



AUTOVUOKRAAMON VARAUSJÄRJESTELMÄ

Opinnäytetyö

Teemu Tarvainen

Hyvinvointiteknologian koulutusohjelma

Hyväksytty ____ . ____ . ____ _____

SAVONIA-AMMATTIKORKEAKOULU TEKNIikka KUOPIO

Koulutusohjelma

Hyvinvointiteknologian koulutusohjelma

Tekijä

Teemu Tarvainen

Työn nimi

Autovuokraamon varausjärjestelmä

Työn laji

Opinnäytetyö

Päiväys

27.5.2010

Sivumäärä

44

Työn valvoja

Lehtori Jussi Koistinen

Yrityksen yhdyshenkilö

Laura Tarvainen

Yritys

Autovuokraamo Matka-Rent Oy

Tiivistelmä

Tämän opinnäytetyön aiheena oli suunnitella ja toteuttaa varausjärjestelmä Autovuokraamo Matka-Rent Oy:lle. Yritys sijaitsee Jyväskylässä ja sen liiketoiminta koostuu ajoneuvojen, peräkärryjen ja erilaisten varusteiden vuokraustoiminnasta. Aiemmin käytetty menetelmä varausten ylläpitoon oli ”kynä ja paperi” ja se oli koettu liian vaikeasti hallittavaksi. Sovelluksen ominaisuudet suunniteltiin yrityksen päivittäisiin tarpeisiin.

Sovellus toteutettiin PHP-ohjelmointikieltä ja jQuery JavaScript -kirjastoa käyttäen. Ulkoasu rakennettiin XHTML-kuvauskielellä, jonka muotoilu määriteltiin CSS-tyylitiedostoilla. Sovelluksessa käsiteltävät tiedot tallennetaan MySQL-relaatio-tietokantaan.

Sovellus on suunniteltu julkaistavaksi yleisessä verkossa ja sitä ohjelmoitaessa on otettu huomioon tunnetuimmat web-sovellusten haavoittuvuudet. Suojautuminen haavoittuvuuksilta on myös esitelty työssä. Työn tuloksena sovelluksen ensimmäinen versio saatiin valmiiksi ja otettiin koekäyttöön yrityksessä.

Avainsanat

PHP, MySQL, www

Luottamuksellisuus

julkinen

SAVONIA UNIVERSITY OF APPLIED SCIENCES

Degree Programme

Welfare Technology

Author

Teemu Tarvainen

Title of Project

Reservation System for Car Rental Agency

Type of Project

Final Project

Date

27 May 2010

Pages

44

Academic Supervisor

Mr Jussi Koistinen, Lecturer

Company Supervisor

Ms Laura Tarvainen

Company

Autovuokraamo Matka-Rent Oy

Abstract

The aim of this thesis was to design and implement a reservation system for Autovuokraamo Matka-Rent Oy. The company is situated in Jyväskylä and its business consists of rental operations for vehicles, trailers and different kind of accessories. The previous method for reservation maintenance was a “pen and paper” system and that was experienced difficult to manage. The features of the program were designed to meet the needs of daily work in the company.

The program was implemented with PHP and jQuery JavaScript library. Program appearance was built with the XHTML description language and CSS style sheets. The data handled with the program was stored in a MySQL relational database.

The program was designed to be published on a public network and when programming it, most common security vulnerabilities in web programs were taken into account. Also security protection for vulnerabilities was studied. The outcome of this thesis was that the first release of the program was introduced for test-use.

Keywords

PHP, MySQL, www

Confidentiality

public

ALKUSANAT

Tämä opinnäytetyö on tehty Savonia-ammattikorkeakoulussa keväällä 2010 Autovuokraamo Matka-Rent Oy:lle. Työn tarkoituksena oli suunnitella ja toteuttaa web-sovellus vuokraamon asiakkaiden, ajoneuvojen ja varausten hallintaan.

Kiitän Autovuokraamo Matka-Rent Oy:n Laura Tarvaista mahdollisuudesta tehdä opinnäytetyö yritykseen ja lehtori Jussi Koistista opinnäytetyöni ohjaamisesta.

Jyväskylässä 27.5.2010

Teemu Tarvainen

SISÄLLYS

1 JOHDANTO.....	7
1.1 TYÖN TAUSTA	7
1.2 TYÖN TAVOITTEET	7
2 TEKNIIKAT JA YMPÄRISTÖ.....	9
2.1 XHTML.....	9
2.2 JAVASCRIPT	9
2.3 CSS	9
2.4 PHP.....	10
2.5 MYSQL.....	10
2.6 AJAX.....	10
2.7 KEHITYSYMPÄRISTÖ.....	11
3 MÄÄRITTELY	12
3.1 TOIMINTOJEN MÄÄRITTELY	12
3.2 TIETOJEN MÄÄRITTELY	13
4 SOVELLUKSEN TOIMINNOT.....	16
4.1 KIRJAUTUMINEN.....	16
4.2 KALENTERI.....	16
4.3 ASIAKASHALLINTA.....	20
4.4 KÄYTTÄJIENHALLINTA	22
4.5 TUOTTEIDENHALLINTA.....	24
4.6 TILASTOT	26
5 SOVELLUKSEN TOTEUTUS.....	28
5.1 TIETOKANTA	28
5.2 SOVELLUKSEN RAKENNE.....	30
5.2.1 Hakemistorakenne.....	30
5.2.2 Luokat	31
5.2.3 Sivujen koodin rakenne	32
5.2.4 Sovelluksen JavaScript.....	33
5.3 TIETOTURVA	35
5.3.1 Cross-site scripting (XSS).....	35
5.3.2 Cross-site request forgery (CSRF).....	36
5.3.3 SQL-injektio	36
5.3.4 Istunnon kaappaus	37
5.3.5 Resurssien suojaus	37
5.4 TESTAUS.....	38
6 SOVELLUKSEN KÄYTTÖÖNOTTO.....	40
6.1 PALVELINVAATIMUKSET	40
6.2 SOVELLUKSEN SIIRTÄMINEN PALVELIMELLE.....	40
6.3 KÄYTTÖÖNOTTOTESTAUS	41
7 PÄÄTELMÄT	42
LÄHTEET	43
LIITE 1. AJONEUVON VUOKRASOPIMUS	44

LYHENTEET JA TERMIT

AJAX	Asynchronous JavaScript and XML. Web-sovelluksien kehityksessä käytetty tekniikka, jolla lisätään sivun dynaamisuutta.
CSRF	Cross-Site Request Forgery. Tietoturva-aukko, jonka avulla voidaan käyttää hyväksi sivustolle kirjautuneen henkilön selainta vahingollisten komentojen suorittamiseksi.
CSS	Cascading Style Sheets. (X)HTML-dokumenttien ulkoasun määrittelyyn käytettävä kuvauskieli
JavaScript	Oliopohjainen komentosarjakieli, jonka avulla lisätään www-sivuston dynaamisuutta.
jQuery	Ilmainen JavaScript kirjasto, jonka avulla voi helposti toteuttaa monia muuten vaikeasti toteutettavia toiminnallisuuksia.
MySQL	Avoimeen lähdekoodiin perustuva, monipuolinen ja tehokas relaatiotietokanta.
PHP	<i>PHP: Hypertext Preprocessor</i> . web-sovellusten toteuttamiseen kehitetty oliopohjainen ohjelmointikieli.
SaaS	Software as a service. Malli, jossa sovellus myydään verkon yli web-palveluna.
SQL-injektio	Tekniikka tietoturva aukon hyödyntämiseksi tietokantapohjaisiin järjestelmiin tunkeuduttaessa.
XHTML	HTML:stä kehitetty www -sivujen merkintäkieli.
XSS	Cross-Site Scripting. Hyvin yleinen tietoturva-aukko web-sovelluksissa, jonka avulla sivustolle voi lisätä sinne kuulumatonta koodia.
XAMPP	Apache, MySQL, PHP, Perl, phpMyAdmin ym. samassa helposti asennettavassa paketissa.
W3C	World Wide Web Consortium, Ylläpitää ja kehittää useita www-julkaisussa käytettäviä standardeja.

1 JOHDANTO

Tietotekniikan käyttö työnteon apuna on jo lähes joka yrityksen arkea. Laajakaistayhteyksien yleistymisen myötä myös erilaisten web-sovellusten suosio on kasvanut. Web-sovellukset voi hankkia palveluna (SAAS) tai räätälöitynä sovelluksena. Palveluna myytävissä sovelluksissa on usein mahdollisuus saada joitain ominaisuuksia mukautettua omaan käyttötarkoitukseen sopivaksi. Kuitenkin niissä saattaa olla tarpeettomia ominaisuuksia, joille ei tulevaisuudessakaan löytyisi käyttöä, tai niistä voi puuttua jotain yrityksen liiketoimintaa tukevia toimintoja. Räätälöidyt sovellukset on alusta alkaen rakennettu tilaajan tarpeisiin sopiviksi, joten niissä on kaikki tarvittavat ominaisuudet. Tällaisen sovelluksen laajentaminen tulevaisuudessa on myös tarvittaessa mahdollista.

1.1 Työn tausta

Autovuokraamo Matka-Rent Oy:ssä varaukskalenteria on pidetty aiemmin paperilla, mikä on aiheuttanut ongelmia kalustomäärän kasvettua ja toiminnan jakaannuttua kahteen eri toimipisteeseen. Käytännössä varauksen vastaanotto onnistuu vain henkilöltä, jolla on varauskirja käytössään. Muiden henkilöiden tarvitessa tietoa varaustilanteesta on täytynyt soittaa sillä hetkellä varauksia hoitaneelle henkilölle. Tämä on aiheuttanut kustannuksia työntekijöiden välisen välttämättömän puhelinliikenteen sekä puheluiden kuluttaman työajan menetyksen muodossa. Ylimääräisen työn lisäksi on toisinaan saattanut tulla päällekkäisiä varauksia. Varausjärjestelmä helpottaisi paljon päivittäistä toimintaa.

1.2 Työn tavoitteet

Aluksi työn tavoitteena oli suunnitella ja toteuttaa web-selaimella käytettävä varausjärjestelmä, joka olisi sisältänyt toimivalle varausjärjestelmälle välttämättömät toiminnot: asiakasrekisterin, tuoterekisterin, varauksalenterin ja mahdollisuuden tulostaa vuokrasopimukset. Kuukauden työskentelyn jälkeen näytti kuitenkin siltä, että opinnäytetyö jäisi suppeaksi. Laajuuden lisäämiseksi sovellukseen mietittiin lisätoimintoja, jotka tukisivat yrityksen toimintaa. Tarvittavat toiminnot löytyivät helposti. Esimerkiksi autojen huoltotiedot oli aiemmin seurattu suoraan huoltokirjasta ja autokohtaista tuloseurantaa oli tehty erikseen taulukkolaskentaohjelmalla. Tämä oli taas aiheuttanut huoltovälien ja tulojen

seurannassa ylimääräistä työtä, joten näitä asioita helpottamaan sovellukseen lisättiin huolto- ja korjaustietojen ylläpito sekä tilastosivu autokohtaisten vuokratulo- ja huoltokuluseurannan mahdollistamiseksi. Lisäksi sovelluksen tilastosivulle lisättiin mahdollisuus nähdä sovelluksen käyttäjien tekemien varausten, luovutusten ja vastaanottojen määrät. Näillä lisäyksillä sovelluksesta saatiinkin mielekäs kokonaisuus toimintoja, jotka helpottavat yrityksen päivittäistä työskentelyä.

2 TEKNIIKAT JA YMPÄRISTÖ

Tässä luvussa esitellään lyhyesti työssä käytetyt tekniikat ja sovelluksen kehitysympäristö. Näiden tekniikoiden valintaan on vaikuttanut niiden yleisyys alalla, helppo opittavuus ja hyvä dokumentaatio. Lisäksi valintaan vaikutti se, että kaikki tekniikat voi ottaa käyttöön ilmaiseksi.

2.1 XHTML

XHTML on HTML:stä kehitetty merkintäkieli, jota käytetään www-sivujen muodostamiseen. XHTML täyttää XML:n muotoiluvaatimuksen ja siinä on HTML:ää tarkemmat muutosäännöt. Suurimpia eroja on pienien kirjaimien käyttö tageissa sekä jokaiselle tagille vaadittava sulkeva tagi. Lisäksi XHTML-dokumentissa ei saa olla ulkoasumäärittelyjä, vaan ne on hoidettava erillisillä tyylisivuilla. Tämän sovelluksen sivut on toteutettu XHTML 1.0 Strict -määrittelyä vastaavaksi. [1]

2.2 JavaScript

JavaScript on oliopohjainen ohjelmointikieli, jota kaikki yleisimmät nykyselaimet tukevat. Skriptit voivat olla osana HTML-dokumenttia tai kuten yleensä, omissa tiedostoissaan, jotka selain lataa sivun mukana. JavaScript on tulkittava kieli eli skripti suoritetaan selaimen JavaScript-tulkissa [2]. Tämän sovelluksen JavaScript-koodit ovat omissa tiedostoissaan ja XHTML-dokumentissa vain kutsutaan erillisen tiedoston sisältämiä funktioita onclick- ja onload-attribuuteissa.

2.3 CSS

Cascading Style Sheet, CSS, on W3C:n ylläpitämä määrittely tyylikielelle, joka on kehitetty helpottamaan web-sivustojen muotoilua. CSS-määrittelyjä on olemassa useita, mutta yleisesti määrittelyistä käytetään vain nimeä CSS. Määrittelyn lisääminen sivulle voidaan tehdä sivun alkuun kirjoittamalla, upottamalla suoraan (X)HTML-elementtiin tai tallentamalla erilliseen tyylitiedostoon. Erillinen tiedosto on linkitettävä (X)HTML-dokumenttiin <link>-tagilla, jolloin selain osaa näyttää sivun muotoilun mukaisesti.

Erillinen tyylitiedosto on suosittu tapa, koska se mahdollistaa saman tyylitiedoston käyttämisen usealla sivulla. [3]

2.4 PHP

PHP oli aluksi vain kokoelma web-sovellusten tekemistä helpottavia CGI -skriptejä. Ajan saatossa se on kehittynyt monipuoliseksi ohjelmointikieleksi, jolla on vahva asema dynaamisten palveluiden tuottamiseen tarkoitettujen kielten joukossa. PHP-koodia voidaan sijoittaa `www`-sivulle (X)HTML-elementtien sekaan tagien `<?php` ja `?>` sisään kirjoitettuna. PHP on tulkettava kieli, eli se suoritetaan aina palvelimella ennen sivun lähettämistä selaimelle. Tällöin (X)HTML-elementtien sekaan kirjoitetusta koodista näkyy sivuilla vain sen tuottamat tulosteet. [4, s. 12]

2.5 MySQL

MySQL on maailman suosituin avoimen lähdekoodin relaatiotietokanta. Tietokannan vahvuuksiin luetaan nopeus, luotettavuus ja helppokäyttöisyys. Lisäksi vahvuuksiin voi lukea myös laajan käyttöjärjestelmä tuen, sillä MySQL on saatavilla yli 20 alustalle. Myös ohjelmointirajapinnat ovat saatavilla yleisimmille ohjelmointikielille. Tietokanta käyttää SQL-kieltä, jonka avulla tehdään kantaan kyselyitä. [5]

2.6 AJAX

AJAX on web-sivustojen vuorovaikutteisuutta parantava tekniikka. AJAX-tekniikan perusajatuksena on sivuston osan muuttuessa ladata vain muuttunut tieto palvelimelta ja päivittää tiedot selaimessa näkyvään sivuun ilman koko sivun uudelleenlatausta. AJAX käyttää hyväkseen XMLHttpRequest-objektia, jonka avulla HTTP-pyyntö lähetetään palvelimelle ja vastaanotetaan palvelimen vastaus. Selainten tuki XMLHttpRequest-objektille on vaihtelevaa, mikä on otettava huomioon AJAX-sovellusta toteutettaessa. Tässä työssä on käytetty ilmaista jQuery-JavaScript-kirjastoa, joka sisältää AJAX:n lisäksi myös paljon muita ominaisuuksia. Tällä tavoin selainkohtainen käyttäytyminen jää kirjaston kehittäjän huoleksi. [6]

2.7 Kehitysympäristö

Sovelluksen kehityksessä on käytetty NetBeans IDE 6.8 -kehitysympäristöä. NetBeans valikoitui kehittäjäksi, koska se on ilmainen sovelluskehitin ja sillä voi PHP-dokumenttien lisäksi kirjoittaa XHTML-, CSS- ja JavaScript-dokumentit. Lisäksi se oli ennestään tuttu, joten ylimääräinen opetteluun kuuluva aika säästyi itse työn tekoon. Kehityksessä tarvittava palvelinympäristö on asennettu XAMPP-paketista, joka sisältää kaikki tarvittavat komponentit: Apache HTTP -palvelimen, MySQL-tietokantapalvelimen sekä PHP-kääntäjän. Lisäksi paketissa on mm. phpMyAdmin-työkalu, jonka avulla MySQL-tietokannan hallinta onnistuu helposti graafisen käyttöliittymän avulla.

3 MÄÄRITTELY

Tämän sovelluksen on tarkoitus mahdollistaa autovuokraamon varaustenhallinta ja varaustenhallintaan liittyvät toiminnot. Sovellukselta vaadittavia ominaisuuksia mietittiin yhdessä yrityksen henkilökunnan kanssa. Varaustenhallinnan mahdollistamiseksi täytyy sovelluksessa olla tuote- ja asiakashallinta sekä kalenteri, jonka avulla varauksia syötetään ja katsellaan. Näiden lisäksi välttämätön toiminto on käyttäjienhallinta, jonka avulla sovelluksen ylläpitäjä hallinnoi sovelluksen käyttäjiä sekä näiden käyttöoikeuksia sovelluksen resursseihin. Lisäksi sovellukseen lisättiin yrityksen toimintaa tukevia toimintoja, joita ovat huoltotietojen ylläpito sekä tilastot vuokratuotoista, huoltokuluista sekä käyttäjien tekemistä tapahtumista.

3.1 Toimintojen määrittely

Asiakasrekisteriin tallennetaan autonvuokrauksen kannalta oleelliset yhteystiedot. Asiakasrekisterin tietoja voidaan selata, lisätä, muokata ja poistaa. Asiakkaan tietoja tarkasteltaessa nähdään myös asiakkaan tekemät varaukset.

Tuoterekisteriin lisätään kaikki vuokrattavat tuotteet joita ovat ajoneuvot, peräkärryt ja näiden lisävarusteet. Tuotteiden erilaisuuden vuoksi eri tuotteilla voi olla hyvin erilaiset tuotetiedot ja pakolliset tuotetiedot on mietittävä tarkkaan. Tuoterekisterin tietoja voidaan selata, lisätä, muokata ja poistaa.

Huoltotietojen ylläpidon avulla kirjataan tuotteelle huolto- ja korjaustapahtumia. Huoltotietoja lisätessä voidaan syöttää huoltovälin sekä huoltokustannusten seuraamiseksi vaadittavat tiedot. Huoltotietoja voidaan lisätä, muokata ja poistaa.

Käyttäjähallinnassa sovelluksen ylläpitäjä voi hallita käyttäjiä ja käyttäjien oikeuksia. Tämä ominaisuus on lisätty, jotta resurssit saadaan vain niiden käyttöön, joille ne on tarkoitettu ja, että sovelluksen osat olisivat tarjottavissa tulevaisuudessa myös asiakkaiden käyttöön. Lisäksi sovelluksen käyttäjä voi itse vaihtaa salasanan sekä sähköpostiosoitteen.

Varauskalenterin avulla on mahdollista lisätä, muokata ja poistaa varauksia sekä tarkastella kalenterinäköymästä varaustilannetta. Varausta lisättäessä on mahdollista syöttää varauksen

yksilöimiseen ja ajankohtaan liittyvät tiedot sekä mahdolliset erityistiedot kyseiselle varaukselle.

Tilasto-sivulla näytetään tietoja, joita on aiemmin kerätty taulukkolaskentaohjelmaan tai joita ei ole seurattu ollenkaan. Näytettäviä tietoja ovat tuote- ja tuoteryhmäkohtaiset tulo- ja huoltokulut sekä käyttäjien tekemien varausten määrä.

Vuokrasopimuksen tulostaminen aloitetaan varauksen tietojen täydentämisellä ja tarkastamisella. Kun kaikki tiedot on syötetty, voidaan sopimus ja sopimusehdot tulostaa kahtena kappaleena.

Kirjautumisen avulla tarkistetaan käyttäjän oikeudet päästä sovelluksen resursseihin. Kirjautuminen toteutetaan käyttäjänimi-salasanaparin avulla. Kirjautuminen ei ole yksittäinen sivulla oleva osa, vaan sisältyy kirjautumistarkistusten myötä joka sivulle.

3.2 Tietojen määrittely

Tietojen määrittelyn avuksi sain yrityksestä vuokrasopimuskaavakkeen ja MS Excel-taulukoita, joissa on käsitelty tietoja, joita sovelluksessa tullaan käsittelemään. Lisäksi toimintojen määrittelyn perusteella pystyin miettimään tarvittavia tietoja.

Asiakasrekisterissä tarvittavat tiedot ovat:

- yrityksen nimi
- etunimi
- sukunimi
- henkilötunnus
- osoite
- postinumero
- postitoimipaikka
- sähköpostiosoite
- puhelinnumero
- ajokortin antoaika ja -paikka.

Tuotetta varattaessa pitää asiakasrekisteriin merkitä vähintään etunimi, sukunimi ja puhelinnumero. Vuokrattaessa tuotetta pitää muutkin tiedot syöttää, lukuun ottamatta yrityksen nimi ja sähköpostiosoite tietoja.

Tuotteille tarvittavia tietoja ovat:

- tuotetyyppi
- merkki
- malli
- polttoaine
- vaihdelaatikko
- mittarilukema
- rekisteröintipäivämäärä
- katsastuspäivämäärä
- rekisterinumero
- polttoainemittarin näyttämä
- virhe
- virheen kuvaus.

Koska samassa rekisterissä säilytetään kaikkien tuotteiden tietoja, ei jokaiselle tuotteella ole välttämättä käytettävissä samoja tietoja. Jokaiselta tuotteelta löytyy kuitenkin tuotetyyppi, merkki ja malli. Virhetiedolla tarkoitetaan tietoa tuotteen virheettömyydestä.

Huoltotietojen ylläpitoon tarvittavia pakollisia tietoja ovat

- päivämäärä
- huoltotoimenpiteen nimi
- huoltokuvaus
- mittarilukema
- summa.

Käyttäjätietojen ylläpitoon tarvittavia pakollisia tietoja ovat:

- käyttäjänimi
- salasana
- email
- käyttäjärooli.

Käyttäjärooliksi voidaan määrittää ylläpito, työntekijä tai asiakas.

Varaukselle tarvittavat tiedot on jaettu kahteen ryhmään. Ensin on listattu varausta lisättäessä tarvittavat tiedot ja toisena vuokrattaessa ja ajoneuvoa palautettaessa lisättävät tiedot. Varausta lisättäessä tarvittavat tiedot ovat:

- tuotteen tiedot
- asiakkaan tiedot (aiemman määrittelyn mukaan)
- sovittu noutopäivämäärä
- sovittu noutoaika
- sovittu palautuspäivämäärä
- sovittu palautusaika
- hinta
- lisätietoja.

Tuotetta vuokrattaessa ja palautettaessa lisättäviä tietoja ovat:

- noutoaika
- palautusaika
- omavastuu
- mittarilukema vuokrattaessa
- mittarilukema palautettaessa
- ajoneuvon käyttömaa (kotimaa/ulkomaa)
- ennakkomaksusumma
- lisämaksu.

4 SOVELLUKSEN TOIMINNOT

Tämän sovelluksen käyttöliittymää suunnitellessani kiinnitin huomiota käyttöliittymän selkeyteen ja helppouteen, koska sovellusta käyttävät paitsi kokeneet käyttäjät, myös tietokoneita vähän käyttäneet henkilöt. Pyrin pitämään näkymässä olevien toimintojen ja kenttien määrän mahdollisimman pienenä ja jakamaan ne selkeisiin kokonaisuuksiin. Sovelluksen jokaisen näkymän ylälaidassa sijaitsee sovelluksen valikko, jossa ovat käyttäjätyypille sallittujen resurssien valinnat. Jokaisesta valinnasta aukeaa sivu, jonka kautta hallinnoidaan kyseisen ryhmän toimintoja.

4.1 Kirjautuminen

Sovellukseen saavuttaessa avautuu kuvan 4.1 kirjautumissivu. Käyttäjän kirjautuessa kirjautumissivulta sovellukseen, alkaa varsinaisen sovelluksen käyttö. Tälle sivulle ohjataan aina silloin, kun käyttäjä ei ole kirjautuneena sovellukseen tai istunto on vanhentunut. Myös sovelluksesta ulos kirjaututtaessa palataan takaisin tälle sivulle.



The image shows a login form for 'Marevar - varausjärjestelmä'. It features a header with the name and a subtitle. Below the header are two input fields: 'Käyttäjätunnus:' and 'Salasana:'. A 'Kirjaudu' button is positioned below the password field.

Kuva 4.1. Sovelluksen kirjautumissivu.

4.2 Kalenteri

Sisäänkirjautumisen jälkeen käyttäjä ohjataan automaattisesti kuvassa 4.2 esitettävään kalenterinäkymään. Koska kalenteri on sovelluksen tärkein yksittäinen osa, ja sen voi myös olettaa olevan eniten käytetty, suoraan kalenteriin ohjaaminen parantaa käytettävyyttä. Kalenterin tuoteryhmittäinen kuukausinäkymä näyttää kaikkien tuoteryhmään kuuluvien merkinnät alku- ja loppukellonajoin. Tällöin on mahdollista nopeasti

vilkaisulla saada tietoa päivän tilanteesta. Kalenteri näkymää voi selata kahdella eri tavalla; Viikkonäkymän nuolilla selaamalla siirrytään viikko kerrallaan eteen tai taakse ja kuukausinäkömään nuolen painallus siirtää kuukauden kerrallaan.

Asiakkaat	Tuotteet	Kalenteri	Käyttäjät	Tilastot	adminmarevar	Kirjaudu ulos
-----------	----------	-----------	-----------	----------	--------------	---------------

Pakettiauto	Valitse
-------------	---------

	«	Viikko 15 Huhtikuu, 2010						»
	Ma 12.4.	Ti 13.4.	Ke 14.4.	To 15.4.	Pe 16.4.	La 17.4.	Su 18.4.	
Peugeot Partner XXX-1			09:00-12:00	09:00-14:00	10:00-20:00	12:00-16:00		
Ford Transit XXX-11					10:00->	->10:00		

	Huhtikuu, 2010							»
Vk	Ma	Ti	Ke	To	Pe	La	Su	
13					1	2	3	4
14	5	6	7	8	9	10	11	
15	12	13	14	15	16	17	18	
16	19	20	21	22	23	24	25	
17	26	27	28	29	30	1	2	

Kuva 4.2. Varausjärjestelmän kalenterinäkömä.

Ford Focus JJJ-346	Valitse
--------------------	---------

Ford Focus JJJ-346

«	Maanantai 3.5.2010		»
Aika	Tiedot		
0:00			
1:00			
2:00			
3:00			
4:00			
5:00			
6:00			
7:00			
8:00			
9:00			
10:00			
11:00			
12:00			
13:00			
14:00		Lisää varaus	
15:00		Lisää varaus	
16:00		varattu: Nouto: 16:00, Aivar Kuldnohk puh. 045055840	
17:00		varattu: Aivar Kuldnohk	
18:00		varattu: Aivar Kuldnohk	
19:00		varattu: Aivar Kuldnohk	
20:00		varattu: Palautus: 20:00	
21:00		Lisää varaus	
22:00		Lisää varaus	
23:00		Lisää varaus	

«	Toukokuu, 2010							»
Vk	Ma	Ti	Ke	To	Pe	La	Su	
17						1	2	
18	3	4	5	6	7	8	9	
19	10	11	12	13	14	15	16	
20	17	18	19	20	21	22	23	
21	24	25	26	27	28	29	30	
22	31	1	2	3	4	5	6	

Kuva 4.3. Kalenterin päivänäkömä.

Tarkempaa tietoa päivän tilanteesta saadaan klikkaamalla kalenterin ruutua halutun ajoneuvon ja päivämäärän kohdalta, jolloin avautuu kuvan 4.3 päivänäkömä. Kalenterin selaushetken takaiselta ajalta näytetään vain merkinnät, mikäli niitä on kyseisenä päivänä ollut ja tyhjät ruudut näytetään harmaina. Päivänäkömässä kalenterin merkinnöistä näkyvät

aloitus- ja lopetusaikojen lisäksi asiakkaan nimi, puhelinnumero sekä merkinnän tila. Asiakastiedoista tärkeimmät on nimi ja puhelinnumero, mikäli ilmenee tarve ottaa yhteyttä asiakkaaseen. Lisäksi tilatiedosta näkee heti, onko merkintä vasta varaus vai onko tuote jo vuokrattu. Kalenterimerkintää klikkaamalla aukeaa kuvan 4.4 näkymä, jossa voi tarkastella tehdyn varauksen tietoja. Näkymästä on mahdollisuus päästä muokkaamaan varauksen tietoja, peruuttaa varaus tai aloittaa tuotteen vuokraaminen. Vuokrattaessa aukeaa vuokrauslomake, johon täydennetään tarvittavat tiedot. Täydennyksen jälkeen tiedot tallennetaan ja lomake voidaan aukaista tulostettavaksi. Sovelluksesta tulostettu sopimuskaavake on esitettyä liitteenä.

Varauksen tiedot

[Peruuta varaus](#)

Asiakas

Yritys:

Nimi: Aivar Kuldnokk

Osoite: hõpõlõnläiskiksentie 1

Postinumero: 12345

Postitoimipaikka: Tallinna

Puhelin: 0450555840

Email:

Tuote

Tyyppi: Henkilöauto

Merkki ja malli: Ford Focus

Rekisterinumero: JJJ-346

Ajankohta ja hinta

Alkaa: 04.05.2010 20:00

Päättyy: 05.05.2010 10:00

Hinta: 150.00

Lisätietoja

[Muokkaa varausta](#) [vuokraa](#)

Kuva 4.4. Varauksen tietojen tarkastelu.

Myös varauksen lisääminen aloitetaan kalenterin päivänäkymästä klikkaamalla kalenterin kohtaa ”lisää varaus”. Varauksen lisääminen on jaettu neljään vaiheeseen, jotka ovat ajankohta, asiakastiedot, sovitut hinta- ja lisätiedot sekä valmis. Näin toimimalla kaikkia

mahdollisia tietoja ei tarvinnut mahduttaa samaan näkymään ja lopputuloksesta tuli selkeämpi. Varauksen lisäämisen vaiheet tulevat järjestyksestä, jossa varauksen tietoja yleensä käsitellään asiakkaan kanssa keskusteltaessa. Kuvassa 4.5 on varauksen lisäämisen ensimmäinen vaihe, jossa näytetään vuokrattavan ajoneuvon tiedot, tarkennetaan lähtöaika ja valitaan palautusaika.

Ajankohta Asiakastiedot Hinta ja lisätiedot Valmis

Tuote

Tyyppi: Henkilöauto

Merkki ja malli: Ford Focus

Rekisterinumero: JJJ-346

Nouto

Päivämäärä: 06.05.2010

Aika: 10: 00

Palautus

Päivämäärä: pp.kk.vvvv

Aika: :

seuraava

Kuva 4.5. Varauksen lisäämisen ensimmäinen vaihe.

Palautusajan valinnassa käytetään apuna päivämääränvalitsinta. Kun päivämäärä-kenttää klikataan hiirellä tai kenttä tulee muulla tavalla valituksi, aukeaa kuvan 4.6 päivämäärän valinta-komponentti. Vasta päivämäärän valinnan jälkeen on mahdollistettu valita kellonaika. Päivämäärän ja kellonajan valinta on rajoitettu vain vapaina oleviin aikoihin, jolloin palautusaika ei voi mennä seuraavan, mahdollisesti järjestelmässä olevan, varauksen päälle.

Palautus

Päivämäärä: pp.kk.vvvv

Aika:



Toukokuu 2010						
Ma	Ti	Ke	To	Pe	Lä	Su
						1
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Kuva 4.6. Päivämäärän valintatyökalu.

Toisessa vaiheessa lisätään asiakkaan tiedot. Tiedot voi lisätä varaukseen hakemalla hakukentän avulla asiakasrekisterissä olevia tietoja. Hakutulokset listataan näkymään, josta nimeä klikkaamalla tiedot tallentuvat suoraan varaukseen ja siirrytään kolmanteen vaiheeseen. Lisättäessä varausmerkintää asiakkaalle, jonka tietoja ei ole vielä asiakasrekisterissä, voidaan ne syöttää lomakkeen avulla. Etunimi, sukunimi ja puhelinnumero ovat pakollisia tietoja varausta tehtäessä, jotta asiakas voidaan yksilöidä ja yhteydenotto asiakkaaseen on myöhemmin mahdollista.

Kolmannessa vaiheessa näytetään asiakkaan tiedot ja pyydetään syöttämään vuokraukselle sovittu hinta- ja lisätiedot. Tämän jälkeen varauksen voi tallentaa ja siirtyä neljänteen vaiheeseen, jossa esitetään varauksen tiedot kootusti.

4.3 Asiakashallinta

Asiakashallinnan avulla voidaan käsitellä yrityksen asiakkaiden tietoja. Navigointi-valikon ”asiakkaat” -valintaa painamalla aukeaa kuvassa 4.7 esitettävän asiakashallinnan pääsivu, jolta asiakastietoja voidaan selata. Pääsivulla näkyvien asiakkaiden määrä on rajattu 20 nimeen kerrallaan ja ne esitetään aakkosjärjestyksessä sukunimen mukaan listattuna.

Marevar - varausjärjestelmä -		
Asiakkaat	Tuotteet	Kalenteri
Käyttäjät	Tilastot	adminmarevar
		Kirjaudu ulos
<< Edelliset	Lisää asiakas	Seuraavat >>
Nimi	Yritys	Postitoimipaikka
Autonvuokraaja, Aili Ekonen, Eemeli Hiltunen, Aatu		Helsinki
Jumppanen, Jorma Juselius, Jorma Kuldnokk, Aivar	joopajoo123 oy ab	
Tarvainen, Teemu		Jyväskylä

Kuva 4.7. Asiakashallinnan pääsivu.

Asiakashallinnan pääsivulta voidaan ”Lisää asiakas” -valintaa klikkaamalla aloittaa uuden asiakkaan tietojen syöttäminen rekisteriin. Tällöin avautuu kuvan 4.8 asiakastietojen syöttölomake. Lisättäessä tietoja lomakkeen avulla, tulee syöttää vähintään etunimi, sukunimi ja puhelinnumero. Yritettäessä tallentaa puutteellisin tiedoin täytettyä lomaketta näytetään virheilmoitus puutteellisten kenttien kohdalle.

Lisää asiakas

Etunimi:

Sukunimi:

Henkilötunnus:

Yritys:

Osoite:

Postinumero:

Postitoimipaikka:

Email:

Puhelin:

Kuva 4.8. Asiakkaan lisäyslomake.

Asiakkaan tietojen katselu tapahtuu klikkaamalla halutun asiakkaan riviä pääsivulla esitettävästä listauksesta. Tiedot avautuvat kuvassa 4.9 esitettävään näkymään. Näkymässä on mahdollisuus muokata asiakkaan tietoja tai poistaa ne kokonaan. ”Muokkaa” -painiketta painettaessa asiakkaan tiedot avataan kuvassa 4.7 esitettyyn lomakkeeseen, jolta tarvittavat

muutokset voi tehdä. Asiakkaan poistaminen on mahdollista vain, mikäli asiakkaalla ei ole varauksia järjestelmässä. Lisäksi sivulla näytetään asiakkaan varaukset. Tämä ominaisuus oli toivottu, jotta tietyn asiakkaan tulevan varauksen tiedot olisi nopeasti saatavilla. Toiminnosta voi olla hyötyä myös laskutettaessa, jolloin tarvitaan menneen varauksen tietoja.

Asiakkaat	Tuotteet	Kalenteri	Käyttäjät	Tilastot	adminmarevar	Kirjaudu ulos
-----------	----------	-----------	-----------	----------	--------------	---------------

Asiakastiedot

Etunimi:	Aili
Sukunimi:	Autonvuokraaja
Henkilötunnus:	123123-1234
Yritys:	
Osoite:	Vuokrakuja 1 A 2
Postinumero:	00001
Postitompipaikka:	Helsinki
Email:	
Puhelinnumero:	0123456789

[Poista](#)
[Muokkaa](#)

Tulevat varaukset		
Nouto	Palautus	Hinta
18.04.2010 9:00	18.04.2010 19:00	200.00

Menneet varaukset		
Nouto	Palautus	Hinta
14.04.2010 9:00	14.04.2010 12:00	100.00
15.04.2010 9:00	15.04.2010 14:00	100.00
16.04.2010 10:00	16.04.2010 20:00	100.00
17.04.2010 12:00	17.04.2010 16:00	100.80

Kuva 4.9. Asiakastietojen katselu.

4.4 Käyttäjienhallinta

Tässä sovelluksessa jokaisella käyttäjällä on käyttäjätyyppi. Käyttäjätyyppejä on kolmenlaisia: ylläpito, työntekijä ja asiakas. Käyttäjien pääsy sovelluksen resursseihin hallitaan tämän tyyppin avulla. Esimerkiksi pääsy käyttäjienhallinnan pääsivulle on sallittu vain ylläpitäjälle, muut voivat muokata vain omia tietojaan. Käyttäjienhallintaan tultaessa näkyy kuvan 4.10 näkymä. Tähän näkymään on listattu sovelluksen käyttäjät samaan tyyliin kuin asiakashallinnassa asiakkaat.

Marevar
- varausjärjestelmä -

Asiakkaat Tuotteet Kalenteri Käyttäjät Tilastot adminmarevar [Kirjaudu ulos](#)

Lisää käyttäjä

Nimi	Email	Käyttäjätyyppi		
adminmarevar	admin@matkarent.fi	Ylläpito	Muokkaa	Poista
Hugo22	hugo22@hugolandia.fi	Asiakas	Muokkaa	Poista
Teemutar	teemu.tarvainen@luukku.com	Työntekijä	Muokkaa	Poista

Kuva 4.10. Käyttäjän hallinnan pääsivu.

Koska käyttäjien tietoja tarvitsee tarkastella hyvin harvoin, yleensä vain muutoksia tehtäessä, ei varsinaista näkymää ole tietojen näyttämiseen. Käyttäjän tiedot saadaan näkyviin, kun aloittaa tietojen muokkaamisen, jolloin ne näytetään kuvan 4.11 muokkaus lomakkeessa. Samaa lomaketta käytetään uutta käyttäjää lisättäessä ja vanhan käyttäjän tietoja muokatessa.

Käyttäjätiedot

Nimi:

Email:

Salasana:

Salasana uudelleen:

Käyttäjätyyppi:

Luotu:

Perustaja:

Muokattu:

Muokkaja:

Kuva 4.11. Käyttäjän lisäys ja muokkaus lomake.

Lisäksi sovelluksen jokaisella käyttäjällä on ns. omasivu. Kuvassa 4.12 esitettävään oma-sivu-osioon pääsee yläreunan navigointi palkista omaa käyttäjänimeä klikkaamalla. Käyttäjänimi sijaitsee uloskirjautumis-painikkeen vieressä. Tältä sivulta käyttäjä voi

päivittää salasanan sekä sähköposti osoitteen. Sivulla näkyy myös käyttäjän omat varaukset, jotta käyttäjä voi seurata tekemiään varauksia. Tämä ominaisuus on lisätty tulevaisuutta ajatellen, jolloin sovellus on mahdollisesti tarjolla myös asiakkaiden käyttöön.

Päivitä tietosi		Varauksesi									
Uusi salasana:	<input type="text"/>	Tulevat varaukset									
Salasanan varmistus:	<input type="text"/>	Ei varauksia									
Vanha salasana:	<input type="text"/>	Menneet varaukset									
	<input type="button" value="Vaihda salasana"/>	<table border="1"> <thead> <tr> <th>Nouto</th> <th>Palautus</th> <th>Hinta</th> </tr> </thead> <tbody> <tr> <td>14.04.2010 5:00</td> <td>14.04.2010 14:00</td> <td>250.00</td> </tr> <tr> <td>14.04.2010 15:00</td> <td>15.04.2010 14:00</td> <td>300.00</td> </tr> </tbody> </table>	Nouto	Palautus	Hinta	14.04.2010 5:00	14.04.2010 14:00	250.00	14.04.2010 15:00	15.04.2010 14:00	300.00
Nouto	Palautus	Hinta									
14.04.2010 5:00	14.04.2010 14:00	250.00									
14.04.2010 15:00	15.04.2010 14:00	300.00									
Email:	<input type="text" value="teemu.tarvainen@luukku.com"/>										
	<input type="button" value="Tallenna"/>										

Kuva 4.12. Käyttäjän hallinnasta voi muokata omia tietoja.

4.5 Tuotteidenhallinta

Tuotteiden hallinnan avulla sovellukseen voidaan lisätä vuokrattavat tuotteet, katsella ja muokata niiden tietoja sekä tarvittaessa poistaa tuote. Lisäksi tätä kautta syötetään tuotekohtaiset huoltotiedot. Tuotehallinnan pääsivu on kuvassa 4.13. Sivun on hyvin samankaltainen asiakas- ja käyttäjähallinnan pääsivujen kanssa. Tuotteen tietoja pääsee suoraan muokkaamaan tai ne voi poistaa kokonaan pääsivulta. Tuotteen poistaminen on kuitenkin estetty, mikäli tuotteelle on olemassa varauksia.

Asiakkaat				Tuotteet		Kalenteri		Käyttäjät		Tilastot		adminmarevar		Kirjaudu ulos	
Edelliset		Lisää tuote		Seuraavat											
Tyyppi	Merkki	Malli	Rekisterinumero	Muokkaa	Poista										
Pikkubussi	Ford	Transit	FFF-666	Muokkaa	Poista										
Suksiboksi	Thule	265		Muokkaa	Poista										
Pikkubussi	VW	Caravelle	JJJ-345	Muokkaa	Poista										
Pakettiauto	Peugeot	Partner	XXX-1	Muokkaa	Poista										
Pakettiauto	Ford	Transit	XXX-11	Muokkaa	Poista										
Henkilöauto	Ford	Focus	JJJ-346	Muokkaa	Poista										

Kuva 4.13. Tuotehallinnan pääsivu.

Pääsivulta tuotteen kohdalta klikkaamalla avautuu kuvan 4.14 tuotetietonäkymä. Myös tästä näkymästä on mahdollista aloittaa tuotteen tietojen muokkaaminen tai poistaa tiedot. Lisäksi näkymässä esitetään huoltotiedot ajoneuvon perustietojen oikealla puolella.

Ajoneuvon tiedot		Huoltotiedot			
VW Caravelle ; JJJ-345		Päivämäärä	mittarilukema	Huollon nimi	Lisätietoja
Tyyppi:	Pikkubussi	12.12.2009	900	Käyttöauto	Öljynvaihto
Polttoaine:	Diesel	05.01.2010	21566	Käyttöauto	öljyhuolto
Vaihteet:	Manuaali	06.03.2010	32120	Veho	valohoittoa 2h
Mittarilukema:	35591	01.04.2010	34210	Raskone	Joojoo huo...
Rekisteröity:	20.04.2009	02.04.2010	34320	Raskone	edellisen ...
Katsastettu:	20.04.2009	03.04.2010	34352	Raskone	Valohoittoa
Polttoaine mittari:	100 %	04.04.2010	34450	Käyttöauto	lisää jooj...
Virheetön:	Ei	Lisää huoltotieto			
Virheen kuvaus:	noniinnojo				
Poista Muokkaa					

Kuva 4.14. Tuotetietojen esittäminen sivulla.

Huoltotietoja voi tarkastella tarkemmin klikkaamalla haluttua huoltotietoa, jolloin yksityiskohtaiset tiedot esitetään ajoneuvotietojen alla. Kuvassa 4.15 on esitetty huoltotiedon näyttäminen. Tästä näkymästä pääsee muokkaamaan huoltotietoa tai sen voi poistaa kokonaan. Muokattaessa ja lisättäessä huoltotietoja, aukeaa samaan kohtaan lomake, jolla tiedot tallennetaan.

Huollon tiedot	
Päivämäärä:	03.04.2010
Mittarilukema:	34352
Nimi:	Raskone
Huollon kuvaus:	Valohoittoa
Hinta Alv 0%:	0
Alv:	0
Hinta Alv 22%:	230
Poista Muokkaa	

Kuva 4.15. Huoltotiedon esittäminen.

4.6 Tilastot

Tilasto-sivun avulla sovelluksen ylläpitäjä voi seurata autojen vuokratuottoja, huoltokuluja sekä käyttäjien tekemien varausten, luovutusten ja vastaanottojen määrää. Tilasto-sivu on jaettu kahteen eri näkymään. Sivulle tultaessa avautuu kuvan 4.16 vuokratuloja sekä korjauskuluja esittävä näkymä. Sivulla voidaan tarkastella yhdellä kertaa joko kaikkien tuotteiden, yhden tuoteryhmän tai yksittäisen tuotteen tuloja sekä huolto ja korjauskuluja. Tiedot esitetään taulukossa, jossa tuotekohtaiset summat esitetään vuoden ajalta kuukausittain. Lisäksi esitetään kuukauden ja vuoden kokonaissumma.

Ajoneuvot Käyttäjät

Tyyppi: Tuote: Vuosi:

Vuokratulot

Ajoneuvo	Tammi	Helmi	Maalis	Huhti	Touko	Kesä	Heinä	Elo	Syys	Loka	Marras	Joulu
Peugeot Partner XXX-1				400.80								
Ford Transit XXX-11				250.00								
Yhteensä				650.8								

Kokonais summa: 650.8€

Huoltokulut

Ajoneuvo	Tammi	Helmi	Maalis	Huhti	Touko	Kesä	Heinä	Elo	Syys	Loka	Marras	Joulu
Peugeot Partner XXX-1				150								
Ford Transit XXX-11				150								
Yhteensä				150								

Kokonais summa: 150€

Kuva 4.16. Tilasto sivun vuokratulojen ja huoltokulujen esitys tuoteryhmästä pakettiauto.

Näkymien välillä liikutaan sivun ylä laidassa olevista ”Ajoneuvot”- ja ”Käyttäjät”-painikkeista. Kuvan 4.17 tapahtumien määrän seurannassa esitetään käyttäjien järjestelmässä tekemiä tapahtumia. Tapahtumat esitetään taulukoissa varaukset, vuokraukset ja vastaanotot vuosi kerrallaan. Vuodelle ajoittuvat tapahtumat esitetään käyttäjän mukaan kuukausittaisina määrinä. Lisäksi esitetään kuukausittainen kokonaissumma.

Ajoneuvot Käyttäjät

Vuosi:

Varaukset

Käyttäjä	Tammi	Helmi	Maalis	Huhti	Touko	Kesä	Heinä	Elo	Syys	Loka	Marras	Joulu
adminmarevar				10								
Hugo22												
Teemutar												
Yhteensä				10								

Yhteensä: 10 varausta

Luovutukset

Käyttäjä	Tammi	Helmi	Maalis	Huhti	Touko	Kesä	Heinä	Elo	Syys	Loka	Marras	Joulu
adminmarevar				5								
Hugo22												
Teemutar												
Yhteensä				5								

Yhteensä: 5 luovutusta

Vastaanotot

Käyttäjä	Tammi	Helmi	Maalis	Huhti	Touko	Kesä	Heinä	Elo	Syys	Loka	Marras	Joulu
adminmarevar				5								
Hugo22												
Teemutar												
Yhteensä				5								

Yhteensä: 5 vastaanottoa

Kuva 4.17. Tilasto -sivun näkymä käyttäjien tekemien tapahtumien seurantaan.

5 SOVELLUKSEN TOTEUTUS

Sovelluksen toteutuksessa ensimmäinen tehtävä on suunnitella tietokannan rakenne. Hyvin suunnitellun tietokannan päälle on hyvä alkaa suunnitella sovelluskoodin rakennetta, tietoturvaa ja testausta.

5.1 Tietokanta

Tietokannan rakenteen suunnittelun lähtökohtana oli aiemmin määritellyt toiminnot ja toiminnoille tarvittavat tiedot. Annoin tietokannalle nimen Marevar sovelluksen työnimen mukaan. Tietokantaan loin seuraavat taulut:

- asiakas
- huolto
- käyttäjä
- tuote
- varaus.

Taulujen sarakkeet sain toimintojen ja tietojen määrittelyn yhteydessä määritellyistä tiedoista. Lisäksi varaus-tilaan laitoin kaikki varaukseen ja vuokraukseen liittyvät tiedot.

Tämän lisäksi jokaiseen tauluun lisäsin sarakkeet:

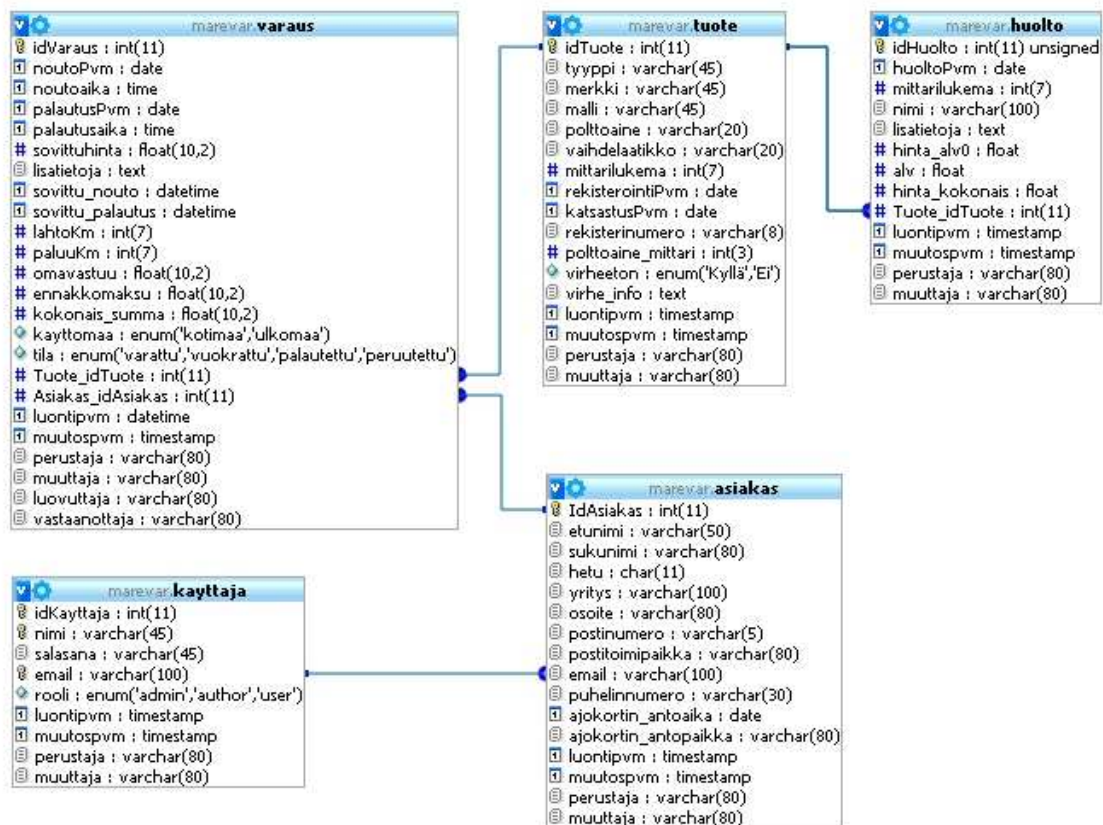
- luontipvm
- muutospvm
- perustaja
- muuttaja.

Sarakkeisiin luontipvm ja muutospvm tallennetaan taulun rivin luonti- ja muutoshetken aikaleimat. Sarakkeisiin perustaja ja muuttaja tallennetaan kirjautuneena olevan käyttäjän käyttäjänimi. Näiden tietojen perusteella pystytään jäljittämään tietoja lisänneet ja muuttaneet käyttäjät. Jokaisen taulun perusavaimeksi lisäsin sarakkeen, jonka nimi on aina muotoa idTaulunnimi. Esimerkiksi asiakas-tilan id on idAsiakas.

Varaus-tilaan lisäsin näiden lisäksi kentät:

- luovuttaja
- vastaanottaja
- tila.

Luovuttaja- ja vastaanottaja-sarakkeisiin tallennetaan kyseisen operaation tekevän käyttäjän nimi, joka saadaan sovelluksen kirjautumistiedosta. Näiden sarakkeiden avulla voidaan tilasto-sivulle laskea tapahtumien määriä taulusta. Taulun tila-sarake puolestaan kertoo varauksen tilan. Tila voi olla varattu, vuokrattu, palautettu tai peruttu. Tämän sarakkeen avulla saadaan helposti selvitettyä merkinnän tila kyselyä tehtäessä, eikä varauksen tiedot katoa kokonaan, mikäli sen peruuttaa. Kuvassa 5.1 on kuvattu sovelluksen tietokantarakenne.



Kuva 5.1. Marevar-varausjärjestelmän tietokannan rakennekaavio.

Tietokantaa on sovelluksessa käytetty PHP:n tarjoaman MySQLi-laajennuksen avulla. Laajennus mahdollistaa MySQL-version 4.1.3 myötä tulleiden uusien toiminnallisuuksien käytön [7, s. 361.]

5.2 Sovelluksen rakenne

Sovellus on toteutettu proseduraalisella ohjelmointimenetelmällä. Tästä huolimatta sovelluksessa on neljä luokkaa, mutta myös näiden luokkien tehtävät olisi voinut hoitaa tavallisilla funktioilla. Tässä luvussa kuvaan sovelluksen rakenteen.

5.2.1 Hakemistorakenne

Sovelluksen päähakemiston sisältö on esitetty kuvassa 5.2. Se sisältää näkymien tuottamisessa käytetyt, suoraan kutsuttavissa olevat tiedostot sekä omat alihakemistot erityyppisille tiedostoille.



Kuva 5.2. Sovelluksen päähakemiston sisältö.

ajax-kansio sisältää sovelluksen AJAX-toimintoja varten luodut tiedostot. Tiedostot on nimetty sivun tai toiminnon mukaan lisäämällä ”_ajax” tiedostonimeen. Esimerkiksi addBooking.php-sivulla käytettyjen AJAX-toiminnallisuuksien vaatimat toiminnot ovat tiedostossa nimellä addBooking_ajax.php.

class-kansio sisältää sovelluksen neljä luokkaa. Luokan sisältävät tiedostot on nimetty niiden käyttötarkoituksen ja luokan nimen mukaan. Esimerkiksi Session-luokka on session.php-tiedostossa.

include-kansio sisältää päänäkymiin sisällytettävät tiedostot, joita ei voi kutsua suoraan selaimella. Tiedostoja ovat esimerkiksi sivun navigointi-valikon muodostava navigation.php ja varauksen tiedot näyttävä showBooking.php.

jQuery-kansio sisältää sovelluksessa käytetyn jQuery JavaScript -kirjaston vaatimat tiedostot jaettuna tiedostotyyppin mukaan kahteen alikansioon. Kansio sisältää kirjaston mukana tulevat tyylitiedostot css-kansiossa ja JavaScript-tiedostot js-kansiossa.

js-kansio sisältää sovellukseen kirjoitetut JavaScript-tiedostot. Tiedostot on nimetty sivun mukaan, jolla kyseistä tiedostoa käytetään. Näiden tiedostojen avulla käytetään jQuery-kirjaston tarjoamia toimintoja.

nbproject-kansio sisältää sovelluskehittimen projektiin liittyviä tiedostoja. Tämä kansio on vain sovelluskehittäjä varten, eikä sitä siirretä kehitysympäristöstä tuotantopalvelimelle sovellusta käyttöönotettaessa.

stylesheet-kansio sisältää sovelluksen kaksi tyylitiedostoa. Toinen tyylitiedosto on jokaisen ruudulla näkyvän sivun ulkoasun määrittelyä varten ja toinen vuokrasopimusten tulostamiseen käytetty tyylitiedosto.

5.2.2 Luokat

Sovelluksessa on neljä luokkaa, jotka ovat Database, Form, Session ja Calendar. Näiden avulla sovelluksen koodia on jaettu loogisiin kokonaisuuksiin. Lisäksi niiden avulla koodin määrää ja uudelleen kirjoittamisen tarvetta on saatu vähennettyä. Näiden luokkien sisältämät toiminnot olisi voinut toteuttaa myös tavallisina funktioina, jolloin sovelluksen toiminta olisi saatu täysin samanlaiseksi.

Database-luokkaan on sijoitettu kaikki tietokantayhteyksiin ja kyselyihin liittyvät toiminnot. Tarvittaessa tietokantayhteyttä on sivulla muodostettava tästä luokasta olio, minkä jälkeen kaikki tietokanta toiminnot ovat käytettävissä. Tällä tavalla tietokantayhteyden asetukset tarvitsee muuttaa vain yhteen tiedostoon siirrettäessä sovellusta palvelimelle. Myös tehtäessä tietokantaan muutoksia muutoskohteiden määrä pienenee oleellisesti.

Form-luokka toimii apuna muodostettaessa sivustolla tarvittavia lomakkeita. Luokan avulla tulostetaan sovelluksessa käytetyt kentät sekä kenttien virheilmoitukset. Tämä on tehty selkeyttämään koodia, kun sivun ulkoasu ja logiikka toteutetaan samassa tiedostossa. Samalla myös XHTML-lomakkeilla käytetyn koodin uudelleenkirjoittamisen tarve vähenee huomattavasti.

Session-luokkaan on toteutettu session, eli istunnon, aloittamiseen sekä kirjautumisen tarkistamiseen tarvittavat metodit. Jokaisen sivun alussa kutsutaan luokan `getInstance()`-metodia, joka palauttaa jo olemassa olevan olion luokasta tai luo uuden ja palauttaa sen. Tämän jälkeen kutsutaan `checkLogin()`-metodia, joka palauttaa kirjautuneen käyttäjän käyttäjätiedot taulukoituna tai ohjaa kirjautumattoman käyttäjän kirjautumissivulle.

Calendar-luokka toimii kalenterin toimintojen muodostamisen apuna. Luokan avulla on mahdollista tuottaa kalenterin viikko-, päivä- ja kuukausinäkymät sekä näkymiin kuuluvat tapahtumat.

5.3.3 Sivujen koodin rakenne

Sovelluksen sivuja toteutettaessa jokaiselle sivulle on luotu oma tiedostonsa, joka muodostaa toiminnallisuuden yhdessä pienempien, sivuun sisällytettävien, aputiedostojen kanssa sekä käyttämällä apuna sovellukseen kehitettyjä luokkia. Tällä tavoin on sivun toteutukseen tarvittavaa ohjelmakoodia saatu yksinkertaistettua ja hallintaa helpotettua.

Sovelluksen jokaisen näkyvän sivun toteuttavan tiedoston rakenne on samanlainen ja ne alkavat ohjelmariveillä

```
date_default_timezone_set("Europe/Helsinki");
mb_internal_encoding("UTF-8");
require_once '/class/database.php';
require_once '/class/session.php';
require_once '/class/form.php';
$session = Session::getInstance();
$db = Database::getInstance();
$form = Form::getInstance();
```

joilla asetetaan aikavyöhyke ja mbstring-lisäosan merkistökooodaus. Tämän jälkeen sisällytetään tiedostoon tarvittavat luokkatiedostot sekä luodaan luokista oliot.

Kun tarvittavat asetukset on asetettu ja oliot muodostettu, tarkistetaan onko käyttäjä kirjautunut sivustolle. Tätä varten on ohjelmoitu Session-luokkaan `checkLogin()`-metodi, joka palauttaa kirjautuneen käyttäjän tiedot taulukkona tai uudelleen ohjaa anonyymi käyttäjän sovelluksen kirjautumis sivulle. Kirjautumis tarkistus tehdään jokaisella sivulla lukuun ottamatta kirjautumis sivua.

Kirjautumis tarkistuksen jälkeen toteutetaan sivustolta tulevien syötteiden tarkistaminen. Esimerkkinä uloskirjautumisen suorittava osa

```

if (isset($_GET["logout"]) &&
    $_GET['logout'] == "true" &&
    isset($_GET['token']) &&
    mb_strlen($_GET['token']) == 32 &&
    $_SESSION['token'] == $_GET['token']) {

    $sessio->logout();
    unset($sessio);
    $sessio = Session::getInstance();
}

```

jossa tarkistetaan tarvittavien muuttujien olemassa olo `isset()`-funktiolla ja arvojen oikeellisuus. Mikäli muuttujat on asetettu ja arvot ovat oikeat, kutsutaan `$sessio`-olion avulla session luokan `logout()`-metodia, joka tuhoaa session tiedot ja sessiotunnistetta säilyttävän keksin. Tämän jälkeen tuhoaan `unset()`-funktiolla `$sessio`-olio ja viimeisenä muodostetaan uusi `$sessio`-olio, jotta uudelleen kirjautuminen olisi mahdollista ilman sivun päivittämistä.

Tarkistusten jälkeen tiedostoissa tehdään XHTML-dokumentin vaatimat määrittelyt `<head>`-osion sisältävine JavaScript- ja CSS-tiedostojen linkkauksineen. Itse sivulla näkyvä sisältö on jaettu `<div>`-elementeillä header- ja content -osiin. Sivujen header -osaan sisällytetään sovelluksen nimi ja perusnavigointi `navigation.php`-tiedostosta ja content-osaan toteutetaan sivuston toimintaan tarvittava sisältö XHTML-merkkeistä ja PHP-koodia käyttäen.

5.3.4 Sovelluksen JavaScript

Käyttöliittymän käytettävyyttä on osalla sivuista parannettu JavaScriptin avulla. JavaScriptiä kirjoitettaessa on käytetty apuna ilmaista jQuery-kirjastoa. Esimerkiksi kuvassa 4.5 esitetty Datepicker-päivämäärävalitsin ja kaikki sivustolla käytetty AJAX-toiminnallisuus on toteutettu jQuery-kirjaston avulla. Valmiin kirjaston käyttäminen helpottaa sovelluksen toteuttamista tarjoamalla yleisesti tarvittuja komponentteja valmiiksi saataville. Myös selainten vaihteleva tuki AJAX:n käyttämälle XMLHttpRequest-oliolle on otettu kirjastossa valmiiksi huomioon, eikä enää sovelluksen kehittäjän tarvitse tätä asiaa miettiä.

jQuery-kirjaston ja DatePicker-komponentin käyttöön ottaminen on yksinkertaista ja hyvin dokumentoitu jQuery:n internet-sivuilla. Käytännössä kirjaston tiedostot pitää sisällyttää XHTML-dokumentin <head>-osioon <link>- ja <script>-tagien avulla, kuten muutkin JavaScript- ja CSS-tiedostot. Tämän jälkeen kirjaston tarjoama toiminnallisuus on käytettävissä sivulla.

DatePicker-komponenttia käytettäessä komponentti pitää alustaa ennen käyttöä. Tämä on toteutettu sovellukseen kutsumalla sivun <body>-tagin onload-attribuutissa komponentin alustavaa funktiota

```
function initDatePickerStart(min_date,max_date,selected){
    $('#date_start').datepicker({
        dateFormat: 'dd.mm.yy',
        firstDay: 1,
        minDate: min_date,
        maxDate: max_date,
        defaultDate: selected,
        hideIfNoPrevNext: true
    });
    $('#date_start').datepicker($.datepicker.regional['fi']);
}
```

Tämä funktio on toteutettu sivun nimellä varustettuun JavaScript-tiedostoon ja se ottaa parametreikseen kolme päivämäärää. Näistä ensimmäinen määrittelee pienimmän mahdollisen valittavan päivämäärän, toinen suurimman ja kolmas tällä hetkellä valitun päivämäärän. Funktiossa määritellään mahdollisten päivämäärä valintojen lisäksi myös päivämäärän muoto, viikon ensimmäinen päivä, ”seuraava/edellinen”-painikkeiden näkyminen sekä viimeisenä otetaan käyttöön suomen kieli.

Myös AJAX-toiminnallisuuden ohjelmoiminen helpottuu huomattavasti kirjaston avulla. Yksinkertaisimmillaan tarvitsee vain kutsua XHTML-dokumentin elementtien onchange-attribuutissa tehtävän hoitavaa funktiota, joka sisältää esimerkiksi koodin

```

$.ajax({
    type: "POST",
    url: "editBooking_ajax.php",
    data: {
        date: date,
        bid: bid,
        datestart: dateStart
    },
    success: function(data){
        $("#select_time_end").html(data);
    }
});

```

Koodissa määritellään ”POST”-tiedonsiirtotyyppi, kohteeksi editBooking_ajax.php-tiedosto ja HTTP-pyynnön mukana toimitettava data. Lisäksi voidaan määritellä pyynnön vastauksen saavuttua kutsuttava funktio. Tässä ei kuitenkaan määritellä erikseen kutsuttavaa funktiota, vaan vastauksessa saapuvan datan käsittelijä on toteutettu suoraan success-asetuksen yhteyteen, minkä avulla data liitetään sivulle.

AJAX-toiminnallisuutta käyttäviä toimintoja ovat sovelluksessa lähinnä kellonaikojen valinnat ja huoltotietojen näyttäminen tuote-sivuilla. Esimerkiksi varausta lisättäessä tai muutettaessa minkä tahansa päivämäärä valinnan muuttamisen jälkeen ladataan palvelimelta AJAX:n avulla päivämäärälle mahdolliset kellonaikavalinnat. Tämä parantaa käytettävyyttä helpottamalla kellonaikojen valintaa sekä vähentää päällekkäisten varausten lisäämisen mahdollisuutta.

5.3 Tietoturva

PHP-sovelluksia uhkaa samat haavoittuvuudet kuin muitakin web-sovelluksia. Olen pyrkinyt estämään yleisimmät haavoittuvuudet ja kuvaamaan tässä luvussa suojauksen toteutuksen.

5.3.1 Cross-site scripting (XSS)

Web-sivustojen yksi suurimmista uhkista on Cross-site scripting (XSS). Sivustolla olevan haavoittuvuuden avulla hyökkääjä voi lisätä sivustolle sitä haittaavaa koodia. Tämän koodin avulla sivusto voidaan saada näyttämään sinne kuulumatonta sisältöä tai

suorittamaan hyökkäjän haluamia toimintoja. Usein tämä tapahtuu muokkaamalla sivun osoitetta liittämällä siihen sivustolla käytetyn muuttujan arvoksi JavaScript-koodia ja lähettämällä linkki henkilölle, jonka tietoja halutaan kalastella. Toinen mahdollisuus on lähettää sivulle sivulla näytettävää materiaalia, kuten vieraskirjan merkintä.

Sovelluksessa liikuteltavan datan alkuperästä ei voi olla koskaan täysin varma. Niinpä yleinen ohje on olettaa datan olevan haitallista, kunnes toisin on todistettu. XSS-hyökkäyksiltä voi suojautua kunnollisella datan suodatuksella. Jo pelkkä PHP:n `htmlspecialchars()`- ja `striptags()`-funktioiden käyttö estää lomakkeilta tai osoiteriviltä lähetetyn haitallisen koodin suorittamisen. [8]

5.3.2 Cross-site request forgery (CSRF)

CSRF-hyökkäyksessä käytetään hyväksi käyttäjän avoinna olevaa istuntoa. Hyökkääjä asettaa sivulle linkin, joka osoittaa ennalta arvaamattomaan paikkaan. Kun käyttäjä klikkaa toisessa selainikkunassa sijaitsevaa linkkiä, saa hyökkääjä aikaan haluamansa toiminnon palvelussa, johon linkkiä klikannut käyttäjä on kirjautunut.

CSRF-hyökkäykseltä suojautumiseen vaaditaan jokaiseen HTTP-pyyntöön lisätty satunnainen merkkijono, joka luodaan lauseella

```
$token = md5(uniqid(rand(), true));
```

Sama merkkijono tallennetaan myös istuntoon, jolloin lomakkeen käsittelijässä voidaan verrata istunnossa olevaa merkkijonoa HTTP-pyyntöön mukana tulleeseen merkkijonoon. Mikäli merkkijonot eivät vastaa toisiaan, pyyntö voidaan hylätä. [8]

5.3.3 SQL-injektio

SQL-injektio perustuu myös virheellisiin syötteisiin sovelluksessa. Virheellisillä syötteillä on tarkoitus saada aikaan hyökkäjälle mieluinen SQL-lause. Tällaisilla lauseilla on mahdollista saada tietokannasta tietoa, johon oikeudet eivät tavallisesti riitä, tai jopa tuhota kokonaisia tauluja. SQL-injektionilta suojautumiseksi kaikki tietokantaan menevä tieto suodatetaan, käsitellään `mysqli_real_escape_string()`-funktiolla ja lisätään heittomerkkeihin. [8]

5.3.4 Istunnon kaappaus

PHP:ssä istunnot käyttävät evästeitä istunnon tunnistamiseen. Kun istunto aloitetaan, lähetetään selaimelle PHPSESSID-niminen eväste, jonka arvo on satunnainen merkkijono. Istunnon tiedot tallennetaan palvelimelle merkkijonoa vastaavaan tiedostoon [4, s. 273]. Sivustolle hyökkäävä henkilö haluaa yleensä saada toisen sivustolla vierailevan istunnon haltuunsa. Tältä suojautumiseen on sovelluksessa käytetty seuraavia keinoja.

Ensimmäinen keino on käyttää PHP:n `session_regenerate_id()`-funktioita istunnon tunnisteen vaihtamiseen aina sivua ladattaessa ja käyttäjän kirjautuessa sivustolle. Tämä estää tehokkaasti istuntotunnisteen arvaamisen [8].

Toinen sovelluksessa käytetty tapa suojautua istunnon kaappaukselta on HTTP-otsikkotiedoissa kulkeva User-Agent-tiedon vertailu. Ensimmäisellä sivun latauksella selaimen lähettämään User-Agent-tietoon lisätään koodilla

```
$str = $_SERVER['HTTP_USER_AGENT'];  
$str .= "merkkeja123!?" ;  
$_SESSION['sormenjalki'] = md5($str);
```

merkkejä ja merkkijonosta tallennetaan md5-tiiviste istuntoon. Tämän jälkeen aina HTTP-pyyntöön mukana tulevasta User-Agent-tiedosta muodostetaan samalla tavalla md5-tiiviste ja näitä vertaillaan keskenään. Mikäli tiedot ovat erilaiset, tuhoetaan istunto ja ohjataan käyttäjä kirjautumissivulle. [8, s. 29.]

5.3.5 Resurssien suojaus

Ohjelman tiedostojen suojaukseen on sovelluksessa käytetty kahta eri tapaa. Sovelluksen päähakemistossa olevat tiedostot on suojattu sovelluksen käyttäjätasojen ja niiden tarkistuksen avulla, jolloin tiedoston voi avata vain riittävän korkeat oikeudet omistava ja kirjautuneena oleva käyttäjä. Sivuille sisällytettävät tiedostot ovat include-kansiossa ja näitä tiedostoja ei ole tarkoitus voida avata suoraan selaimella. Nämä tiedostot on suojattu mahdollisen suoran avaamisen estämiseksi sijoittamalla jokaisen tiedoston alkuun tarkistus

```

if(count(get_included_files()) < 2) {
    echo "Kutsuit tiedostoa suoraan.";
    exit ();
}

```

Tarkistuksessa käytetty `get_included_files()`-funktio palauttaa kaikkien sisällytettyjen tiedostojen ja itsensä nimen taulukkona. Kutsuttaessa tiedostoa suoraan taulukossa on vain yksi tiedostonimi, jolloin virheilmoitus tulostuu ja suoritus keskeytyy. [4 s. 379.]

Alihakemistojen tiedostolistauksen estämiseksi on jokaiseen alihakemistoon lisätty tyhjä `index.php`-tiedosto, jonne on toteutettu vain uudelleenohjaus sovelluksen kirjautumisivulle. Tiedostolistauksen voisi Apache-palvelimella estää myös `.htaccess`-tiedostolla, mutta koska lopullista tuotantopalvelinta ei ole vielä päätetty, ei sovellusta kannata sitoa yhteen tiettyyn palvelimeen.

5.4 Testaus

Sovelluksen ulkoasun ja toimintojen toimivuus on testattu uusimmilla ja ehkä yleisimmillä Microsoft Internet Explorer 8 ja Mozilla Firefox 3.6 -selaimilla. Ulkoasu on testattu aina yhden isomman kokonaisuuden valmistuttua. Sovelluksen toiminta on testattu ohjelmointityön edetessä testaamalla jokainen sovellukseen ohjelmoitu toiminto yksittäin. Tarkoituksena oli testata kaikki lomakekentät erilaisilla syötteillä, jokaisen painikkeen ja linkin toiminta sekä varauksen tietojen syötössä käytettävien, AJAX-tekniikalla päivitettävien, kellonaikavalintojen toimivuus. Kaikki testaamistyö on tehty manuaalisesti ja se onkin ollut paljon aikaa vievä osuus.

Esimerkiksi kirjautumislomakkeen testaus on suoritettu syöttämällä käyttäjänimi- ja salasana-kenttiin seuraavanlaiset syötteet:

- Kentät jätetään tyhjäksi.
- Käyttäjätunnus-kenttä on tyhjä, salasana kirjoitetaan.
- Salasana-kenttä on tyhjä, käyttäjätunnus kirjoitetaan.
- Kenttiin syötetään oikea tunnus, mutta väärä salasana.
- Kenttiin syötetään väärä tunnus, mutta oikea salasana.
- Käyttäjätunnus syötetään isoilla kirjaimilla ja salasana oikein.
- Salasana syötetään isoilla kirjaimilla ja käyttäjätunnus oikein.

- Käyttäjätunnus ja salasana syötetään oikein.

Näihin syötteisiin tulostetaan virheilmoitus lukuun ottamatta viimeistä kohtaa, jossa syötetään oikea tunnus-salasana pari ja sovellukseen kirjautuminen tapahtuu.

Jokainen sovelluksen sivu on myös validoitu W3C:n tarjoamilla validaattoreilla ja näin varmistettu XHTML-dokumenttien ja CSS-tyylimäärittelyjen oikeamuotoisuus. Määrittysten mukaiset, validit, sivut toimivat eri selaimissa varmemmin kuten on tarkoitettu.

6 SOVELLUKSEN KÄYTTÖÖNOTTO

Opinnäytetyönä kehitetty sovellus on ensimmäinen kehitysversio, joka on otettu koekäyttöön yrityksen omalla palvelimella. Koekäytön aikana sovelluksen toimintaan ja käyttöön perehdytään yrityksessä ja korjataan myös mahdollisia virhetoimintoja sekä toteutetaan käyttöliittymään tarvittavia muutoksia tai uusia ominaisuuksia. Kun sovellus todetaan toimivaksi, siirretään se lopulliseen sijoituspaikkaan, ulkopuolisen palveluntarjoajan palvelimelle.

6.1 Palvelinvaatimukset

Sovellus vaatii toimiakseen www-palvelimen, jolla on PHP-tuki ja MySQL-tietokanta. PHP:n versio tulee olla sovelluksessa käytettyjen ratkaisujen vuoksi 5.2 tai uudempi. Palvelimelle pitää olla myös asennettuna PHP:n Mbstring-lisäosa. Lisäosa sisältää funktiot merkkijonojen käsittelyyn ja mahdollistaa UTF-8-merkistökoodauksen käyttämisen. MySQL-palvelimen versio pitää olla vähintään 4.1.3, mutta sovellus on testattu vain käyttäen versiota 5.1.

6.2 Sovelluksen siirtäminen palvelimelle

Sovelluksen palvelimelle siirtäminen alkoi luomalla sovelluksen käyttämä tietokanta tuotantopalvelimelle. Tietokannan siirtäminen tapahtui phpMyAdmin-hallintaohjelmalla, joka oli käytössä kehitysympäristössä sekä kohdepalvelimella. Tietokannan siirron toteuttamiseksi oli ensin tallennettava tietokannan rakenne ja tiedot kehitysympäristön hallintaohjelman vienti-toiminnolla tiedostoon. Toiminto tallentaa tiedostoon SQL-lausekkeet, jotka muodostavat tietokannan taulut, ja luo datan tauluihin. Tämän jälkeen kohdepalvelimen hallintaohjelman avulla täytyy luoda tietokanta. Kun tietokanta on luotu, voidaan tietokantaan muodostaa taulut ja data kehitysympäristössä luodusta tiedostosta. Tämä tapahtuu hallintaohjelman tuonti-toiminnolla.

Tietokannan luomisen jälkeen sovelluksen tarvitsemat tiedostot siirrettiin palvelimelle. Tässä yhteydessä on muutettava database.php-tiedostoon tietokannan käyttäjätunnus ja salasana. Joskus voi olla tarvetta myös muuttaa palvelimen osoite ja käytettävän portin numero. Muita tarvittavia muutoksia olivat virheelliset hakemistopolut require_once()-

funktioiden yhteydessä. Ongelma huomattiin vasta sovelluksen tuotantopalvelimelle siirron jälkeen, koska kehitysympäristössä toimi ongelmitta kaksi eri tapaa tiedoston sisällyttämiseen erihakemistopolusta. Hakemistonpolkuna toimi sekä ”/kansio/tiedosto” että ”kansio/tiedosto/”, joista vain jälkimmäinen tapa toimi tuotantopalvelimella.

6.3 Käyttöönottestaus

Tietokannan ja tiedostojen siirtämisen jälkeen testattiin sovelluksen toiminta. Koska tietokanta siirrettiin kehitysympäristöstä suoraan, se sisältää sovelluksen käyttäjätunnukset ja kehitysympäristössä käytettyä testidataa. Sovelluskehityksen aikaisen tarkemman testauksen katsottiin riittävän, joten tässä vaiheessa ei testattu kenttäkohtaisesti koko sovellusta; vaan muutama kenttä testattiin, jotta voitiin havaita virhesyötteiden suodatuksen toimivuus myös tällä palvelimella. Pääpaino testauksessa oli sovelluksen linkkien, painikkeiden ja jokaisen toiminnon läpikäynnissä. Näin voitiin todeta tietokantaoperaatioiden sekä tiedostoja sisällyttävien funktioiden toimivuus.

7 PÄÄTELMÄT

Työssä toteutettiin varausjärjestelmä autovuokraamolle. Toteutuksen avulla on helppo hallinnoida asiakastietoja, asiakkaiden varauksia, vuokrattavien tuotteiden varauksia ja seurata tuotteilla kerättyjä tuloja sekä huoltokuluja. Sovellus tuo yritykselle lisäarvoa vähentämällä ylimääräistä puhelinliikennettä ja helpottamalla päivittäistä työskentelyä.

Varausjärjestelmän tarpeesta oli yrityksessä keskusteltu jo aiemmin ja tammikuussa 2010 työn tekemistä opinnäytetyönä alettiin suunnitella. Työ saatiin alulle nopeasti, jo helmikuun alussa, ja työn valmistumista suunniteltiin toukokuun alkupuolelle. Työn aikataulutukseen haasteita toivat sovelluksen toimintojen lisääminen kehityksen aikana ja henkilökohtaiset kiireet. Mahdolliset ongelmat oli huomioitu jo projektia suunniteltaessa ja niihin varauduttu, joten työn valmistuminen viivästyi vain vähän.

Ohjelmointikielen valinnassa oli aluksi kaksi vaihtoehtoa, Java ja PHP. Javasta oli hieman aiempaa kokemusta, mutta opinnäytetyön valvojan, lehtori Jussi Koistisen ehdottamaan PHP-kieleen tutustumisen jälkeen oli valinta selvä. Vaikka käytetystä kielestä ei ollutkaan aiempaa kokemusta, oli alkuun pääseminen helppoa ja jo muutaman päivän tutustumisen perusteella osasi välttää web-sovelluksiin liittyvät pahimmat sudenkuopat.

Sovelluksen toimintojen suunnittelua helpotti autovuokraamon toiminnan tunteminen, ja mielestäni toiminnoista onnistuttiin saamaan selkeitä ja helposti käytettäviä, vaikka ulkoasu ei ole aivan ammattimainen. Tämäkään sovellus ei ole täydellinen ja mahdollisuuksia toimintojen paranteluun ja lisäämiseen on olemassa lähes rajattomasti. Ensimmäinen, ja ehkä tärkein, asia on tutustua tulevaisuudessa SSL-salaukseen ja ottaa se käyttöön tässä sovelluksessa. Sovelluksen tietoturva on jo nyt suhteellisen hyvä, eikä ihan yksinkertaisimmilla toimenpiteillä sivuille voi tuottaa haittaa, mutta salaamaton liikenne on aina riski.

Yrityksestä saamani palaute on ollut hyvää ja kokonaisuutena katson onnistuneeni hyvin, vaikka aiempaa kokemusta web-ohjelmoinnista ei juuri ollutkaan. Tämän työn aikana tutustuinkin web-sovelluksissa tärkeisiin tietoturva-asioihin ja opin, kuinka PHP-ohjelmointikielillä toteutetun sovelluksen uhkia voidaan helposti karsia.

LÄHTEET

- 1 XHTML Tutorial. Artikkelel www-sivulta, W3School.com.
13. huhtikuuta 2010 [online]
<http://www.w3schools.com/xhtml/>
- 2 JavaScript Tutorial. Artikkelel www-sivulta, W3School.com
13. huhtikuuta 2010 [online]
<http://www.w3schools.com/js/>
- 3 CSS Tutorial. Artikkelel www-sivulta, W3School.com
14. huhtikuuta 2010 [online]
<http://www.w3schools.com/css/>
- 4 Heinisuo, Rami ja Rauta, Ilkka (2007). PHP ja MySQL, tietokantapohjaiset verkkopalvelut, 4. uudistettu painos, Talentum Media Oy
- 5 MySQL. Tietokantapalvelimen kotisivut,
19. huhtikuuta 2010 [online]
<http://dev.mysql.com/doc/refman/5.1/en/features.html>
- 6 AJAX Tutorial. Artikkelel www-sivulla,
19. huhtikuuta 2010 [online]
<http://www.w3schools.com/Ajax/>
- 7 Julie C. Meloni, Sams Teach Yourself PHP, MySQL and Apache All in One, Fourth Edition, 2008.
- 8 PHP Security Consortium. PHP Security Guide 1.0,
13. huhtikuuta 2010 [online, pdf]
<http://phpsec.org/php-security-guide.pdf>

Autovuokraamo Matka-Rent Oy

Ahjokatu 12, 40320 Jyväskylä

P. 010 666 2030

Ajoneuvon vuokrasopimus**Vuokraaja**

Nimi	Aili Autonvuokraaja
Osoite	Vuokrakuja 1 A 2
Postinumero	00001
Postitoimipaikka	Helsinki
Puhelinnumero	0123456789
Henkilötunnus	123123-1234
Ajokortin antoaika	20.01.2000
Ajokortin antopaikka	Inari

Aika

Sovittu noutaika	20.05.2010 10:00
Noudettu	20.05.2010 8:49
Sovittu palautusaika	20.05.2010 17:00
Palautettu	

Lisätietoja

Vuokraaja ilmoittaa käyttävänsä autoa vain kotimaassa.

Mukana TomTom -navigaattori ja nokkakärryt.

Ajoneuvo

Merkki ja malli	Ford Transit
Tyyppi	Pakettiauto
Rekisterinumero	XXX-11
Polttoaine	Diesel
Polttoaine mittarin näyttämä	100%
Ajoneuvon kunto	Ennen vuokralleottoa virheetön.
Lähtö km	1252
Paluu km	

Hinta

Sovittu hinta	80.00 €
Ennakkomaksu	80.00 €
Palautetaan	
Lisämaksu	