



Satakunnan ammattikorkeakoulu

Hermann Piltti

2008

ASIAKASPALAUTEJÄRJESTELMÄ

Tekniikka Rauma

Tietotekniikan koulutusohjelma

ASIAKASPALAUTEJÄRJESTELMÄ

Piltti, Juha Hermann
Satakunnan ammattikorkeakoulu
Tekniikka Rauma
Tietotekniikan koulutusohjelma
Asiakas: Raumaster Paper Oy
Työn valvoja yrityksessä: Petri Laukkanen
Toukokuu 2008
Ohjaaja: Mikko Javanainen
UDK: 004.056, 004.49
Sivuja: 34

Asiasanat: palaute, PHP, SQL, WWW, tietoturva

Opinnäytetyön tarkoituksena oli luoda sähköisessä muodossa oleva dynaaminen ja turvallinen asiakaspalautejärjestelmä Raumaster Paper Oy:lle.

Työn alkuvaiheisiin kuuluivat koko järjestelmän tarkka määrittely ja suunnittelu. Asiakaspalautejärjestelmään suunniteltiin kaksi eri osiota, asiakas- sekä hallintapuoli. Suurin osa työmäärästä kului tietokantojen ja hallintasivujen suunnittelemiseen sekä koko järjestelmän dynaamiseksi toteuttamiseen. Tietokanta suunniteltiin MySQL-järjestelmän päälle.

Ohjelmointivaiheessa työssä keskityttiin dynaamisen järjestelmän luomiseen kuin myös tietoturvaan. Tietoturvaa luotiin käyttämällä PHP-koodaukseen liittyviä komentoja ja menetelmiä. Kaikki järjestelmään tehdyt sivut toteutettiin kyseisellä ohjelmointikielellä.

Lopputuloksena saatiin aikaan asiakkaalle helppokäyttöinen, mutta silti yritykselle dynaaminen ja turvallinen asiakaspalautejärjestelmä.

CUSTOMER FEEDBACK SYSTEM

Piltti, Juha Hermanni

Satakunta University of Applied Sciences

School of Technology Rauma

Information Technology

Commissioned by: Raumaster Paper Oy

Supervisor: Petri Laukkanen, Engineering Manager, Mechanics

May 2008

Tutor: Mikko Javanainen, Lecturer

UDC: 004.056, 004.49

Pages: 34

Keywords: feedback, PHP, SQL, WWW, security

The purpose of this Bachelor's thesis was to create an electronic and safe customer feedback system for Raumaster Paper Oy.

The initial stage of this project was to design and define the whole customer feedback system accurately. The customer feedback system was divided into two separate parts during the design process of the system. These parts were the customer feedback and the system management part. Most of the working hours were spent on designing a system which would be both dynamic and safe at the same time. The security was achieved by using the well-known security methods of PHP. All the web pages of the system were made by using PHP and MySQL.

The final result of this project was an easy-to-use customer feedback system which was safe and dynamic for company use.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

SYMBOLI- JA TERMILUETTELO

1	JOHDANTO	7
2	OPINNÄYTETYÖONGELMA.....	8
2.1	Tarkoitus	8
2.2	Työkalut	8
2.2.1	PHP	8
2.2.2	MySQL.....	10
2.3	Aineisto	10
2.4	Aikataulu.....	10
2.5	Yrityksen esittely	10
3	JÄRJESTELMÄN SUUNNITTELU JA MÄÄRITTELY.....	12
3.1	Rakenteen suunnittelu	12
3.2	Taulujen määrittely	13
3.2.1	Kysymykset.....	13
3.2.2	Tunnukset.....	13
3.2.3	Vastaukset	14
3.2.4	Vastaukset_text	15
3.3	Taulujen fyysinen luonti	15
3.3.1	Kysymykset.....	15
3.3.2	Tunnukset.....	16
3.3.3	Vastaukset	17
3.3.4	Vastaukset_text	18
4	WEB-LOMAKKEET.....	19
4.1	Lomakkeiden suunnittelu	19
4.1.1	Hallintalomakkeet	19
4.1.2	Asiakaspalautelomake.....	21
4.2	Lomakkeiden toteutus	22
4.2.1	Hallintalomakkeet	22
4.2.2	Asiakaspalautelomake.....	28

5	TIETOKANNAN HYÖDYNTÄMINEN	32
6	YHTEENVETO	33

SYMBOLI- JA TERMI LUETTELO

APACHE	Avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma Linux-ympäristöön
BOOLEAN	Totuusarvo tietotyyppi, arvoina 0 tai 1
HTML	Hypertext Markup Language, on avoimesti standardoitu kuvauskieli. Tunnetaan erityisesti kielenä, josta web-sivut rakentuvat
HTTP	Hypertext Transfer Protocol, on protokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoon
IIS	Internet Information Services, on Microsoftin kehittämä palvelinohjelmistokokonaisuus, joka on tarkoitettu käytettäväksi Windows-pohjaisissa palvelimissa
INTEGER	Kokonaisluku tietotyyppi, arvoina esim. -120 tai 14
MD5	Message-Digest, algoritmi, jota käytetään merkkijonojen kryptaamiseen
MySQL	Avoimeen lähdekoodiin perustuva suosittu ja tehokas SQL-tietokannan hallintajärjestelmä
ORACLE	Suosittu tietokantaohjelmisto
PHP	PHP: Hypertext Preprocessor, ohjelmointikieli, jota käytetään erityisesti web-palvelinympäristöissä dynaamisten web-sivujen luonnissa
SESSION	Istunto php-ohjelmointikielessä
SQL	Structured Query Language, kyselykieli relaatiotietokantoihin
URL	Uniform Resource Locator, käytetään osoittamaan WWW-sivuja
VARCHAR	Merkkijono tietotyyppi, arvoina esim. ”merkkijono” tai ”a2”
WEB	Tarkoittaa samaa kuin WWW
WWW	World Wide Web, yksi Internetin palvelumuoto

1 JOHDANTO

Valitsin opinnäytetyöni aiheeksi sähköisessä muodossa olevan asiakaspalautejärjestelmän. Olen työskennellyt Raumaster Paper -yrityksessä nyt toista vuotta. Koska koulusta valmistumiseni alkoi hämmöittää lähitulevaisuudessa, kysyin töissä ollessani, onko mahdollista tehdä opinnäytetyö heille. Selvisi, että tarvetta olisi toimivalle asiakaspalautejärjestelmälle. Löysin työn aihetta etsiessäni yrityksen vanhan paperiversion asiakaspalautelomakkeista, ja silloin mieleeni heräsi ajatus, että eihän tämä ole nykypäivää eikä kovin käytännöllistäkään. Näin ollen päätin viedä ideani asiakaspalautejärjestelmän uudistamisesta eteenpäin ja ehdotin esimiehelleni Petri Laukkaselle, voisinko tehdä uuden ja toimivamman asiakaspalautejärjestelmän. Tähän ehdotukseen yrityksessä oltiin tyytyväisiä.

Aloin pohtimaan miten paperiversiolle aikoinaan toteutetun asiakaskyselyn voisi toteuttaa nykyaikaisemmalla tavalla. Ensimmäinen ajatus uudesta järjestelmästä oli sellainen, että sen pitäisi olla niin yksinkertainen ja helppo käyttää, jotta asiakaskyselyyn vastaaminen olisi asiakkaille miellyttävää ja että he todella tekisivät sen. Aikani tätä mietittyäni, päädyin ratkaisuun, joka olisi web-pohjainen. Tässä ratkaisussani asiakkaalle annettaisiin selaimella täytettäväksi lomake, jolle asiakas antaisi 1-5 arvosanoja tiettyihin kysymyksiin. Samalla alkoi kehittyä ajatus itse fyysisestä tietokannasta ja järjestelmän toiminnasta.

Asiakaspalautejärjestelmän perusajatushan on luoda yritykselle informaatiota siitä, mitä mieltä heidän asiakkaansa ovat yrityksen onnistumisesta projektissa ja sen toimittamisesta asiakkaalle. Tätä informaatiota asiakkailta pitää kerätä vuosia, ennen kuin varsinaisia tuloksia alkaa selkeästi näkyä. /1, s. 4,6/

2 OPINNÄYTETYÖONGELMA

2.1 Tarkoitus

Työn tarkoituksena on luoda yritykselle dynaaminen, helposti hallittava ja myös asiakkaiden kannalta miellyttävä asiakaspalautejärjestelmä sähköiseen muotoon. Lopputuloksena yritykselle muodostuu ajan kuluessa, asiakaspalautteiden kerryttyä, tärkeää tietoa koskien projektien laatua niiden alkuvaiheesta aina projektien päättymiseen asiakkaalle valmiina tuotteina. Täten yritys voi nähdä kehityksen suuntaa, missä se on onnistunut ja missä taas ei.

2.2 Työkalut

Järjestelmän alustaksi valitsin ilmaiset työkalut PHP- sekä MySQL-serverin. Päädyin näihin työkaluihin siksi, että yrityksellä on jo olemassa WWW-sivut palvelimella, jossa toimii PHP-serveri. MySQL-serveri toimii toisella palvelimella.

2.2.1 PHP

Alun perin PHP (Hypertext Preprocessor) oli eräänlainen palvelimen puolella ajettavien komentojen kokoelma WWW-pohjaisia sovelluksia varten. Ensimmäisestä versiosta lähtien PHP:n kehitys on ollut nopeaa ja nykyään tätä dokumenttia kirjoittaessa PHP:sta on julkaistu versio 5.2.5. PHP:n versio neljä oli jo täysiverinen ohjelmointikieli, mutta PHP5 on tuonut mukanaan paljon uutta tehoa ja toimintoja edelliseen verrattuna. PHP sisältää kaikki ohjelmointikielelle tyypilliset rakenteet eli esimerkiksi luokat, funktiot, for- ja while-silmukat jne. PHP perustuu avoimeen lähdekoodiin ja se on ilmainen. /2, s. 16-17/

PHP-koodia voidaan kirjoittaa tavallisen HTML-koodin sekaan. Erona normaaliin HTML-sivuun on se, että aina kun asiakas saapuu WWW-sivulle, missä on PHP-koodia, palvelin suorittaa PHP-koodin. Palvelin muodostaa HTML-sivun normaalin HTML-koodin ja PHP-koodin yhdistelmänä ja vasta sen jälkeen lähettää muodostetun

sivun takaisin asiakkaan selaimeen. Tämän takia asiakas ei koskaan näe PHP-koodia katsoessaan HTML-sivun lähdekoodia. Juuri tämän takia päädyin työssäni PHP-ohjelmointiin, koska sillä voi luoda monimutkaisiakin tietokantahakuja ja tulostaa juuri tietynlainen HTML-sivu asiakkaalle. /2, s. 16-17/

PHP-serveri voidaan asentaa toimintakuntoon joko Linux-ympäristössä Apache-WWW-palvelimen päälle tai Windows-ympäristössä IIS-WWW-palvelimen päälle. Vaikka PHP-koodi ajetaan palvelimen päässä, on PHP-sivujen ajaminen suhteellisen nopeaa, koska erillistä koodin tulkin käynnistämistä ei tarvita. /2, s. 16, 49, 50/

PHP-kieltä voidaan käyttää monien tietokantojen kanssa. Tähän listaan kuuluvat mm. Oracle, Sybase, Solid, mSQL, MySQL, PostgreSQL ja Microsoft SQL. Täydellinen listaus toimivista tietokannoista PHP:n kanssa löytyy osoitteesta <http://www.php.net>. Tässä työssä käytetään MySQL tietokantaa. /2, s. 49/

2.2.1.1 PHP:n turvallisuus

Aina, kun kyse on nettisovelluksista, täytyy ottaa tietoturva-asiat huomioon. PHP alustana ei ole sen turvallisempi tai turvattomampi kuin muutkaan WWW-sovelluksiin tarkoitetut ohjelmointikielät. /2, s. 60/

Koska PHP-sivun tuottama sisältö riippuu kovin usein käyttäjän antamasta syötteestä lomakkeilta, on ensisijaisen tärkeää, että syötettä ei luulla oletusarvoisesti turvalliseksi. Syötteen oikeellisuus pitäisi aina tarkastaa, ennen kuin sitä käytetään ohjaamaan sovelluksen toimintaa. Jos syötteitä ei tarkisteta, voi syötteen antaja päästä käsiksi esimerkiksi palvelimen kaikkiin tiedostoihin tietyissä tilanteissa. Kun käyttäjältä estetään tietoturva-aukkojen hyödyntäminen, siitä ei ole suositeltavaa kuitenkaan mainita käyttäjälle. Koska jos käyttäjälle ilmoitetaan, että juuri tämä ja tämä asia on estetty, voivat ilmoitukset paljastaa käyttäjälle muita potentiaalisia turva-aukkoja sovelluksessa. /2, s. 60, 61, 63/

2.2.2 MySQL

MySQL on tehokas, nopea ja varma avoimeen lähdekoodiin perustuva relaatiotietokantojen hallintajärjestelmä. Se, mikä tekee edellä mainittavien asioiden lisäksi MySQL:stä suosituksen maailmalla, on sen hinta: maksuton. MySQL soveltuu hyvin niin pienten kuin isojen, ruuhkaisten sivustojen käyttöön. /3, s. 15/

Relaatiotietokantojen hallintajärjestelmä koostuu tietokannoista, tauluista ja niiden sarakkeista sekä riveistä datoineen. Relaatiotietokanta tarkoittaa joukkoa yhteen liitettyistä tauluista, jotka ovat suhteessa toisiinsa tiettyyn sarakkeeseen merkittyjen arvojen pohjalta. Yhteen liittäminen tapahtuu käyttämällä avaimia. Suurin hyöty relaatioista on juuri se, että on mahdollista pilkkoa suuri määrä dataa pieniin toisiinsa liitettyihin tauluihin, välttämällä samalla toistamasta samaa dataa monessa eri paikassa ja täten säilyttämään datan yhtenäisyys. /3, s. 8, 15/

2.3 Aineisto

Ohjelmointiin perustuva aineisto on itse keräämääni Internetistä, koulusta sekä kirjastosta. Teoriaan perustuva aineisto on kerätty pääosin kirjallisuudesta.

2.4 Aikataulu

Työn aihetta suunniteltiin vuoden 2007 marras-joulukuussa, jonka jälkeen vielä tarkasteltiin aihetta kokonaisuutena ja selvitettiin vaatimukset ja rajattiin tarkemmin työn laajuus tammikuussa 2008. Työn virallinen aloituspäivä on 25.1.2008, ja työ valmistuu vuoden 2008 toukokuussa.

2.5 Yrityksen esittely

Raumaster Paper Oy on raumalainen 40 henkilöä työllistävä yritys. Yritys perustettiin vuonna 2003 liiketoimintasiirtona Raumaster Oy:stä, jonka osastona se oli siihen asti toiminut Raumaster Oy:n perustamisesta lähtien.

Raumaster Paper myy, markkinoi, suunnittelee ja toimittaa paperirullien ja arkkilavojen käsittely- ja pakkausjärjestelmiä sekä hylsyn sahaus- ja käsittelyjärjestelmiä paperiteollisuuteen. Asiakaskunta koostuu monikansallisista paperia valmistavista konserneista ympäri maailman. Laitekirjoon kuuluvat mm. erilaiset kuljettimet, käärintäkoneet, hylsysahat ja automaattiset lajittelujärjestelmät automaatiojärjestelmineen ja valvomoineen.

Raumaster Paperin pääkonttori sijaitsee Raumalla, ja oma kokoonpano/testausyksikkö on vastikään aloittanut toimintansa Porissa. Yrityksen vuotuinen liikevaihto on noin 12 miljoonaa euroa. /4/

3 JÄRJESTELMÄN SUUNNITTELU JA MÄÄRITTELY

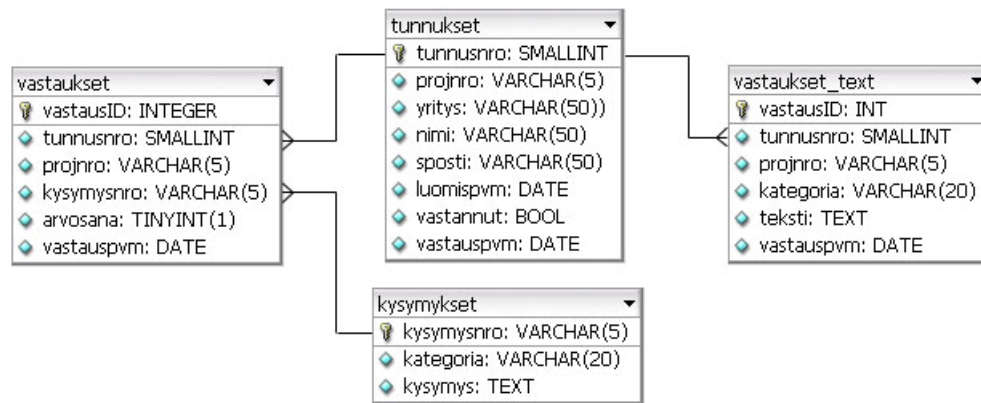
3.1 Rakenteen suunnittelu

Itse fyysisen tietokannan ja siihen liittyvien taulujen suunnittelu ovat tärkeä osa työtä, sillä jos tietokannan taulut suunnitellaan huonosti, se näkyy mm. järjestelmän hitautena ja jopa toimimattomuutena. Tietokannan huolimaton suunnittelu vaikuttaa myös ylläpitävyyteen. Näiden edellä mainittujen seikkojen huomioon ottaminen hankaloitti tietokannan suunnittelua, sillä järjestelmästä piti tehdä vielä dynaaminen. Dynaamisuus tässä tapauksessa tarkoittaa sitä, että jälkepäin on pystyttävä lisäämään uusia kysymyksiä, eikä järjestelmä saa olla lukittu vain yhteen muottiin, eli tässä tapauksessa lukittuna tiettyyn määrän tiettyjä kysymyksiä. Asiakkaalle luodaan täten aina viimeisin versio asiakaspalautelomakkeesta tietokannan tietojen perusteella. /3, s. 28/

Nämä kriteerit huomioon ottaen päädyin lopulta ratkaisuun, jossa on neljä eri taulua: tunnukset, kysymykset, vastaukset sekä oma taulu vastaukset_text ylimääräisille vapaa-valinnaisille tekstikenttävastauksille. Tämän ratkaisun valitsin siksi, että jälkepäin voidaan lisätä kysymyksiä Kysymykset-tiluun sekä tarkastella Tunnuksset-tilusta sitä, ketkä asiakkaista ovat vastanneet kyselyyn ja milloin. Tunnuksset-tilu toimii samalla myös paikkana, mistä tarkistetaan, onko sivulle tulijalla oikeudet suorittaa palautteen teko. Tietokannasta on tämän ratkaisun avulla myös helpompi hakea kysymykset kyselylomakkeelle.

Tietokannan relaatiot eri taulujen välillä ovat tässä työssä ainoastaan monen suhde yhteen -tyylisiä. Relaatioita tarvitaan vastaukset tauluissa, koska niissä olevat kentät, tunnusnro ja kysymysnro, viittaavat Tunnuksset- ja Kysymykset-tilujen pääavainkenttiin. Monen suhde yhteen -relaatio tarkoittaa sitä, että Vastaukset-tilussa voi olla monta vastausta samaan kysymykseen, mutta yksi kysymysnumero voi viitata ainoastaan yhteen kysymykseen.

Kuvassa 1 ovat esillä taulujen suunnittelukaaviot, relaatiot sekä kenttien tietotyypit.



Kuva 1. Järjestelmän tietokannan rakennemalli.

3.2 Taulujen määrittely

3.2.1 Kysymykset

Kysymykset-taulun kysymysno-kentän tietotyyppiä valittiin *varchar(5)*, koska tähän kenttään kysymyksien numerot tulevat muodossa ”1.1”. Koska merkkejä mahtuu 5 kpl, tästä seuraa, että kysymyksen kysymysnumero voi olla maksimissaan ”99.99”. Piste vasemmalla puolella on kategoriaa kuvaava numero ja pisteen oikealla puolella kategorian kysymyksen numero.

Seuraavana on kategoria-kenttä, jonka tietotyyppiä valittiin *varchar(20)*, eli 20 merkin mittainen tila. Kysymykset-taulun kysymys-kentälle on varattu *text*-tietotyyppiä oleva koko. Siihen mahtuu $2^{16} = 65\,536$ tavua. Näin ollen pitempikin kysymys on mahdollista lisätä Kysymykset-tauluun.

3.2.2 Tunnukset

Tunnukset-tauluun muodostetaan käyttäjätunnukset asiakkaille. Tätä taulua käytetään asiakaspalautelomaketta ladattaessa tarkistamaan, onko sivua lataavalla henkilöllä oikeudet päästä sivulle.

Taulu koostuu tunnusnro-, projnro-, yritys-, nimi-, sposti-, luomispvm-, vastannut- ja vastauspvm-kentistä. Tunnusnro-kentän tietotyyppiä valittiin *smallint*. Seuraavana on projnro-kenttä, mihin mahtuu tietotyyppin *varchar(5)* mukaan viiden merkin mittainen merkkijono. Tämä merkkimäärä riittää, sillä Raumaster Paperin projektinumerot ovat muotoa Axxxx, missä x-kirjaimet ovat numeroita.

Yritys-, nimi- ja sposti-kenttien tietotyypeiksi valittiin 50 merkin mittainen *varchar*-tietotyyppi. Tämän jälkeen vuorossa ovat luomispvm-, vastannut- ja vastauspvm-kentät. Päivämääräkenttien tietotyyppiä valittiin luonnollisesti *date*-tietotyyppi ja vastannut-kenttä määritellään *bool*, eli *boolean*-tyyppiseksi. *Boolean*-tietotyyppin valinnan perusteluna on se, että vastannut-kenttää käytetään merkkimaan ainoastaan se asia, onko asiakas täyttänyt asiakaspalautelomakkeen (0=ei, 1=kyllä). Toista täyttökertaa ei sallita.

3.2.3 Vastaukset

Vastaukset-taulu on tässä tietokannassa se keskeisin taulu, mihin asiakkaiden kaikki arvosanavastaukset lisätään ja mistä ne myöhemmin voidaan lukea.

Taulu koostuu vastausID-, tunnusnro-, projnro-, kysymysnro-, arvosana- ja vastauspvm-kentistä. VastausID-kentän numeroksi valittiin *int unsigned*-tietotyyppi, koska vastauksia saattaa tulla todella paljon. Jos tietokannassa olisi kysymyksiä 100 kpl, siitä seuraa, että jokainen asiakas joka antaa palautteen, muodostaa näin ollen 100 kpl rivejä Vastaukset-tauluun. 10 kpl asiakasvastauksia veisi näin ollen 1000 riviä Vastaukset-taulun koosta. Koska vastausID-kentän tietotyyppiä varattiin *int unsigned*, vastauksia mahtuu näin ollen $2^{32} = 4\,294\,967\,295$ kappaletta.

Vastaukset-taulun tunnusnro-kenttään tulee asiakkaan tunnusnumero, mikä on loogisesti kytketty Tunnuksset-taulun tunnusnro-kenttään monen suhde yhteen -relaatiolla. Siksi tunnusnro-kentällä on sama tietotyyppi kuin Tunnuksset-taulun tunnusnro-kentällä.

Projnro-kentälle valittiin tietotyyppi *varchar(5)* kuten aikaisemminkin. Projektinnumero tulee Vastaukset-tauluun palautelomakkeen kautta.

Seuraavana määritellään kysymysnro-kenttä, joka on samaa tietotyyppiä kuin Kysymykset-aulussa oleva kysymysnro-kenttä. Näin on tehty siksi, että tähän kenttään tulee pelkästään kysymyksen numero, joka on kytketty loogisesti Kysymykset-aulun kysymysnro-sarakkeeseen monen suhde yhteen -relaatiolla.

Toiseksi viimeinen kenttä Vastaukset-aulussa on arvosanakenttä. Siihen riittää tietotyyppi *tinyint(1) unsigned*, eli yksi numero jonka arvo on väliltä 0-9. Tähän tietotyyppiin valintaan päädytään, koska HTML-pohjaiselta palautelomakkeelta tulee radiopainikkeiden kautta pakotetusti valitut arvosanat eli arvot väliltä 1-5.

Viimeinen kenttä Vastaukset-aulussa on vastauspvm eli päivämäärä. Tähän kenttään tulee vastauksen päivämäärä, jotta tiedetään koska vastaus on annettu.

3.2.4 Vastaukset_text

Vastaukset_text-aulu on pelkästään asiakkaan valinnaista vapaamuotoista tekstipalautetta varten. Taulu koostuu vastausID-, tunnusnro-, projnro-, kategoria-, teksti- sekä vastauspvm-kentistä. Kaikkien muiden kenttien tietotyyppi on sama kuin Vastaukset-aulukossakin, erona vain se, että taulussa on teksti-niminen kenttä, jonka tietotyyppiä valittiin *text*, jotta myös pitempi palaute on mahdollista ottaa vastaan asiakkaalta. /5, 6/

3.3 Taulujen fyysinen luonti

Seuraavaksi taulut pitää luoda fyysisesti tietokantaan. Esittelen seuraavaksi taulujen luontikoodit ja sen jälkeen kerron, mitä luontikoodit tekevät.

3.3.1 Kysymykset

```
CREATE TABLE kysymykset (
    kysymysnro VARCHAR(5) NOT NULL UNIQUE PRIMARY KEY,
    kategoria VARCHAR(20) NOT NULL,
    kysymys TEXT NOT NULL
);
```

Yläpuolella oleva koodinpätkä luo tietokantaan Kysymykset-nimisen taulun. Sen pääavaimeksi luodaan kysymysnro-niminen kenttä, joka on tietotyypiltään määrittelemäni *varchar(5)*. Kenttä ei voi koskaan olla tyhjä, ja tämän lisäksi kysymysnumeron on oltava uniikki.

Seuraavat kaksi koodiriviä määrittelevät kentät kategoria ja kysymys. Kenttiin määritellään *varchar(20)*- ja *text*-tietotyypit, eli kategoria voi olla 20 merkkiä pitkä ja kysymykselle on varattu paljon isompi tila, kuten aikaisemmin tässä dokumentissa olen maininnutkin.

3.3.2 Tunnukset

```
CREATE TABLE tunnukset (
    tunnusno SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT
    PRIMARY KEY,
    projnro VARCHAR(5) NOT NULL,
    yritys VARCHAR(50) NOT NULL,
    nimi VARCHAR(50) NOT NULL,
    sposti VARCHAR(50) NOT NULL,
    luomispvm DATE NOT NULL,
    vastannut BOOL NOT NULL DEFAULT 0,
    vastauspvm DATE NULL DEFAULT null
);
```

Yllä olevalla koodilla luodaan tietokantaan Tunnukset-niminen taulu. Ensimmäiseksi luodaan tunnusno-kenttä, josta muodostetaan pääavain ja kentän tietotyyppi on *smallint unsigned*. Kenttä ei voi olla tyhjä, ja kentän numero kasvaa automaattisesti lisättäessä tunnuksia tauluun.

Tämän jälkeen luodaan projnro-kenttä, ja sen tietotyyppi on *varchar(5)*. Kentät yritys, nimi ja sposti määritellään tietotyypeiksi *varchar(50)*. Kentät eivät saa olla tyhjiä. Lopuksi luodaan vielä luomispvm-, vastannut- ja vastauspvm-kentät. Luomispvm- ja vastauspvm-kenttien tietotyyppiä valitaan *date*, eli päivämäärä-muotoa oleva kenttä. Vastannut-kenttä saa tietotyypin *boolean*.

3.3.3 Vastaukset

```
CREATE TABLE vastaukset (
    vastausID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    tunnusno SMALLINT UNSIGNED NOT NULL,
    projnro VARCHAR(5) NOT NULL,
    kysymysno VARCHAR(5) NOT NULL,
    arvosana TINYINT(1) UNSIGNED NOT NULL,
    vastauspvm DATE NOT NULL,
    CONSTRAINT vastaukset_FK_A
    FOREIGN KEY (tunnusno)
    REFERENCES tunnuksset (tunnusno),
    CONSTRAINT vastaukset_FK_K
    FOREIGN KEY (kysymysno)
    REFERENCES kysymykset (kysymysno)
);
```

Yllä olevalla koodilla luodaan Vastaukset-niminen taulu, jonka pääavaimeksi asetetaan vastausID-kenttä. Kentän tietotyyppiä asetetaan *int unsigned*, ja kentän arvo kasvaa automaattisesti aina, kun uusi vastaus luodaan Vastaukset-tiluun. Tämä kenttä ei voi olla tyhjä.

Seuraavaksi luodaan tunnusno-kenttä, ja sen tietotyyppiä asetetaan *smallint unsigned*. Tämäkään kenttä ei saa olla tyhjä. Sen jälkeen luodaan projektinumeroa varten kenttä projnro tietotyyppiä *varchar(10)*. Myöskään tämä kenttä ei saa olla tyhjä.

Lopuksi luodaan vielä kysymysno-, arvosana- sekä vastauspvm-kentät. Kysymysno-kenttä määritellään *varchar(5)*-tyyppiä ja arvosana *tinyint*-tyyppiä. Kumpikaan näistä ei saa olla tyhjä. Vastauspvm-kentän tietotyyppi on *date*-tyyppiä, ja kenttä ei saa olla tyhjä.

Kenttien luonnin jälkeen luodaan relaatiot Tunnukset- ja Kysymykset-tiluihin. Relatiot ovat muodoltaan monen suhde yhteen. Vastaukset-tilun tunnusno-kenttä toimii viiteavaimena, ja se viittaa Tunnukset-tilun tunnusno-kenttään. Toinen viiteavain on kysymysno-kenttä, joka viittaa Kysymykset-tilun kysymysno-kenttään.

3.3.4 Vastaukset_text

```
CREATE TABLE vastaukset_text (
    vastausID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    tunnusno SMALLINT UNSIGNED NOT NULL,
    projnro VARCHAR(5) NOT NULL,
    kategoria VARCHAR(20) NOT NULL,
    teksti TEXT NOT NULL,
    vastauspvm DATE NOT NULL,
    CONSTRAINT vastaukset_FK_A
    FOREIGN KEY (tunnusno)
    REFERENCES tunnuksset (tunnusno)
);
```

Vastaukset_text-taulu on alkuosaltaan ihan samanlainen kuin Vastaukset-taulu. VastausID-, tunnusno- ja projnro-kenttien tietotyypit ovat samat kuin Vastaukset-taulussa, mutta sen jälkeen tulee eroavaisuuksia.

Edellisten kenttien jälkeen luodaan kategoria-kenttä tietotyypillä *varchar(20)*. Tämän jälkeen luodaan teksti-kenttä, jonka tietotyypiksi asetetaan *text*. Viimeinen kenttä Vastaukset_text-taulussa on vastauspvm, joka on tietotyypiltään muotoa *date*. Mitkään kentät eivät tässäkin taulussa saa olla tyhjiä.

Kenttien luonnin jälkeen tehdään vielä monen suhde yhteen -relaatio. Tämä tapahtuu siten, että tunnusno-kentästä tehdään viiteavain, joka viittaa Tunnukset-taulun tunnusno-kenttään. /6/

4 WEB-LOMAKKEET

4.1 Lomakkeiden suunnittelu

Tietokannan taulujen suunnittelun ja toteuttamisen jälkeen tuli eteen WWW-sivujen suunnittelu. Aikaisemmin koko järjestelmää pohtiessani olin tullut siihen päätökseen, että asiakaspalaute-WWW-sivu täytyy tehdä helpoksi ja mukavaksi asiakasta varten. Myös hallintasivut täytyisi luoda yritystä varten käytettävyyden takia.

Koska päätin, että asiakaspalautesivun pitää myös olla turvallinen, täytyy esim. hakkeiden ja muiden asiaan kuulumattomien henkilöiden pääsy estää palautesivulle. Tämä yhdistettynä siihen, että sivuille pääsy pitää asiakkaalle olla kuitenkin helppo, tulin johdopäätökseen, että asiakkaalle lähetettäisiin sähköpostilla viesti. Sähköpostiviesti sisältäisi ilmoituksen, mitä yritys asiakkaan toivoisi tekevän, sekä linkin asiakaspalautelomakkeelle. Linkki koostuisi URL-osoitteesta, jonka mukana olisi salattu tunnusosa. Salattua osaa linkistä verrattaisiin sivulle tultaessa tietokannassa oleviin Tunnuksettaulun tunnuksiin. Täten välttyään siltä ratkaisulta, että asiakas joutuisi sivuille tultaessa kirjautumaan käyttäjätunnuksella ja salasanalla palautelomakkeelle, mikä olisi asiakkaan kannalta turhauttavaa.

4.1.1 Hallintalomakkeet

Koska päädyin ratkaisuun, että palautejärjestelmä toteutettaisiin WEB-pohjaisena, tarvitaan myös selkeät hallintasivut nettiin. Hallintasivuihin kuuluisi tunnustenluontilomake asiakkaita varten, ei-vastanneiden tunnusten poistolomake, uuden kysymyksen lisäyslomake sekä hallintasivujen käyttäjän salasanavaihtoa varten oleva lomake.

4.1.1.1 Tunnustenluontilomake

Tunnustenluontilomakkeella olisi alasvetovalikko, mihin muodostetaan lista Raumaster Paperin projektinnumeroista. Projektinumeron avulla tulisi myös asiakkaan nimi omaan

kenttäänsä. Nämä tiedot tulisivat Raumaster Paperin omista tietokannoista. Nimikenttä olisi vain luku -muotoa.

Projektinumerokentän sekä asiakkaan nimen jälkeen olisi vapaasti täytettävänä kaksi tekstikenttää: nimi ja sähköposti. Näiden tarve perustuu siihen, että asiakkaalla voi olla tarvetta antaa palaute eri henkilöiden kautta, eli näin voidaan lähettää asiakaspalautepyyntö tietyille henkilöille, tiettyihin sähköpostiosoitteisiin.

Viimeinen lomakkeen kenttä olisi päivämääräkenttä, mikä olisi valmiiksi täytetty sen hetkisen päivämäärän mukaan, ja sitä ei voisi muokata.

Lopuksi hallintasivun käyttäjälle tulostettaisiin lähetysnappi, jota painettaessa tietokantaan luotaisiin tunnukset valitulle asiakkaan henkilölle ja samalla henkilölle, lähetettäisiin sähköpostilla linkki asiakaspalautesivulle.

4.1.1.2 Ei-vastanneiden tunnusten poistolomake

Poistolomakkeella käyttäjälle tulostettaisiin tietokannasta Tunnukset-taulusta luettelo niistä asiakkaiden tunnuksista, joilla palautetta ei vielä ole annettu. Käyttäjällä olisi mahdollisuus poistaa näitä tunnuksia painamalla Poista-painiketta, joka tulostettaisiin jokaisen ei-vastanneen tunnuksen perään.

4.1.1.3 Uuden kysymyksen lisäyslomake

Jotta järjestelmä olisi dynaaminen, kysymysten lisäyslomakkeella on mahdollista lisätä uusi kysymys Kysymykset-tauluun.

Käyttäjälle tulostettaisiin alasvetovalikolla kaikki Kysymykset-taulussa olevat kategoriat, jotta hän voisi valita, mihin kategoriaan hän aikoo uuden kysymyksen lisätä. Kysymyksen numerosta ei käyttäjän tarvitsisi huolehtia, vaan koodi hoitaisi sen asian tietokannan tietojen perusteella.

Näin ollen käyttäjälle ei tulostettaisi alavetovalikon lisäksi muita kenttiä kuin yksi tekstikenttä kysymystä varten sekä lähetyspainike, jolla kysymys tietokantaan lisättäisiin.

4.1.1.4 Salasananvaihtolomake

Koska nykyään on puhuttu paljon tietoturva-asioista, päätin ottaa sen myös huomioon tässä opinnäytetyössä. Päätin, että hallintasivuilla käyttäjä voisi myös vaihtaa salasanan, millä hallintasivuille pääsee sisään.

Tällä lomakkeella käyttäjälle tulostettaisiin kolme tekstikenttää ja painike, jolla salasanan voisi vaihtaa. Tekstikentät olisivat: vanha salasana, uusi salasana ja uusi salasana uudelleen.

4.1.2 Asiakaspalautelomake

Palautesivulle päästäessä asiakkaalle tulostettaisiin tekstikenttiin asiakkaan tiedot, projektinnumero sekä päivämäärä. Nämä kentät olisivat vain lukumuotoa ja kenttien tiedot tulisivat tietokannasta. Kenttien tiedot menisivät numerovastausten mukana Vastaukset-tauluun tietokantaan.

Asiakastietojen jälkeen tulostettaisiin tietokannan Kysymykset-taulun kaikki kysymykset näytölle kategorioittain numerojärjestykseen. Jokaisen kysymyksen perään muodostettaisiin kuusi radiopainiketta. Ensimmäiset viisi painiketta olisivat arvosanoja 1–5 varten, ja kuudes olisi ”ei mielipidettä” -painike. Tämä viimeinen valinta olisi oletuksena valittuna.

Jokaisen tulostetun kategorian kysymysten perään muodostettaisiin myös tekstikenttä asiakkaan vapaata kommenttia varten. Täten asiakas voisi antaa numeroarvosanojen lisäksi myös vapaamuotoisen kirjallisen palautteen jokaisesta aiheesta. Nämä tekstikenttien tekstit tallennettaisiin Vastaukset_text-tauluun tietokantaan, ja arvosanat menisivät Vastaukset-tauluun.

4.2 Lomakkeiden toteutus

Tietoturvan takia ohjelmoimissani PHP-lomakkeissa käytetään aina käyttäjän vapaa-
muotoisen tekstisyötteen tarkistamiseen PHP-funktioita, millä estetään esimerkiksi
käyttäjän oman PHP-koodin tai HTML-koodin ajaminen. Tämä tarkoittaa sitä, että käyt-
täjän syötteestä karsitaan kaikki koodeihin liittyvät elementit pois ja jätetään pelkkä
teksti jäljelle.

4.2.1 Hallintalomakkeet

Jokaisen hallintalomakkeisiin kuuluvan PHP-tiedoston alku on sama kaikilla. Koska
kyse on hallintasivuista, on järkevää muodostaa täten sessio eli istunto. Istunto luodaan
koodilla `session_start()`. Istunnon tarkoitus on ylläpitää tietoa, onko käyttäjä kirjautunut
järjestelmään sisään eli onko hänellä oikeus käyttää hallintasivuston lomakkeita. Jos
käyttäjä yrittää päästä muihin kuin sisäänkirjautumislomakkeisiin käsiksi ilman hyväk-
syttyä sisään kirjautumista, hänet ohjataan takaisin sisäänkirjautumislomakkeelle. Istun-
to päättyy joko sillä, että käyttäjä sulkee selaimen tai jos hän kirjautuu ulos hallin-
tasivuilta siellä olevan linkin avulla.

4.2.1.1 login.php

Sisäänkirjautumislomakkeena toimii login.php. Tälle sivulle tultaessa käyttäjälle tulos-
tetaan kaksi tekstikenttää, käyttäjänimi ja salasana, ja sisäänkirjautumisnappi.

Admin login

Please type username and password.

Username:

Password:

Kuva 2. Järjestelmän ylläpitäjän kirjautumislomake.

Kun käyttäjä on täyttänyt kentät ja yrittää kirjautua sisään, tiedot lähetetään post-
metodilla login.php sivulle itselleen tarkastettavaksi. Tarkistuksessa katsotaan, ettei
käyttäjä lähettänyt tyhjiä kenttiä. Tästä ilmoitetaan käyttäjälle (ks. kuva 2).

Jos käyttäjä täyttää molemmat kentät, tehdään käyttäjän syötteistä hash-salauskoodi tietyllä itse kehittämälläni rakenteella. Hash-koodi muodostetaan PHP:ssa `$hashi = md5(syöte);` käskyllä. Tietoturvasyistä en sitä salauksen muodostavaa syötettä voi tässä yhteydessä mainita.

Hash-koodin luonnin jälkeen otetaan yhteys tietokannan Tunnukset-tauluun ja tarkistetaan, täsmääkö hash-koodi siellä olevaan pääkäyttäjän tunnuksen hashiin. Jos hash-koodi täsmää, ohjataan käyttäjä `addaccount.php`-sivulle. Muussa tapauksessa kerrotaan virheenä käyttäjälle, ettei tunnus tai salasana olleet oikein.

4.2.1.2 `addaccount.php`

Kun sisäänkirjautumissivulta on onnistuneesti kirjaututtu sisään, käyttäjä ohjataan `addaccount.php`-sivulle eli tunnustenluontilomakkeelle, niin kuin yllä jo mainitsin. Käyttäjän eteen avautuu lomake (ks. kuva 3), joka sisältää seuraavat kentät: project number, company name, person name, person e-mail ja date.

Project number -kenttä muodostuu alavetovalikosta, mihin muodostetaan Raumaster Paperin omista tietokannoista projektinumerolistaus kaikista projekteista. Kun käyttäjä valitsee jonkin projektinumeron valikosta, muodostetaan sitä projektia vastaava asiakkaan nimi alapuolella olevaan company name -kenttään Raumaster Paperin oman tietokannan mukaan. Nimikenttä on vain luku -muotoa.

Näiden kahden kentän jälkeen on vuorossa käyttäjälle täytettäväksi tarkoitetut kentät. Person name -kenttään käyttäjä kirjoittaa sen henkilön nimen, kenelle hän haluaa asiakkaan yrityksessä asiakaspalautelomakkeen lähetettävän. Nimikentän alapuolelle tulee tämän tietyn henkilön sähköpostiosoite. Viimeinen kenttä on päivämääräkenttä tunnusten luontia varten, jotta tiedetään, milloin tunnukset on asiakkaalle luotu. Tämä kenttä on vain luku -muotoa.

Administrator: [Add account] - [Remove accounts] - [Add question] - [Change password]	[Logout]
---	--------------------------

Add new account

Fill in this form to provide a customer access to the answer request form.

Project number:

Company name:

Person name: **Please type person name!**

Person e-mail: **Please type email address!**

Date:

Kuva 3. Uuden tunnuksen luontilomake.

Kun käyttäjä lähettää tiedot ”Create access” -napilla, tiedot lähetetään addaccount.php-tiedostolle itselleen takaisin post-metodilla tarkistuksia varten. Aluksi tarkistetaan, ettei käyttäjä jättänyt täyttämättä kenttiä. Jos tyhjiä kenttiä yritettiin lähettää, annetaan siitä asiaan kuuluva virheilmoitus (ks. kuva 3). Jos taas tiedot täytettiin oikein, otetaan yhteys tietokantaan ja tarkistetaan, ettei samaa tunnusta ole ennestään olemassa. Tarkistuksessa verrataan, ettei samalla projektinumerolla ole ennestään samaa sähköpostiosoitetta. Tämä tarkoittaa sitä, että samalle projektille voidaan lähettää asiakaspalautekysely usealle henkilölle samaan yritykseen, jos vaikka toinen henkilö yrityksessä tietää tietyn osa-alueen asiat ja toinen taas toisen. Näin saadaan koko palaute annettua asiakkaan puolelta.

Jos tunnus oli jo olemassa, annetaan käyttäjälle virheilmoitus, että tunnus on jo olemassa. Muussa tapauksessa luodaan sähköposti asiakkaalle. Sähköpostiviestissä on ilmoituksen lisäksi mukana aikaisemmin mainitsemani HTTP-osoite, joka koostuu URL-osasta, mihin liitetään myös hash-koodi, jonka avulla asiakas pääsee suoraan käsiksi asiakaspalautekyselylomakkeeseen. HTTP-osoite olisi täten muotoa: index.php?u=[tähän hash-koodi].

4.2.1.3 removeaccount.php

Jos käyttäjä on kirjautuneena sisään, hänellä on pääsy myös tunnusten poistoon. Koska tietokannan Tunnukset-taulua pidetään samalla myös lokina, ei sellaisia tunnuksia voi poistaa, jotka ovat jo vastauksen antaneet.

Sivulle tultaessa otetaan yhteys tietokannan Tunnukset-tauluun ja tulostetaan vain ne tunnukset, jotka eivät vielä ole vastausta antaneet (ks. kuva 4). Näin voidaan poistaa vahingossa väärin kirjoitetut tunnukset sekä myös sellaiset tunnukset, joista tiedetään, etteivät niiden tunnusten omistajat aio vastata asiakaspalautekyselyyn. Mikäli tietokannasta ei löydy yhtään ei-vastannutta tunnusta, tästä ilmoitetaan henkilölle.

Administrator: [Add account] - [Remove accounts] - [Add question] - [Change password]						[Logout]
Remove non-answered accounts						
Projectno	Company	Person name	Person email	Create date	Answered?	
AXXXX	Testi Yritys	Testi Henkilö	testihenkilo@testiryitys.fi	2008-03-21	no	Remove
AXXXX	Testi Yritys	Testi Henkilö2	testihenkilo2@testiryitys.fi	2008-03-23	no	Remove

Kuva 4. Ei-vastanneiden tunnuksien poistolomake.

Jos käyttäjä painaa haluamansa rivin "Remove" -nappia, lähetetään tieto removeaccount.php-tiedostolle itselleen post-metodilla. PHP-koodissa on kohta, jossa tarkastetaan onko poistopainiketta painettu. Jos oli painettu, otetaan yhteys tunnukset kantaan ja poistetaan valitun asiakkaan tunnus tietokannasta.

4.2.1.4 addquestion.php

Koska asiakaspalautealomakkeen koodi ei ole lukittu tiettyyn määrään kysymyksiä, on uusia kysymyksiä varten oma lisäyslomake.

Aluksi otetaan yhteys tietokannan Kysymykset-tauluun ja haetaan vastaus siihen, mitä kategorioita Kysymykset-taulusta löytyy. Tämän tiedon perusteella käyttäjälle tulostetaan alasvetovalikkoon lista kategorioista, joista hän voi valita sen kategorian, johon hän haluaa uuden kysymyksen lisäävän.

Tämän jälkeen tulostetaan tekstikenttä käyttäjän uutta kysymystekstiä varten. Kun käyttäjä on kirjoittanut kysymyksen, hän painaa hänelle tulostettua lisäyspainiketta, jolla itse lisäys tapahtuu.

Administrator: [Add account] - [Remove accounts] - [Add question] - [Change password]	[Logout]
---	--------------------------

Add new question to the database

Please type a question!

Category:

New question:

Kuva 5. Uuden kysymyksen lisäyslomake.

Kun käyttäjä painaa lisäyspainiketta, tarkastetaan ensimmäisenä se, ettei käyttäjä yritä lähettää tyhjää kenttää. Jos näin kävi, tästä ilmoitetaan käyttäjälle asianmukaisella tavalla (ks. kuva 5). Jos taas tarkistuksen ohi päästiin, otetaan yhteys tietokantaan ja tutkitaan, mikä on viimeisimmän kysymyksen kysymysnumero käyttäjän valitsemaan kategoriaan. Tämä tehdään siksi, että tiedetään, millä numerolla käyttäjän uusi kysymys voidaan kantaan lisätä.

Kannasta haettu viimeisin kysymysnumero pitää purkaa osiin, koska se on muotoa ”1.1”. Osiin purkamisen syy perustuu siihen, että on helpompi lisätä kategoriaan lisättävän kysymyksen numeroa, kun siitä erottaa itse kysymyksen numeron kategorianumerosta. Osiin jakaminen tapahtuu `split()`; käskyllä ja jakaminen tapahtuu pisteen kohdalta kahteen eri muuttujaan. Piste vasen puoli kuvaa kysymysnumerossa kategorian numeroa, ja pisteen oikea puoli kuvaa kysymyksen numeroa kyseisessä kategoriassa. Tällä tavalla voidaan kysymysnumeroa lisätä yhdellä ja saadaan oikea kysymysnumero uudelle lisättävälle kysymykselle.

Kun uusi kysymysnumero on saatu selville, voidaan kysymys lisätä Kysymykset-tauluun ottamalla yhteys tietokantaan. Kysymyksen onnistuneen lisäyksen jälkeen käyttäjälle informoidaan asiasta asianmukaisella tavalla linkin kera, josta pääsee takaisin `addquestion.php`-sivulle.

4.2.1.5 `changepasswd.php`

Tein hallintosivuille mahdollisuuden vaihtaa pääkäyttäjän salasanan, koska nykyään on yleistä, että järjestelmissä vaihdetaan salasanat tietyn väliajoin, jotta tietoturvasuus paranisi.

Tälle salasananvaihtosivulle tullessa käyttäjälle tulostetaan lomake, missä on kolme kenttää: vanha salasana, uusi salasana ja uusi salasana uudelleen.

Administrator: [\[Add account\]](#) - [\[Remove accounts\]](#) - [\[Add question\]](#) - [\[Change password\]](#) [\[Logout\]](#)

Change admin password

Old password doesn't match!

Old password:

New password:

New password again:

Kuva 6. Järjestelmän ylläpitäjän salasananvaihtolomake.

Jos käyttäjä painaa ”Change password” -painiketta, lähetetään kenttien tiedot change-passwd.php-tiedostolle itselleen tarkistuksia varten post-metodilla. Mikäli käyttäjä ei täyttänyt kaikkia kenttiä, annetaan siitä asiaan kuuluva ilmoitus. Jos kaikki kentät syötettiin täytettyinä, otetaan yhteys tietokannan Tunnukset-tauluun ja tarkistetaan ensimmäisenä se asia, syöttikö käyttäjä vanhan salasanan oikein. Jos salasana oli väärin, annetaan siitä asianmukainen ilmoitus käyttäjälle (ks. kuva 6). Mikäli vanha salasana oli oikein, tehdään vielä toinen tarkistus, että käyttäjä syötti uuden salasanan molempiin kenttiin samalla tavalla, jotta vältetään kirjoitusvirheitä.

Kun kaikki tiedot on oikein syötetty ja syötteet menneet tarkistuksista läpi, luodaan uudesta salasanasta tietyllä itse kehittämälläni rakenteella hash-koodi. Koodi tallennetaan Tunnukset-tauluun vanhan salasanahashin päälle. Näin salasana on vaihdettu, ja käyttäjälle ilmoitetaan onnistuneesta vaihdoksesta.

4.2.1.6 logout.php

Tämä PHP-sivu (ks. kuva 7) on uloskirjautumista varten. Sivulle tullessa tarkistetaan, onko istunto auki, ja jos on, niin suljetaan istunto ja sen istuntomuuttujat. Tämän jälkeen käyttäjä ohjataan takaisin sisään kirjautumissivulle.

Admin login

You have been succesfully logged off.

Username:

Password:

Kuva 7. Järjestelmän ylläpitäjä on uloskirjautunut järjestelmästä.

4.2.2 Asiakaspalautelomake

4.2.2.1 index.php

Tämä sivu toimii asiakaspalautelomakkeena (ks. kuva 8). Sivulle tultaessa tarkastetaan, ettei sivulle tulla ilman hash-koodia tai väärällä sellaisella. Tässä tapauksessa henkilö ohjataan noaccess.php-sivulle, ja täten pääsy itse asiakaspalautelomakkeelle estetään.

Jos http-osoitteessa on hash-koodi mukana, otetaan yhteys tietokannan Tunnukset-tauluun ja muodostetaan taulussa olevan rivin tiedoista tietyllä itse kehittämälläni rakenteella hash-koodi, jota verrataan linkissä saatuun koodiin. Jos linkissä saatu hash-koodi täsmää tietokannasta luotuun hash-koodiin, otetaan muuttujiin talteen sen kyseisen taulun rivin tiedot. Tämä tarkoittaa sitä, että oikea asiakas löytyi, ja täten henkilön pääsy sivulle myönnetään.

Kun pääsy on myönnetty henkilölle, tulostetaan tietokannasta saadut asiakkaan tiedot seuraaviin kenttiin: project number, company name, person name, person e-mail ja date. Date-kentän päivämäärä on sen hetkinen päivämäärä. Kentät ovat vain luku -muotoa.

Seuraavaksi asiakkaalle tulostetaan käyttöohje asiakaspalautelomakkeen täyttöö varten, jonka jälkeen otetaan yhteys tietokannan Kysymykset-tauluun. Tietokannasta selvitetään ensimmäiseksi mitä kategorioita siellä on, minkä perusteella voidaan tulostaa asiakkaalle kaikki Kysymykset-taulun kysymykset kategorioittain numerojärjestykseen.

Jokaisen kysymyksen perään luodaan radiopainikkeet arvosanoja 1–5 varten, ja kuudes painike on sitä varten, että henkilöllä ei ole mielipidettä asiaan. Se tarkoittaa käytännös-

sä sitä, ettei sen kysymyksen vastausta oteta huomioon ollenkaan, kun vastauksia sijoitetaan tietokantaan. Tämä valinta on oletuksena valittuna.

Kun kaikki kategorian kysymykset on tulostettu, lisätään vielä tekstikenttä valinnaista sanallista palautetta varten. Tämän tekstikentän vastaus menee omaan tauluunsa tietokannassa.

Give feedback

Project number:

Company name:

Person name:

Person e-mail:

Date:

Please tick the appropriate column with your mouse according to:

- 1 = very unhappy
- 3 = neutral
- 5 = very satisfied
- = no opinion

SERVICE		1	2	3	4	5	-
1.1	The clarity and comparability of the quotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
1.2	Delivery speed/time	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
1.3	Delivery accuracy to stated delivery time	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
1.4	Delivery time flexibility	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
1.5	Did the delivery meet the expectations provided by the quotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
1.6	Ability to handle situations related to the project	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<hr/> <p><i>Comments:</i></p> <div style="border: 1px solid black; height: 60px; width: 100%;"></div>							

Kuva 8. Asiakaspalautelomake tulostettuna asiakkaalle.

Kun asiakas painaa sivun loppuun luotua ”Submit” -painiketta, lähetetään lomakkeen tiedot post-metodilla index.php-tiedostoon itselleen tarkistusta varten. Jos asiakas yrittää lähettää tyhjää vastausta, eli asiakas ei valitse yhtään arvosanaa mihinkään kysymykseen, ilmoitetaan siitä asiakkaalle eikä jatketa eteenpäin, ennen kuin edes yksi arvosana annetaan.

Hyväksytyyn napinpainalluksen jälkeen alkavat varsinaiset toimenpiteet. Aluksi eritellään asiakkaan vastauksista erilleen tekstikenttien vastaukset ja kysymyksien arvosanavastaukset. Tämä siksi, että osataan laittaa oikeat vastaukset oikeisiin tauluihin tietokannassa. Tämän jälkeen otetaan yhteys tietokantaan ja haetaan asiakkaan tunnusnumero, koska tätä tietoa ei ole missään kentissä ollut aikaisemmin, koska ei ole tarvetta näyttää asiakkaalle liikaa informaatiota jo tietoturvasyistäkin. Tunnusnumeron lisäksi joudutaan vielä hakemaan uudestaan kategoriat muuttujiin, koska nämä tiedot eivät välity lomakkeen lähetyksessä.

Nyt kaikki oleelliset tiedot ovat kasassa ja voidaan ottaa yhteys tietokannan Vastaukset-tauluun ja sijoittaa kaikki asiakkaan antamat numeroarvosanat tietokantaan. Myös valinnaiset tekstivastaukset sijoitetaan tämän jälkeen omaan Vastaukset_text-tauluun.

Kun kaikki vastaukset on sijoitettu tietokantaan, pitää vielä ottaa yhteys Tunnukset-tauluun ja päivittää asiakkaan tunnuksen tiedot, että tämä on vastauksen suorittanut onnistuneesti ja minä päivänä vastaus tehtiin. Kyseisten toimenpiteiden jälkeen asiakkaalle lähetetyllä linkillä ei enää voi päästä asiakaspalautelomakkeelle sisään. Asiakas ohjataan tämän päivityksen jälkeen sent.php-lomakkeelle ja kiitetään onnistuneesta vastauksesta.

4.2.2.2 noaccess.php

Henkilö ohjataan aina noaccess.php-sivulle, jos henkilöllä ei ole käyttöoikeutta asiakaspalautejärjestelmään. Sivulla ilmoitetaan asiasta henkilölle (ks. kuva 9).

Access Denied!

You don't have permissions to access this site.

Kuva 9. Ilmoitus käyttäjälle jos hänellä ei ole oikeuksia järjestelmään.

4.2.2.3 sent.php

Sent.php-sivulle tullaan asiakaspalautelomakkeen kautta siinä tapauksessa, kun asiakkaan antamat vastaukset on onnistuneesti lisätty tietokantaan. Asiakkaalle tulostetaan ilmoitus asiasta (ks. kuva 10).

Done!

Your feedback has been sent to Raumaster Paper succesfully!

Thank you!

Kuva 10. Asiakkaalle tulostettu ilmoitus onnistuneesta palautteenannosta.

5 TIETOKANNAN HYÖDYNTÄMINEN

Raumaster Paperilla on jo olemassa valmiit työkalut tietokantojen datan hyödyntämiseen, joten niitä päätettiin hyödyntää. Hyödyntämisellä tarkoitetaan esimerkiksi keskiarvojen laskemista tai graafisten käyrien tekemistä tietyillä arvoilla. Tässä työssä hyvä keskiarvolaskujen kohde voisi olla esimerkiksi jokaisen tietokannassa olevan Vastaukset-taulun kategorian kysymyksiin vastattujen arvosanojen laskeminen.

Määrittelemäni tietokannat lisätään Raumaster Paperilla käytössä olevaan eBic-nimiseen ohjelmistoon, jotta ohjelma voi vastaustauluissa olevaa dataa hyödyntää ja käsitellä. Taulujen lisäys eBic:iin tapahtuu SQL-komentoja käyttäen, eli samoja komentoja käyttäen, kuin tässä työssä itse käytin luomaan tauluja tietokantaan.

EBic on tietyllä tavalla kuin tietovarasto (DW Data Warehouse), joka pohjautuu Oracle-tietokannan pohjalle. Kun varastoon on saatu tietoja muista tietokannoista, voidaan Oracleen liittyvillä Discover BI (Business Intelligence) -työkaluilla käsitellä tallennettua dataa juuri haluttavalla tavalla.

6 YHTEENVETO

Työtä aloittaessani oli hyvin hahmotettuna mielessäni ja paperilla, minkälainen järjestelmän pitäisi olla. Suunnittelussa olivat muutama tietokantaan sijoitettava taulu ja taulujen väliset yhteydet, nettiin tuleva asiakaspalautesivu, Raumaster Paperia varten nettiin tulevat hallintasivut sekä palautteen informaatiota tulkitseva ohjelmisto. Kun ajatuksia alkoi työstää ohjelmoimalla koodiksi, syntyi mutkia matkaan.

Alkuperäiset tietokannat piti suunnitella uudelleen, kun huomasin, että järjestelmä vaatii tiettyjä toiminnallisuuksia toimiakseen järkevästi ja turvallisesti. Alkuperäisessä suunnitelmassani oli yksi taulu enemmän, kuin mitä tässä dokumentissa kuvailin. Yhdistin sen lopulta Tunnukset-tauluun, ja sen jälkeen työnteke nopeutui huomattavasti, kun sai muutoksen aiheuttaman koodinmuokkauksen tehdyksi.

Kuten jo mainitsin, työ tuli tehtyä suhteellisen kivuttomasti valmiiksi juuri hyvän suunnittelun vuoksi. Ei ollut kuin yksi isompi ongelma, joka oli tämä yksi ylimääräinen taulu suunnitteluvaiheessa. Tässä työssä, tällä aikataululla, tuli huomattua se usein kuulemani fraasi: *”Hyvin suunniteltu on puoleksi tehty”*. Nyt kävi juuri näin, mistä olen tyytyväinen.

Järjestelmästä tuli suunnitelmani mukaan dynaaminen. Se on yrityksen puolesta helposti hallittava, kiitos turvallisten hallintasivujen. Asiakkaan on taas miellyttävää antaa asiakaspalaute, kun ei tarvitse kirjautua mihinkään järjestelmään sisälle käyttäjätunnuksilla ja salasanoilla, vaan ainoastaan klikata sähköpostiviestistä linkkiä. Koska työssä otettiin huomioon myös ulkopuolisten henkilöiden pääsyn estäminen järjestelmään ja salausalgoritmina käytettiin md5:tä, on järjestelmä Raumaster Paperia ajatellen turvallinen ja luotettava.

LÄHTEET

- 1: Haapakoski M. & Pajukoski P. 2007. Kehittämisesitysraportti. [Verkkodokumentti, viitattu 31.3.2008] Saatavissa: <http://www.haapavesi.fi/lataa.php/1850/Kehitt%C3%A4misraportti,%20Haapakoski%20&%20Pajukoski%202007.pdf>
- 2: Heinisuo, R. 2004. PHP ja MySQL. Helsinki: Talentum Media Oy
- 3: Meloni, J. 2003. MySQL Trainer Kit. Helsinki: IT-Press
- 4: Laukkanen, P. Yritysesittelyn materiaalia. [Sähköpostiviesti]. Vastaanottaja: hermanni.piltti@raumaster.fi. Lähetetty 31.3.2008 klo 21.44. [viitattu 1.4.2008].
- 5: MySQL-Tietotyypit. [Verkkodokumentti, viitattu 23.3.2008] Saatavissa: <http://dev.mysql.com/doc/refman/5.0/en/data-types.html>
- 6: SQL-Info-Taulujen luontikoodiesimerkkejä. [Verkkodokumentti, viitattu 23.3.2008] Saatavissa: <http://sql-info.de/mysql/examples/CREATE-TABLE-examples.html>