projects) {
            q = hibernateSession.createQuery(" from
                Resolution where id = ?");
            q.setInteger(0, project.getResolution_id());
            List<Resolution> resolutions = new
                ArrayList<Resolution>(q.list());
            project.setResolution(resolutions.get(0));
        }

        hibernateSession.getTransaction().commit();
    }
}
```

```
// Ohjaa projekti-sivulle halutuilla parametreilla  
  
request.setAttribute("projects", projects);  
String path = "/jsp/content/project/project.jsp";  
RequestDispatcher dispatcher =  
request.getRequestDispatcher(path);  
dispatcher.forward(request, response);  
}  
  
@Override  
public void doPost(HttpServletRequest request,  
HttpServletRequest response, SessionBean session) throws  
ServletException, IOException {  
}  
}
```

Esimerkki sivun koostamisesta JSP:lla

```

<%@ taglib uri="http://tiles.apache.org/tags-tiles" prefix="tiles" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<tiles:insertDefinition name="section">
<tiles:putAttribute name="title" value="Project"/>
<tiles:putAttribute name="maincontent">
<h2><fmt:message key="ProjectHeader"/></h2>

<div class="project-container">
<div class="project-table">
<div class="project-add">
<button id='add-project'><fmt:message key="AddProject"/></button>
</div>

<table class="project">
<tr>
<th width="300"><fmt:message key="Name"/></th>
<th width="100"><fmt:message key="Resolution"/></th>
<th width="160"><fmt:message key="Edit"/></th>
</tr>

<c:forEach var="c" items="${projects}" varStatus="loop">
<tr>
<td>
<a href="<c:url value='/preview/?projectId=${c.id}'/>">${c.name}</a>
<input class="project-id" name="project-id" value="${c.id}"
type="hidden"/>
</td>

<td>${c.resolution.name}</td>

<td>
<button class='edit-project' value='${c.id}'><fmt:message
key="Edit"/></button>

<button class='delete-project' value='${c.id}'><fmt:message
key="Delete"/></button>
</td>
</tr>
</c:forEach>
</table>
</div>
</div>
</tiles:putAttribute>
</tiles:insertDefinition>

```