

Lari Kettunen

Web-käyttöliittymän suunnittelu ja toteutus PerusTA2-työajanseurantajärjestelmään

Metropolia Ammattikorkeakoulu
Tietotekniikka
Ohjelmistotekniikka
Opinnäytetyö
3.5.2011

Tekijä Otsikko Sivumäärä Aika	Lari Kettunen Web-käyttöliittymän suunnittelu ja toteutus PerusTA2-työajanseurantajärjestelmään 31 sivua + 2 liitettä 3.5.2011
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Vanhempi ohjelmistosuunnittelija, DI, Jan Svedström FT, lehtori, Vesa Ollikainen
<p>Insinöörityön tarkoituksena oli suunnitella ja toteuttaa web-käyttöliittymä Tamtron Solutions Oy:n PerusTA2-työajanseurantajärjestelmään. Tavoitteena oli luoda helppokäyttöinen, visuaalisesti miellyttävä sekä selkeä järjestelmä, joka toiminnallisuudeltaan vastaa PerusTA2-pääteohjelmiston toiminnallisuutta lisäten siihen web-pohjaisuuden tuomat edut.</p> <p>Työn teoriaosuudessa tutustuttiin työajanseurantaan yleisesti ja esiteltiin sen lyhyt historia. Samoin perehdyttiin käytössä olevaan laitteistoon sekä yrityksen nykyiseen ohjelmistoon. Käytiin myös läpi työn tavoitteita ja toteutusympäristöä.</p> <p>Työn teknisessä osuudessa käsiteltiin hyvän Internet-sivuston suunnitteluun ja käytettävyyteen liittyviä tekijöitä sekä ASP.NET-tekniikan ominaisuuksia ja sen tarjoamia mahdollisuuksia. Lisäksi esiteltiin työn tuloksena toteutunutta järjestelmää ja pohdittiin jatkokehitysmahdollisuuksia.</p> <p>PerusTA2-webkäyttöliittymän kehitystä jatketaan insinöörityön tekemisen jälkeen ja tarkoituksena on, että siitä tulee asiakasystävällinen tuote PerusTA2-järjestelmän tueksi.</p>	
Avainsanat	ASP.NET, web-käyttöliittymä, työajanseuranta

Author	Lari Kettunen
Title	Web User Interface for PerusTA2 Time Attendance System
Number of Pages	31 pages + 2 appendices
Date	3 MAY 2011
Degree	Information and Communications Technology
Degree Programme	Bachelor of Engineering
Specialisation option	Software Engineering
Instructor(s)	Sr. Software Engineer, M.Sc, Jan Svedström Ph.D., lecturer, Vesa Ollikainen
<p>The purpose of this thesis was to design and implement a web user interface for Tamtron Solutions PerusTA2 -time attendance system. The aim was to create an easy-to-use, efficient and usable system, which is comparable to PerusTA2-terminal software in functionality adding the benefits of web environment.</p> <p>The theory of the study presents the principals of time attendance systems and a brief history of the subject. It also shows the used equipment as well as the company's current repertoire. It also discusses the main objectives and the implementation environment.</p> <p>The technical section presents the principals of good web site design and usability. It also discusses the capabilities of ASP.NET technology and its opportunities. In addition, the actual results of the thesis are presented and the possibilities for further development are considered.</p> <p>The development of PerusTA2 web user interface will be continued after the conclusion of this thesis and it is intended to become a customer-friendly product and support for the PerusTA2 system.</p>	
Keywords	ASP.NET, Web user interface, Time and attendance

Sisällys

1	Johdanto	1
2	Tausta	2
2.1	Työajanseurantajärjestelmien historiaa	2
2.2	Työajanseurantatuotteet ja -ratkaisut	3
2.3	Tamtron Solutions Oy:n nykyinen järjestelmä	5
3	Työn tavoitteet	7
3.1	Tehtävänasettelu ja tutkimuskysymykset	7
3.2	Toteutusympäristö ja muut reunaehdot	8
4	Käyttöliittymän suunnittelu	8
4.1	Käytettävyys ja hyvä käyttöliittymä	8
4.2	Merkkauskielet	13
4.3	Validointi	14
4.4	ASP.NET	15
4.5	Navigointi	17
4.5.1	Master Pages	18
4.5.2	Site Map	19
4.5.3	Div	19
4.5.4	Tree View	19
4.6	CSS-tyylisivu	20
4.7	Tietolähteet	22
4.7.1	GridView	23
4.7.2	DetailsView	23
4.7.3	FormView	23
5	Rakenne	24
5.1	Henkilöt	26
5.2	Erikoispäivät	26
5.3	Työvuorot	27
5.4	Pääte	27
5.5	Lisät	27
5.6	Tilat	27

5.7	TA-Tiedot	28
6	Johtopäätökset	28
6.1	Yhteenveto	28
6.2	Jatkokehityssuunnitelmat	29
	Lähteet	32
	Liitteet	
	Liite 1. Esite: T900-tiedonkeruupääte	
	Liite 2. Esite: RFID-tunnisteet	

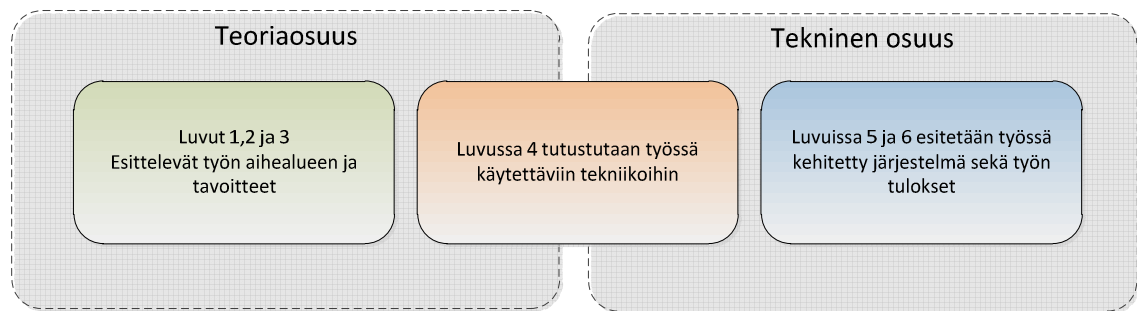
1 Johdanto

Tämän insinööriyön tarkoituksena on suunnitella ja toteuttaa web-käyttöliittymä Tamtron Solutions Oy:n PerusTA2-työajanseurantaohjelmaan. Työ päädyttiin toteuttamaan Microsoftin Visual Studio 2010-kehitystyökaluilla ja ASP.NET-teknologialla, sillä näin ohjelma pysyy yhtenäisenä yrityksen muiden ohjelmien kanssa. Työtä toteuttaessa on tarkoitus panostaa hyvään käytettävyyteen ja toteuttaa asiakkaan näkökulmasta mahdollisimman selkeä, helppokäyttöinen ja graafisesti miellyttävä järjestelmä.

PerusTA2 on Tamtron Solutions Oy:n kehittämä työajanseurantaohjelma, jonka tarkoituksena on korvata kellokortit sähköisellä järjestelmällä. Järjestelmä koostuu ohjelman lisäksi tietokannasta, leimauspäätteestä sekä kortinlukijasta, jolla työntekijä voi leimata itsensä esimerkiksi sisään tai ulos. Ohjelma on kehitetty Microsoftin Visual Studio 2010 -työkaluilla C#-ohjelmointikielellä, se käyttää mysql-pohjaista Unify SQLBase-tietokantaa.

Tamtron Solutions Oy on vuonna 1978 perustettu yritys, joka toimittaa, valmistaa ja kehittää kulunvalvonta-, työajanseuranta- sekä tiedonkeruulaitteita. Yhtiö on toimintansa aikana toimittanut lukuisia järjestelmiä kotimaahan sekä vientiin ja näin vakiinnuttanut asemansa alan ammattilaisena. Yhtiö on osa Tamtron-konsernia ja sen toimipiste sijaitsee Espoon Westendissä.

Insinööriyön alussa luvussa 2 esitellään työajanseurantajärjestelmien taustaa. Siinä käydään lyhyesti läpi työajanseurantajärjestelmien historia sekä perehdytään Tamtron Solutions Oy:n tämän hetkiseen laitteistoon ja järjestelmään työhön liittyviltä osilta. Kolmannessa luvussa perehdytään työn tavoitteisiin ja tutkimusongelmaan, mikä tässä tapauksessa on hyvän web-pohjaisen järjestelmän suunnittelu ja kehittäminen työpöytäohjelman rinnalle. Neljännessä luvussa tutustutaan ASP.NET-tekniikkaan ja käydään läpi työssä käytettävät menetelmät sekä perustellaan niiden käyttö. Luku 5 käsittelee työn tuloksena syntyneen järjestelmän toimintaa ja esittelee sen ulkoasun. Kuudennessa luvussa pohditaan työn tuloksia ja esitetään siinä ilmenneitä jatkokehitysmahdollisuuksia PerusTA2-webkäyttöliittymään liittyen. Kuvassa 1 on esitelty työn rakenne ja jaottelu.



Kuva 1 Kaavio insinööriyön rakenteesta

2 Tausta

2.1 Työajanseurantajärjestelmien historiaa

Työajanseurantajärjestelmiä käytetään nykyään kaikenlaisissa ja -kokoisissa yrityksissä. Niiden historia kuitenkin alkaa teollisuudenalan yrityksistä, joissa työntekijöiden palkka maksettiin tehtyjen tuntien mukaan, joten oli tärkeää tietää henkilöiden tehdyt tunnit. Työntekijöiden määrä saattoi kuitenkin olla suuri, joten oli vaikeaa seurata, mihin aikaan kukin työntekijä oli saapunut tai lähtenyt [5]. Näin ollen piti kehittää järjestelmä aikojen seuraamiseen.

Suomessa työajanseurantaan käytettiin entisaikaan kiilan muotoista puukapulaa, nimeltä päiväpulkka tai pulikka. Puukapuloita oli kaksi, joista toinen oli työntekijöillä ja toinen työnjohtajilla. Päivän päätteeksi työnjohtaja asetti pulikat vastakkain ja veisti niihin yhtä aikaa loven. Näin varmistettiin, että työntekijät tekivät työpäivänsä loppuun asti ja kuun lopussa tai urakkatyön loputtua palkka maksettiin pulikassa olevien lovien mukaan. [15].

Puukapulat kuitenkin kävivät nopeasti melko epätarkoiksi, joten otettiin käyttöön kellokortit. Kellokortti oli jäykkä paperinpala, jonka työntekijä asetti leimauslaitteeseen töihin tullessaan ja sieltä lähtiessään. Laite leimasi korttiin työntekijän saapumis- ja lähtöajan. Yrityksissä olivat erikseen palkanlaskijat, jotka laskivat työntekijöiden palkat kellokortissa olevien tuntien perusteella. Koska työ hoidettiin käsin, saattoi virheitä tulla helposti, joka johti väärin palkkojen maksuun. [5.]

Työajanseuranta vaikenä entisestään, kun vuoro- ja ylityöt sekä työajan liukumat alkoivat yleistyä yrityksissä. Niinpä sähköiset järjestelmät alkoivat syrjäyttää mekaanisia laitteita ja käsintehtyä laskentaa. Sähköinen järjestelmä käyttää sähköisiä tunnisteita, joita voivat olla esim. viivakoodi, magneettinauha tai sormenjälki perinteisen paperinpalan sijaan. Tällaisesta tunnisteesta aikatiedot voidaan lukea sähköisesti suoraan järjestelmään, tietokone voidaan määrittää tekemään laskelmat automaattisesti. [5.]

Tietokoneen käyttäminen työajanseurannassa mahdollisti tarkat tulokset sekä monimutkaiset mahdollisuudet seurata työntekijöiden työaikoja. Lisäksi palkanlaskenta nopeutui huomattavasti tietokoneiden ansiosta ja palkanlaskentahenkilökuntaa tarvittiin vähemmän. Myös ylläpitokustannukset olivat isoja mekaanisia laitteita huomattavasti pienemmät. [5.]

Kuitenkin isoin muutos työajanseurantajärjestelmissä lienee Internetin tuomat mahdollisuudet. Web-pohjainen seurantajärjestelmä mahdollistaa, että erillisiä laitteita ei välttämättä tarvita, vaan työntekijät voivat leimata mistä tahansa web-selaimen sisältävältä laitteelta. Tämä antaa hyvät mahdollisuudet työn tekemiseen esim. kotoa tai ulkomailta käsin. Lisäksi web-pohjaiset järjestelmät ovat yleensä helppokäyttöisiä kaikille Internetin käyttöön totuneille. Kokonaisuudessaan web-pohjaisuus mahdollistaa joustavan ja periaatteessa rajoituksettoman työajanseurannan.

2.2 Työajanseurantatuotteet ja -ratkaisut

Tamtron Solutions Oy:llä on yli 25 vuoden kokemus työajanseuranta- ja tiedonkeruulaitteiden valmistamisessa ja kehittämisessä. Yrityksen tavoitteena on hyödyntää uusinta tekniikkaa, jotta saavutettaisiin parhaat mahdolliset, eli tarkat, luotettavat, turvalliset ja suorituskykyiset tulokset. Toimivan työajanseurannan toteuttamiseksi tarvitaan jokaiselle työntekijälle henkilökohtainen tunniste sekä leimausalue, jossa tietoja voidaan lukea ja kirjata.

T900 on Tamtron Solutions Oy:n tiedonkeruupääte, jota voidaan käyttää tiedonkeruuta, työajanseuranta- ja kulunvalvontakäyttöön. Kuvassa 2 on esitelty T900 ulkonäkö, jossa pyörii PerusTA-ohjelmisto. Laite on teollisuuskäyttöön suunniteltu paneeli-PC, jossa

pyörii Windows Embedded Standard-käyttöjärjestelmä. Näin ollen päätteessä voidaan ajaa Windows-ympäristön ohjelmia, myös .NET-ympäristö on tuettu.



Kuva 2 T900-päätte

T900-laite on varustettu 8,4 tuuman resistiivisellä kosketusnäytöllä, joka on sen ensisijainen ohjaustapa. Laitteessa on kuitenkin myös kaksi USB-liitäntää, joten siihen voidaan tarvittaessa liittää esimerkiksi näppäimistö. Laitteessa on myös kaksi RS-232-sarjaporttia, joihin voidaan liittää tiedonkeruulaitteita, kuten RFID-tunnistekortinlukija. Liitteessä 1 on esitelty laitteen tarkemmat tiedot ja ominaisuudet.

RFID eli Radio Frequency Identification on etätunnistusmenetelmä, joka perustuu radiotaajuuksiin. Se koostuu RFID-tunnisteesta eli saattomuistista, lukijasta ja antennista. RFID-tunnisteeseen on tallennettu yhdeksännumeroinen ID-numero, joka mahdollistaa suuren määrän erilaisia numeroyhdistelmiä. Tieto välitetään lukijalle radiotaajuuksiin tunnistekortissa sijaitsevan antennin välityksellä. Etätunnistus toimii siten, että ID-numero ohjelmoidaan työajanseurantaohjelmaan. Käyttäjän näyttäessä tunnistetta lukijaan ohjelma osaa yhdistää järjestelmässä olevat tiedot ID-numeron perusteella. Kuvassa 3 esitellään RFID-pisaratumiste sekä RFID-lukija. Liitteessä 2 on kerrottu tar-

kemmin RFID-tunnisteista ja esitellään Tamtron Solutions Oy:ssä käytössä olevia tunnisteita.



Kuva 3 RFID-lukija ja RFID-pisaratunniste

2.3 Tamtron Solutions Oy:n nykyinen järjestelmä

Tamtron Solutions Oy:n nykyinen järjestelmä PerusTA2 on suunniteltu itsenäiseksi ja käytettävyydeltään yksinkertaiseksi. Ohjelma mahdollistaa siihen määriteltyjen leimausten tekemisen kulkukortilla sekä myös asetusten hallinnan. Ohjelman käyttöliittymä koostuukin kahdesta osasta: leimausnäkyvästä ja hallintasovelluksesta. Leimausnäkyvä on oletuksena esillä ja siitä voidaan suoraan siirtyä hallintanäkymään. Järjestelmään kuuluu myös Unify SQLBase 11.5 -pohjainen tietokanta, jonne leimaukset ja asetukset tallennetaan, sekä Tamtron Xs Server -palvelinohjelmisto. Kuvassa neljä on esitelty PerusTA2:n hallintanäkymä Tamtron Solutions Oy:n T900-päätelaitteella.



Kuva 4 PerusTA2-hallintanäkymä T900-päätteellä

Hallintasoovellus mahdollistaa järjestelmän tietojen selaamisen ja muokkaamisen. Sovelluksella voidaan muun muassa muuttaa henkilöiden, työvuorojen, erikoispäivien ja lisen asetuksia. Lisäksi sillä voidaan tehdä leimauksia manuaalisesti ja selailla työntekijöiden tilatietoja ja laskea saldoja. Tiedot voidaan myös tarvittaessa tulostaa verkkotulostimeen. Sovellus on jaettu osakokonaisuuksiin aihepiireittäin näkymien perusteella. Ohjelman käyttö on suunniteltu siten, että se onnistuu täysin kosketusnäytöltä. Järjestelmä on myös itsenäinen siinä suhteessa, että kaikki sen tarvitsevat komponentit pyörivät tarvittaessa yhdellä päätelaitteella.

Järjestelmän tietokanta on 27-tauluinen relaatiokanta. Tietokantaan on tallennettu kaikki järjestelmän asetukset sekä ohjelman kautta kirjattavat tiedot. Tietokannan tärkeimmät taulut lienevät henkilö-, yritys- ja työpäivä-taulu. Taulut ovat linkitettyjä toisiinsa Id-avaimen perusteella, jolloin esimerkiksi henkilön Id:n mukaan asetetaan hänelle kulkukortti, yritys, osasto sekä työvuoro, kunkin taulun oman Id:n perusteella.

3 Työn tavoitteet

3.1 Tehtävänasettelu ja tutkimuskysymykset

Tämän työn tarkoituksena on toteuttaa web-käyttöliittymä Tamtron Solutions Oy:n PerusTA2-järjestelmään. PerusTA2-järjestelmä on itsessään toimiva ratkaisu, mutta pieni näyttö rajoittaa laitteen käytettävyyttä. PerusTA2 järjestelmän hallintosovellusta käytetään useimmiten kosketusnäytöltä. Näin ollen asetusten teko ei välttämättä ole niin nopeaa kuin perinteisesti hiirellä ja näppäimistöllä. Järjestelmän käyttö on myös sidottu itse laitteeseen. Web-käyttöliittymä mahdollistaa asetusten tekemisen ja tietojen selailun miltä tahansa internetiin kytketyltä työasemalta, jolloin käyttö on nopeampaa ja tarkempaa.

Internetin yli tehty leimaus on myös paikasta riippumaton, eli se mahdollistaa järjestelmän käyttämisen myös etätöissä. Lisäksi se vähentää kustannuksia sillä päätelaitteiden määrää voi vähentää. Web-suunnittelu kuitenkin tuo lisää haasteita juuri siitä syystä, että erilaisia käyttöympäristöjä on monia. Sovellusta voidaan käyttää monen kokoisilta laitteilta aina kännyköistä ja taulutietokoneista työpöytäkoneeseen. Lisäksi internet-selaimia on monia (Internet explorer, Firefox, Chrome jne.), joilla kaikilla sivuston pitäisi näkyä oikein. Resoluutio on siis tärkeä osa suunnittelussa, sillä sivuston pitäisi näyttää järkevältä pienillä ja suurilla tarkkuuksilla.

PerusTA2 tarjoaa kattavat mahdollisuudet työajanseurantaan liittyen. Web-käyttöliittymän tulisi mahdollistaa samat toiminnot tuoden lisäksi web-ympäristön edut. Tamtron Solutions Oy:n asiakaskunta on laaja ja asiakkaiden tietotekniset taidot vaihtelevat suuresti. Tässä työssä pyritään selvittämään minkälainen on käyttäjän kannalta käytettävä ja hyvä käyttöliittymä. Sitten esitellään menetelmät ja tekniikat joilla nämä periaatteet saataisiin mahdollisimman hyvin hyödynnetyksi valmiiseen tuotteeseen.

Tarkoituksena on, että näin saataisiin asiakasläheinen, näyttävä ja kattava web-käyttöliittymä toteutettua PerusTA2-järjestelmän tueksi.

3.2 Toteutusympäristö ja muut reunaehdot

Tässä työssä valmistuvan web-käyttöliittymän toteutus tehtiin Tamtron Solutions Oy:n tiloissa ja välineillä. Työ tehtiin Visual Studio 2010 -ympäristössä, joka tarjoaa kattavat mahdollisuudet web-järjestelmien kehittämiseen luontevasti mitä näet, sitä saat – periaatteella. Apuna minulla oli käytössä aiheeseen liittyvää kirjallisuutta ja internetistä saatavilla olevat kattavat ASP.NET-aiheiset videot. Myös ohjaajaltani ja Tamtron Solutions Oy:n muista järjestelmistä sain tarvittaessa hyvin apua ja vihjeitä työn tekemiseen.

Lopullisena tarkoituksena on toteuttaa toimiva, näyttävä ja helppokäyttöinen web-käyttöliittymä PerusTA2-ohjelmistolle, jossa on sama toiminnallisuus kuin itse työpöytäohjelmistossa. Tämän työn aikarajoitteiden johdosta kuitenkin ensisijaisena tavoitteena on toteuttaa hyvin suunniteltu pohja täydelle järjestelmälle, joka täyttää käyttöliittymälle asetetut kriteerit. Tarkoituksena on toteuttaa osa toiminnallisuudesta ja jatkossa kehittää järjestelmä valmiiksi tuotteeksi. PerusTA2 ja tässä työssä siihen kehitettävä web-käyttöliittymä ovat Tamtron Solutions Oy:n kaupallisia tuotteita. Tästä syystä tässä dokumentissa ei voida julkaista liian yksityiskohtaista tietoa tuotteista vaan painopiste pidetään käyttöliittymän suunnitteluun ja toteutukseen liittyvissä tekijöissä.

4 Käyttöliittymän suunnittelu

4.1 Käytettävyys ja hyvä käyttöliittymä

Käyttöliittymällä tarkoitetaan sitä ohjelman osaa, jolla käyttäjä on vuorovaikutuksessa ohjelman kanssa. Tämä sisältää käytännössä kaiken, mitä tapahtuu ihmisen ja tietokoneen välillä, eli käyttäjän tietokoneelle syöttämät syötteet ja samoin myös tietokoneelta käyttäjälle vastineeksi tuleva informaatio [9, s.1]. Käyttöliittymän suunnittelu on siinä mielessä hankalaa, että siinä tulee ottaa huomioon kolme tieteenalaa: tietotekniikka, kognitiotieteet ja graafinen suunnittelu. Kognitiotieteillä tarkoitetaan tässä sitä, miten

ihminen käsittelee esitettyä tietoa ja miten hyvin ulkoasu on toteutettu muistettavuuden ja loogisuuden kannalta [9, 2.2].

Käyttöliittymä on tärkeä osa mitä tahansa järjestelmää. Käyttöliittymä on se osa tuotteesta, joka on asiakkaille näkyvässä ja jonka kautta kaikki ohjelman mahdollistamat toiminnot suoritetaan. Hyvin toteutettu käyttöliittymä tekee ohjelmasta helppokäyttöisen ja jättää ohjelmasta kokonaisuudessaan positiivisen mielikuvan. Tosin paraskaan käyttöliittymä ei voi pelastaa kelvotonta ohjelmaa ja päinvastoin huonolla käyttöliittymällä varustetulla, mutta muuten kattavalla ohjelmalla käyttäjä ei saa mitään aikaiseksi. Yleisesti ottaen käyttöliittymä on juuri se tekijä, jonka perusteella käyttäjä valitsee mitä ohjelmaa tai järjestelmää hän haluaa käyttää, jos valinnanvaraa on. Käyttöliittymän arvostus on myös kasvanut aikojen saatossa ja nykyisin asiakkaat vaativat helppokäyttöisyyden lisäksi, että tuote on myös hauska ja trendikäs. [9, s. 4.]

Hyvän käyttöliittymän suunnitteluun kannattaa siis panostaa. Kun halutaan toteuttaa hyvä käyttöliittymä tulee ottaa huomioon ensisijaisesti ryhmä, joka sitä tulee loppuvaiheessa käyttämään. Tässä työssä toteutettava käyttöliittymä tulee luultavasti käyttöön hieman pienempiin ja aloiltaan erilaisiin yrityksiin, joissa ohjelman käyttäjien taso saattaa vaihdella suuresti. Näin ollen ohjelman suunnittelussa otetaan huomioon, että se on helppokäyttöinen myös aloittelijan kannalta. Aloittelijalla tässä tarkoitetaan sitä, että henkilöllä ei ole kovinkaan paljon kokemusta tietotekniikasta. Näin ollen käyttöliittymän pitäisi siis olla mahdollisimman helposti opittava ja yksinkertainen ja tarjota selkeää palautetta sekä tarkat kuvaukset mahdollisista virheistä. [10.]

Yleisesti ottaen parhaimman mahdollisen käyttöliittymän saa toteutettua vain tekemällä jonkinlaisia käyttäjätestauksia. Kuten sanottu, tämän työn kohderyhmä on kuitenkin hyvin laaja ja sisältää eritasoisia henkilöitä eriasteisista tietotekniikan taitajaryhmistä. Samoin käyttäjien ikäryhmä saattaa vaihdella suuresti. Rozell & Gardnerrin vuonna 2000 tekemän tutkimusten mukaan ikä ei kuitenkaan ole niin suuri tekijä käytettävyyttä mitatessa kuin järjestelmien käyttökokemus [9, s.8]. Ikä kuitenkin saattaa tuoda muita esteitä, esimerkiksi näön ja motoriikan huonontuminen, jotka tässä työssä pyritään ottamaan huomioon mm. tekstin väri- ja kokosuunnittelussa. Käyttöliittymä kuitenkin tulee luultavasti käyttöön vain suomalaisiin yrityksiin, jolloin kulttuuritekijöihin, kuten esimerkiksi titteleihin ja numero- tai aikaformaatteihin ei tarvitse keskittyä. Kuitenkin

kohderyhmä on sen verran laaja ja aikataulu ja resurssit suppeita, että tässä työssä pyritään saamaan optimaalinen tulos ilman varsinaista käyttäjätestausta.

Käyttöliittymän visuaalisuus on käytännöllisyyden ohella tärkein osa käyttöliittymän suunnittelussa. Graafisesti hyvin suunniteltu käyttöliittymä tuo ohjelman käyttöön mukavuutta ja helppoutta ja näin tekee kokonaisuudessaan käyttämisestä mielekkäämpää. Graafisella suunnittelulla tarkoitetaan periaatteessa elementtien ulkoasua ja niiden sommittelua kuvaruudulla. Hyvän graafisen suunnittelun ominaisuuksia ovat järjestelmän esteettinen kokonaisuus ja yhdenmukaisuus eli värit, linkit, otsikot, valikot sekä muut komponentit ovat samanlaisia ympäri käyttöliittymää. Toisaalta muunneltavuus, eli se että käyttäjä pystyy itse vaikuttamaan esimerkiksi väreihin ja kirjasimiin oman mieltymyksensä mukaan on myös hyvä ominaisuus, kunhan vain yhdenmukaisuus säilyy. [10.]

Käyttöliittymän muistettavuus on myös tärkeä osa suunnittelua. Periaatteessa tämä tarkoittaa sitä, että jos käyttäjä ei näe kaikkia tarvitsemiaan toimintoja samaan aikaan ruudulla, tulee hänen turvautua muistiinsa siltä kohdin, mitä vaihtoehtoja sivun muissa kohdissa oli. Tämä vaikeuttaa käyttämistä ja altistaa virheille. Tämän työn suunnittelussa on lähdetty siitä, että näkyvissä on aina tarvittavat elementit navigoimiseen ja yksittäiset sivut on pyritty osioimaan siten, että sisältöä on sopivasti. Jos järjestelmää käytetään tarpeeksi pieneltä näytöltä, esim. kännykällä niin muistettavuusongelma saattaa ilmetä. Sivustosta ei ole erikseen tarkoituksena toteuttaa mobiiliversiota, mutta lukuun ottamatta pienen näytön aiheuttamia ongelmia, pitäisi järjestelmän olevan käytettävissä myös matkapuhelimella ja erilaisissa tablet-tietokoneissa onkin jo riittävän iso resoluutio järjestelmän sujuvaan käyttöön. Suunnittelussa on lähdetty siitä, että toiminnot voi tehdä hiiren tai näppäimistön lisäksi myös erilaisilta kosketusnäytöiltä.

Hyvään käyttöliittymään kuuluu myös virheiden mahdollisuuden vähentäminen sekä palautteen antaminen käyttäjän toimintaan. Virheitä voidaan vähentää tarjoamalla käyttäjälle mahdollisuus peruuttaa kaikki tekemänsä muutokset tai vaihtoehtoisesti ilmoittaa, jos joku muutos on peruuttamaton. Samoin jos joku toiminto ei sattumoisin ole käytössä, pitäisi sen valinta tehdä mahdottomaksi. Jos järjestelmä tekee jotain aikaa vievää toimenpidettä, siitä tulisi antaa käyttäjälle ilmoitus, niin hän ymmärtää että jotain tapahtuu, eikä esimerkiksi ala satunnaisesti painelemaan nappeja tai linkkejä.

Käyttöliittymäelementtejä valittaessa on hyvä ottaa huomioon elementtien tunnettavuus. Samankaltaisuus muiden järjestelmien kanssa helpottaa käyttöä, koska elementit ja niiden toiminnallisuus ovat käyttäjälle tuttuja jo ennestään [10]. Tästä syystä tässä työssä on valikko toteutettu Windowsista tuttu kansiorakenne. Samoin erilaisissa valikoissa ja nappuloissa käytetään Windowsista tuttua ulkoasua.

Käyttöliittymän jäsentelyssä on tärkeää, että sen rakenne on looginen ja yhtenäinen. Eli siis elementit pysyvät mahdollisimman muuttumattomina riippumatta sivusta. Kuvassa 5 on esitelty sivuston yleisrakenne. Otsikko-osa on aina samanlainen. Samoin valikko- ja sivukartta-osiot. Muuttuva tieto -osio on ainoa alue, jossa tiedon esittämistapa vaihtelee. Siinä esiteltävä informaatio riippuu siitä mihin käyttäjä on valikon kautta navigoinut. Kuitenkin myös näiden sisältösivujen ulkoasu on pyritty keskenään pitämään yhtenäisenä käyttämällä samankaltaista rakennetta sekä valikoita.



Kuva 5 Käyttöliittymän jäsentely

Järjestelmän tekstin jäsentely on tärkeää luettavuuden kannalta. Yleisesti ottaen ihmiset eivät jaksa lukea hirveän pitkiä tekstejä, joten tässä työssä on tarkoitus antaa tarvittava informaatio lyhyesti ja ytimekkäästi. [10]. Samoin otsikot on eritelty selkeästi sekä kirjasimen kooksi valittu riittävän suuri koko. On myös pyritty siihen, että samantyyppiset asiat on ryhmitelty sivulle asiayhteyden mukaan ja eroteltu otsikoinnilla. Tässäkin kohtaa on tärkeää yhtenäisyys, eli ryhmittely ja tekstin jäsenitys pysyvät samantyyppisenä järjestelmän sivusta riippumatta.

Käyttöliittymän värit ovat myös tärkeä osa käyttöliittymän suunnittelua. Hyvällä värien käytöllä saavutetaan käyttötehokkuuden ja käytön miellyttävyyden kasvaminen. Väreillä voidaan korostaa sivun rakennetta ja helpottaa sivulla käytettävien elementtien erottamista toisistaan. Lisäksi väreillä voidaan helpottaa navigointia, sillä niiden avulla voidaan ilmaista, millä sivulla henkilö on jo käynyt ja millä hän on tällä hetkellä. Värien suhteen pitää kuitenkin olla myös varovainen, sillä esimerkiksi liiallisella värien käytöllä voidaan käyttöliittymästä tehdä sekava ja vaikeakäyttöinen. Onkin suositeltavaa, että sovelluksessa käytettäisiin maksimissaan seitsemää eri väriä. [10]. Vähäinen värien määrä antaa käyttöliittymästä tasapainoisen kuvan, joka on tämänkaltaisessa järjestelmässä tavoiteltavaa. Värien valinnassa on otettu myös huomioon taustojen värityksi niin, ettei se häiritse tekstin lukemista. Käytännössä käytetään siis otsikkotasolla tummaa taustaa ja valkoista tekstiä ja itse sisällön ja valikon kohdalla valkoista taustaa, jotta luettavuus pysyy hyvänä. Tekstin ja muiden elementtien pitäisi olla hyvin hahmotettavissa myös erilaisilla näytöillä sekä kirkkaus- ja kontrasti-asetuksilla.

Käyttöliittymää voidaan piristää ja havainnollistaa käyttämällä avuksi kuvia. Kuva saattaa sisältää huomattavasti enemmän tietoa kuin pelkkä teksti ja parantaa muistettavuutta, jos kuvia voidaan yhdistää reaalielämän tapahtumiin. Esimerkiksi jos kulkukortti-sivulle laitetaan linkin viereen kuva oikeannäköisestä kulkukortista, on kohteen sisältö helpompi hahmottaa. Kuvassa 6 on esitelty ohjelmassa käytettäviä ikoneita, joiden tarkoitus on selkeyttää ja havainnollistaa järjestelmän navigointia. Ikonit on tarkoitettu näyttämään pienenä navigointipuussa ja isompina versioina jokaisen kansio-tason etusivulla. Myös kansio-taso on helposti havaittavissa siinä esitettävän kansio-ikonin johdosta. Ikonit on suunniteltu niin, että ne ovat helposti yhdistettävissä reaalielä-

maan, esimerkiksi henkilö-sivua kuvaa henkilö ja kulkukorttia vastaavasti kuva kulkukortista.



Kuva 6 Esimerkkejä järjestelmässä käytettävistä ikoneista

4.2 Merkkaukielet

Merkkaukielellä tarkoitetaan ympäristöriippumattomia standardeja, joilla informaatiota voidaan luoda, hallita ja siirtää esimerkiksi web-sivujen muodossa. Varsinaisista ohjelmointikielistä ne eroavat siten, että ne eivät itsessään suorita varsinaista tiedon prosessointia. Merkkaukielten ideana on erottaa dokumenttien rakenne, ulkoasu ja sisältö toisistaan. Tärkeimpiä merkkaukieliä ovat SGML (Standard Generalized Markup Language), HTML (Hypertext Markup Language) ja XHTML (eXtensible Hypertext Markup Language). SGML itsessään pohjautuu IBM:n kehittämään GML:ään (Generalized Markup Language), joka julkaistiin 1960-luvulla. SGML on nimetty hieman harhaanjohtavasti, sillä se on itse asiassa metakieli, jolla merkkaukieliä voidaan määritellä. [15.]

SGML on ISO-standardin mukainen tapa dokumentin merkkaukielen ja tunnistasettelun määrittämiseen. Siinä on ideana se, että dokumenteissa on rakenteellisia osia, joiden ulkoasu voidaan tulkita SGML:llä määriteltyjen sääntöjen mukaan. Tällaisen dokumentin lopullinen ulkoasu määräytyy käytettävän laitteen tai ohjelman mukaan. Tästä on etuja esimerkiksi siinä mielessä, että dokumentin suunnittelussa voidaan keskittyä rakenteen suunnitteluun ulkoasun sijasta. Näin ulkoasu voidaan suunnitella erikseen esimerkiksi erilaisia medioita ajatellen. Tämän insinööriyön kannalta SGML on tärkeä siksi, että nykyään tunnetummat merkkaukielet HTML ja XHTML ovat määritellyt SGML:n mukaan. [13.]

HTML on merkkaukielistä ehkä tunnetuin. Se on avoimesti standardoitu kuvauskieli, jolla voidaan kuvata web-sivuja. HTML-koodi koostuu elementeistä, joita voidaan kirjoittaa sisäkkäin ja peräkkäin. Internet selaimet käyttävät näitä elementtejä teknisinä ohjeina, joiden mukaan sivu jäsennetään. HTML tukee tekstin rakenteen merkitsemistä. Esimerkiksi teksti voidaan määritellä mm. otsikoksi tai leipätekstiksi. Rakenteellinen merkkkaus ei periaatteessa ota kantaa tekstin ulkoasuun, mutta usein selaimiin on asetettu jonkinlaiset oletusarvot tietynlaisten elementtien esittämiseen. HTML kuitenkin mahdollistaa myös elementin ulkoasun muuttamisen. Esimerkiksi teksti voidaan lihavoida muuttamatta sen sisältöä tai rakennetta. HTML mahdollistaa myös hypertekstuaaliset merkkaukset, jotka mahdollistavat dokumenttien linkityksen toisiin dokumentteihin. [13.]

HTML on yleisesti ottaen kaikkien erilaisten internet-selaimien tukema. Siitä on kuitenkin vuosien saatossa tullut monia eri versioita, joiden tuki vaihtelee. HTML:n ensimmäinen versio julkaistiin vuonna 1991 ja se kattoi 22 elementtityyppiä. Tällä hetkellä uusin versio on HTML 5, joka sisältää monia uusia ominaisuuksia esimerkiksi multimediaan liittyen. HTML 5:ssä on merkkaukielten tulevaisuus, sillä XHTML 2.0:n kehitys on lopetettu ja sen kehittäjät työskentelevät HTML 5:n parissa. [14.]

XHTML suunniteltiin alun perin HTML:n korvaajaksi ja se on web-sivujen merkintäkieli, joka täyttää XML:n muotovaatimukset. Vanhasta ja web-käyttöön ehkä turhan raskaasta SGML:stä luovuttiin siis XHTML:n myötä ja tilalle tuli XML. Tästä johtuen myös kielten syntaksit eroavat hieman toisistaan. Näkyvin ero lienee se, että XHTML:ssä elementit kirjoitetaan pienellä, kun HTML:ssä kirjainkoko ei ole määritelty. XHTML:n myötä myös yksiosaiset elementit, kuten esimerkiksi `
`, vaativat lopetuksen toisin kuin HTML:ssä. Muitakin eroja kielten väliltä löytyy, mutta periaatteessa XHTML on HTML:ää tiukempi.

4.3 Validointi

Validointi on toimenpide, jossa XHTML-dokumentista etsitään tekijän tai tekijän käyttämän editorin tekemät syntaksi- ja rakennevirheet XHTML-koodista. Validointi tehdään yleensä ohjelmallisesti ja sen on tarkoitus mahdollistaa virheiden korjaus ennen dokumentin varsinaista julkaisemista. Virheetön ohjelmakoodi on web-kehityksessä suota-

vaa vaikka sivu toimisikin kehittäjän mielestä oikein. Selaimet harvemmin antavat virheilmoituksia syntaksi- tai rakennevirheistä ja usein ulkoasu myös näyttää oikealta joten näitä virheitä on hankala havaita ilman sivuston validointia [16]. Visual Studion generoima XHTML ei välttämättä ole automaattisesti validia, mutta koodia käsin muokkaamalla voidaan tämäkin korjata.

4.4 ASP.NET

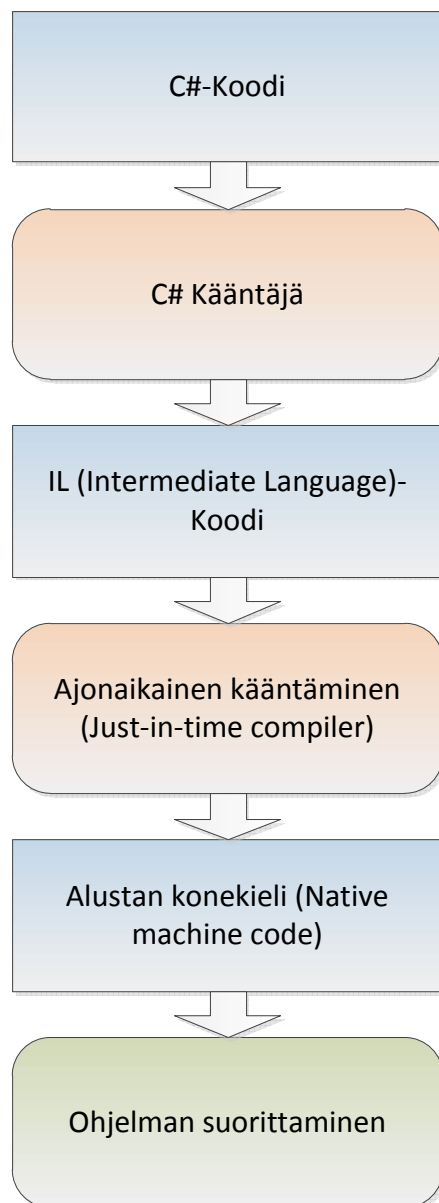
ASP.NET on Microsoftin kehittämä web-sovellusten kehitysympäristö, joka julkaistiin ensimmäisen kerran tammikuussa vuonna 2002 osana .NET frameworkin versiota 1.0. ASP.NET on jatkoa Microsoftin Active Server Pages eli ASP-tekniikalle ja perustuu siis .NET frameworkiin. Tekniikka mahdollistaa dynaamisten www-sivustojen, web-sovellusten ja web-palvelujen toteuttamisen. [1.]

ASP.NET on rakennettu Common Language Runtimeen (CLR) ympärille, joka mahdollistaa, että kehityksessä voidaan käyttää .NET-ympäristön ohjelmointikieliä C#, J# ja VB.NET. Se on palvelinpuolen ohjelmointimenetelmä, eli web-sivuihin kirjoitettu koodi suoritetaan palvelimella, jossa pyörii Microsoftin Internet Information Services- eli IIS-palveluohjelmisto. [1.]

ASP.NET toimii siten, että kun selain pyytää sivua palvelimelta, palveluohjelmisto tarkistaa, pyyntö html-sivua vai ASP.NET-sivua. Mikäli kyseessä on HTML-sivu, tiedot haetaan käyttöjärjestelmästä ja palautetaan selaimelle. Jos kyseessä on ASP.NET sivu, pyyntö siirretään ASP.NET runtimeen käsiteltäväksi, joka suorittaa tiedoston komentosarjat rivi riviltä. Kun prosessi on valmis, selaimelle palautetaan puhdas HTML-sivu. Tämä menetelmä mahdollistaa, että tekniikka on yhteensopiva lähes kaikkien selainten kanssa.

ASP.NET-sovellukset ovat aina käännettyjä. .NET-ohjelmat itse asiassa käännetään kaksi kertaa: ensiksi kirjoitettu C#-koodi käännetään välikieleksi, nimeltä Microsoft Intermediate Language (MSIL), tai lyhyemmin vain IL. Tällä saavutetaan .NET-tekniikan kieliriippumattomuus, sillä kehityskielestä riippumatta tässä vaiheessa syntyvä IL-koodi on aina samanlainen. Kääntämisen toinen vaihe tapahtuu juuri ennen sivun suorittamista. Tässä vaiheessa IL-koodi muutetaan matalan tason konekielelle. Vaihe tunne-

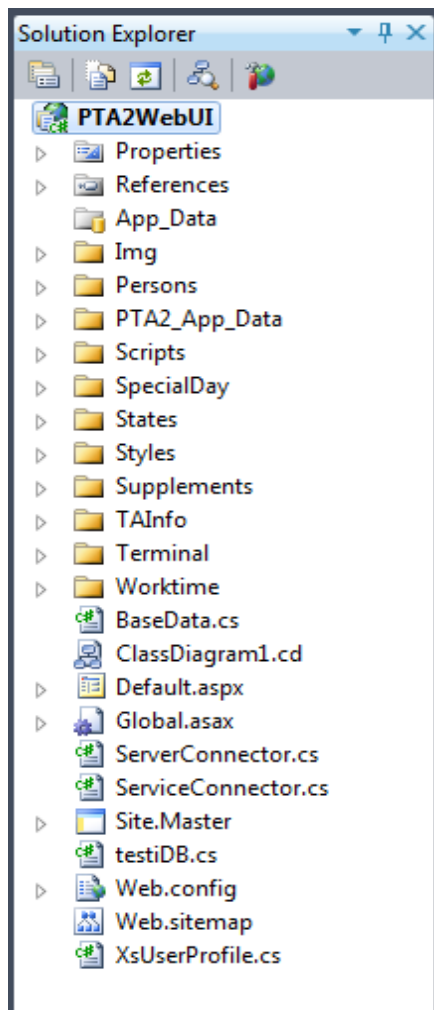
taan nimellä Just-in-Time (JIT) käänös, se tapahtuu samalla tavalla kaikille .NET-sovelluksille. Tämä on kuvattu kuvassa 7. [2 s.6].



Kuva 7 ASP.NET-sivun kääntämisen vaiheet

ASP.NET-tekniikan tärkeimpiin tiedostotyyppeihin kuuluu aspx, joka on varsinainen sovelluksen sivu ja sisältää palvelinpuolenkontrolleja. Master on MasterPages-tiedosto, joka sisältää web-sivuston yhtenäisen ulkoasun määrittäykset. Sitemap sisältää xml-kielillä kuvattua rakenteen web-sivusta. Cs-tiedostot ovat C#-ohjelmointikielillä tehtyjen luokkien lähdekooditiedostoja. CS-päätteisiä ovat myös Code Behind -tiedostot

sekä partial class –luokat. Kuvassa 8 on kuvattu tämän työn kansiorakenne niin kuin se näkyy Visual Studio 2010 –työkalussa.



Kuva 8 Hakemistorakenne

4.5 Navigointi

Navigointi, eli sivujen välillä liikkuminen, on tärkeä osa mitä tahansa web-järjestelmää. Tässä projektissa on lähtökohtana, että navigointi on selkeää, loogista ja helppokäyttöistä. Tämä saavutetaan pitämällä sivuston ulkoasu yhdenmukaisena ja rakenne loogisena sekä näyttämällä tarvittavat navigaatioon liittyvät peruskomponentit sen jokaisella sivulla. Näihin kuuluu navigointipaneeli, joka näytetään aina järjestelmän vasemmassa laidassa. Näin käyttäjän ei ole mahdollista mennä ”umpikujan” ja jokaiselta järjestelmän sivulta on pääsy sen mille tahansa muulle sivulle. Aina näkyvissä on myös sivupolku, josta näkee suoraan, missä kohtaa sivustoa ollaan milläkin hetkellä, sekä voidaan

palata sivuja taaksepäin. Myös alkusivulle voidaan palata järjestelmän kaikilta sivuilta. Yleisesti ottaen käyttäjä on aina tietoinen, missä kohtaa sivustoa hän on, mistä hän on tullut ja toivottavasti myös mihin hän voi jatkaa.

ASP.NET-ympäristö tarjoaa hyvät mahdollisuudet yhtenäisen ja helposti hallittavan navigointiratkaisun toteuttamiseen. Ratkaisu pohjautuu sivustosta tehtävään hierarkkiseen malliin, joka kuvataan xml-pohjaisessa sivukartassa. Tämä mahdollistaa keskitehtyn rakenteen toteuttamisen, jolloin yhtä tiedostoa muuttamalla voidaan muuttaa koko sivun navigointia. Itse sivustosta tehdään yhtenäinen MasterPages-ominaisuudella. Navigointiin liittyvistä ASP.NET:in ominaisuuksista Master Pages, Site Map ja Tree View kerrotaan tarkemmin seuraavissa luvuissa.

4.5.1 Master Pages

Master Pages on ASP.NET ominaisuus, joka mahdollistaa yhtenäisen www-sivuston toteuttamisen ja asettelun standardoimisen. Siinä on ideana, että www-sivustosta tehdään malli, johon voidaan määritellä kiinteää ja mukautettua sisältöä sisältäviä osia. Kun sama sivupohja on käytössä koko www-sivustossa, pysyy sivuston tyyli yhtenäisenä. Samoin tyylin muuttaminen onnistuu helposti, koska pääsivun määritelmiä muuttamalla muuttuvat kaikki sen alaisuudessa olevat sivut. [2 s. 720].

Master Pagesit näyttävät hyvin pitkälti tavallisilta ASP.NET web-lomakkeilta, ne voivat koostua HTML-koodista, web-kontrolleista tai C#-koodista. Erona on kuitenkin se, että sivupohja käyttää Master-direktiiviä Page-direktiivin sijaan, eli se näyttää tältä:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs" Inherits="PTA2WebUI.SiteMaster" %>
```

Toinen ero on se, että Master Pages yhteydessä voidaan käyttää ContentPlaceHolder –kontrollia, jolla tarkoitetaan sivuston aluetta, mihin voidaan lisätä sisältöä.

Content page, eli sisältösivu tarjoaa sisällön ContentPlaceHolder–kontrollille. ASP.NET linkittää Content-kontrollin sopivaan ContentPlaceHolder-kontrolliin täsmäävän tunnuksen perusteella. Tunnus määritellään ContentPlaceHolderID–ominaisuudessa. [2 s. 723]

4.5.2 Site Map

Master Page -mallilla sivustosta saadaan yhtenäisen näköinen, mutta koska sivusto koostuu useammasta sivusta, tarvitaan lisäksi navigointijärjestelmä. ASP.NET tarjoaa tähän joukon navigointiin liittyviä ominaisuuksia, jotka helpottavat navigoinnin toteuttamista. Site Map -ominaisuudella sivustosta tehdään xml-kielillä kuvattu sivukartta, josta selviää sivun nimi, polku ja paikka järjestelmässä. Kuvaus tallennetaan web.sitemap-tiedostoon. Sivukartan tiedot voidaan muuntaa käytettävään muotoon SiteMapDataSource- ja XmlSiteMapProvider -kontrolleissa.

4.5.3 Div

HTML:n div-tagin määrittelee osion HTML-dokumentissa. Sen rakenne muistuttaa paljon yksisoluista taulukkoa ja se mahdollistaakin sisällön sommittelun taulukoiden tapaan. Vanhempi tapa sivun rakenteen toteuttamiseen ovat kehykset, mutta ne aiheuttavat monia ongelmia ja taulukkoratkaisu ei sekään ole ongelmaton. Nykyään on siis kätevin tapa käyttää div-ryhmittelyelementtejä ja säästää taulukot tiedon esittämiseen. [11]. Divien määrittelyä voidaan kätevästi hallinnoida css-tyylisivun kautta ja oikeastaan diville itselleen määritellään ainoastaan sen sisältö.

4.5.4 Tree View

Tree View on ASP.NET-komponentti, jolla sivuston navigointi voidaan toteuttaa puumaisena listana. Puu koostuu solmuista, eli TreeNode-objekteista, joille voidaan määrittellä kuvaava teksti sekä kohde, jonne siirrytään solmua klikattaessa. Solmut voidaan erikseen määrittellä vanhemmaksi (parent) tai lapseksi (child). Näin voidaan toteuttaa monitasoinen rakenne, jossa puu voidaan näyttää kokonaan, tai sisältöä voidaan piilottaa tasojen taakse. Tree View -komponentin sisältö voidaan lukea xml-pohjaisesta sivukartasta, eli web.sitemap-tiedostosta. Komponentti mahdollistaa selkeän navigointivalikon toteuttamisen, jota on myös helppo muokata sivukarttatiedostosta käsin. [2 s. 775]

4.6 CSS-tyylisivu

CSS (Cascading Style Sheets) on tiedon kuvaamiseen tarkoitettu kieli, jonka avulla voidaan määritellä www-sivujen ulkoasu ja esitystapa [8]. Siinä on ideana, että sivuston varsinainen sisältö erotetaan sen ulkoasutyylillä. Periaatteessa tällä tarkoitetaan sitä, että HTML-kielellä kerrotaan, mitä sisältö on ja CSS määrittelee miltä sisältö näyttää.

CSS:n on kehittänyt World Wide Web Consortium (W3C), joka myös ylläpitää kielen määritelmiä. Sen tarkoituksena oli kehitysvaiheessa palauttaa www-sivujen standardinmukaisuus ja yleinen toimivuus erilaisilla selaimilla [8]. Nykyisin CSS-tekniikalla saadaan toteutettua yhtenäinen selaimesta ja käyttöjärjestelmästä riippumaton www-sivusto. Vielä tosin saattaa esiintyä selainkohtaisia bugeja, jotka haittaavat yhtenäisyyttä. CSS on nykyisin kieliperhe, johon kuuluvat tasot CSS1, CSS2 ja CSS3. Tässä työssä käytetään CSS2:n kehittyneempää versiota CSS2.1.

CSS-tyylisäännöt voidaan liittää HTML:n sekaan usealla eri tavalla. Tässä työssä käytetään erillisessä tiedostossa Site.css määriteltyä tyyliä, joka otetaan käyttöön Master Page -sivulla HTML:n title-elementin jälkeen laitettulla link-elementillä seuraavasti:

```
<title>PTA2WEBui</title>
<link href="~/Styles/Site.css" rel="stylesheet" type="text/css" />
```

Näin kyseinen tyyli on käytössä koko sivuston laajuisesti ja järjestelmän ulkoasu pysyy kauttaaltaan yhtenäisenä. Tärkeimpiä asioita, mitä CSS:n avulla tässä työssä määritellään ovat mm. tekstin ja taustojen väri sekä div-osioiden sijoittelu ja niiden koon määrittely. Määritellään myös fontin kirjasimet ja tyypit erikseen leipätekstille ja eritasoisille otsikoille. Tyyllisivulla määritellään myös marginaalit ja reunus, joka on käytössä esimerkiksi valikko-osion laidassa erottamassa valikon itse sisällöstä. Määritellään myös elementtien asettelu jos selaimen ikkunaa pienennetään niin, että kaikki sisältö ei normaalisti näkyisi sekä vierityspalkkien näkyvyys näissä tilanteissa.

Alla on listattu ote Site.css-tyylimäärittelyksestä, jossa asetetaan sivuston runko-osion ominaisuudet, kuten fontti ja taustaväri sekä myös linkkien ominaisuudet erilaisissa tilanteissa, kuten normaalitilanteessa ja hiirenkursori linkin päälle vietynä:

`body`

```

{
  background: #b6b7bc;
  font-size: .80em;
  font-family: "Helvetica Neue", "Lucida Grande", "Segoe UI", Arial, Helvetica,
  Verdana, sans-serif;
  margin: 0px;
  padding: 0px;
  color: #696969;
  overflow: hidden;
}

a:link, a:visited
{
  color: #034af3;
}

a:hover
{
  color: #1d60ff;
  text-decoration: none;
}

a:active
{
  color: #034af3;
}

p
{
  margin-bottom: 10px;
  line-height: 1.6em;
}

h1
{
  font-size: large;
  border-bottom: 2px dotted grey;
}

h2
{
  font-size: medium;
  color: #4c4c4c;
  margin-bottom: 0;
  padding-bottom: 0
}

```

Koodiesimerkki 1 CSS-määrittäjiä liittyen tekstin muotoiluun

Seuraavassa myös ote Site.css-tiedoston div-osioihin liittyvistä määrittäksistä, joissa asetetaan osioiden koot sekä asettelu:

```

.container
{
  float: left;

```

```

        width: 100%;
        background-color: white;
        height: 100%;
        overflow: visible;
    }
    .menu
    {
        float: left;
        width: 30%;
    }

    .content
    {
        background-color: White;
        margin: 1em 1em 1em 1em;
        width: 60%;
    }
}

```

Koodiesimerkki 2 CSS-määrittäjiä div-osioiden ulkoasuun ja asetteluun

4.7 Tietolähteet

Kun sivuston pohjalla on selkeä ja yhtenäinen ulkoasu sekä toimiva navigointi, on seuraava vaihe itse sisällön esittäminen. Sivuston avulla pitäisi pystyä selaamaan, muokkaamaan, lisäämään sekä poistamaan PerusTA2-kulunvalvontajärjestelmän tietokannan sisältöä. ASP.NET-ympäristön dynaaminen data (Dynamic Data) mahdollistaa toimivan tietojenkäsittelyn toteuttamisen ja joustavan tiedon esittämisen esimerkiksi Details-View-, FormView- ja GridView-komponenteissa, joista enemmän seuraavissa kappaleissa.

Graafisen käyttöliittymän suunnittelussa on kannattavaa käyttää hyväksi monitasoarkkitehtuuria. Tällä tarkoitetaan käytännössä sitä, että ohjelman osiot jaetaan kolmeen osaan: malliin, näkymään ja käsittelijään. Näkymä-osiossa kuvataan itse käyttöliittymän ulkoasu ja esitetään tieto käyttäjälle. Malli-osa kuvaa järjestelmän tiedon tallentamisen, ylläpidon ja käsittelyn. Käsittelijä toimii mallin ja näkymän välillä ja vastaanottaa käyttäjältä tulevat käskyt sekä muuttaa mallia ja näkymää niiden seurauksena. Näin saadaan aikaiseksi tietynlainen riippumattomuus, jossa näkymä voidaan vaihtaa tarvittaessa suhteellisen helposti kokonaan toiseen koskematta hirveästi järjestelmän muihin osiin. [11.]

Tässä työssä kaikki tiedot säilytetään tietokannassa, joka on kehitetty PerusTA2-työpöytäohjelmistoa varten. Tietojen käsittelyä varten on toteutettu erikseen jokaista taulua vastaavat tietoluokat, joissa on määritelty operaatiot tietojen lukemiselle, lisäämiselle, päivittämiselle ja poistamiselle. Nämä tietoluokat voidaan ASP.NET:ssä yhdistää ObjectDataSource-kontrolliin, josta tiedot saadaan helposti käyttöön käyttöliittymäelementtien, kuten pudotusvalikko, kanssa. Koko toimenpide on tehty Visual Studiossa helpoksi ja se vaatii vain vähän ohjelmakoodin kirjoittamista. Tietojen sidonta (Data Binding) on ASP.NET:in ominaisuus, jolla voi yhdistää tietolähteen tietoja esittävään kontrolliin ja joka tekee tietojen esittämisestä joustavaa. Tässä työssä sitä esimerkiksi käytetään, kun halutaan esittää tai muokata tietoa monesta tietokantataulusta kerrallaan yhdestä käyttöliittymäkomponentista.

4.7.1 GridView

GridView on ASP.NET-komponentti, joka mahdollistaa suurten tietomäärien esittämisen taulukossa. Se sisältää monia ominaisuuksia valmiina, kuten tietojen valitsemisen ja sivutuksen, joita voidaan käyttää ilman koodin kirjoittamista. Myös tietojen poistaminen, muokkaaminen, valitseminen ja lisääminen on GridView:llä mahdollista, mutta näitä varten tulee kirjoittaa toimintoja vastaavat metodit.

4.7.2 DetailsView

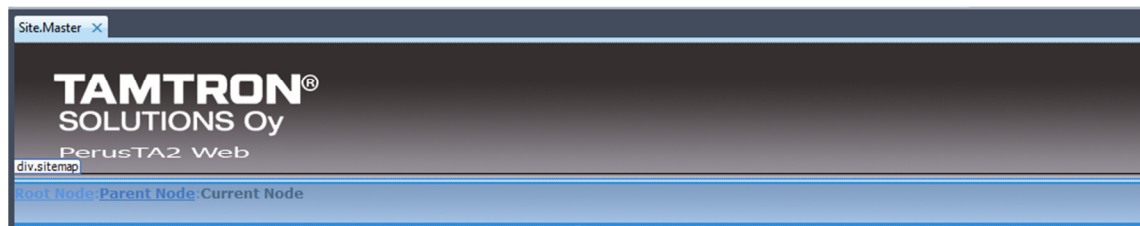
DetailsView on ASP.NET-komponentti, jolla voidaan näyttää yhden tietueen tarkat tiedot kerrallaan. Se koostuu taulukosta, jossa on yksi rivi arvokenttää kohden. DetailsView tukee tietojen muokkausta sekä mahdollistaa tietojen sivutuksen ja helpon selauksen.

4.7.3 FormView

FormView on ASP.NET-komponentti, joka DetailsView:in tapaan on tarkoitettu yhden tietueen tietojen näyttämiseen kerrallaan. Se tarjoaa myös mahdollisuudet sivutukseen ja tietojen selaamiseen. FormView perustuu tyyliin, jotka mahdollistavat kenttien yhdistämisen joustavampaan, ei välttämättä taulukkomaiseen ulkoasuun. FormView koostuu erilaisista malleista, joilla tietoa voidaan käsitellä erilaisilla tavoilla. ItemTemplate on yksi näistä malleista ja sen avulla voidaan näyttää sidottua tietoa vain luku-tilassa. EditItemTemplate-mallin avulla sidottua tietoa voidaan kätevästi muokata ja InsertItemTemplate-mallilla tietokantaan voidaan lisätä uutta tietoa.

5 Rakenne

PerusTA2 web-käyttöliittymän rakenne on kasattu Master Pages -mallin ympärille. Sivuston rakenne on samanlainen riippumatta siitä, millä sivulla käyttäjä on. Yläpalkissa, joka on esitetty kuvassa 8, näytetään jatkuvasti div.header-osio, jossa näytetään yrityksen logo. Sen alla esitetään sivukarttapolku, osiossa div.sitemap, joka päivittyy sen mukaan kun käyttäjä navigoi järjestelmässä. Tämän ansiosta käyttäjä on aina selvillä, missä kohtaa sivustoa hän on. Div-osioista on kerrottu tarkemmin kappaleessa 4.5.3.

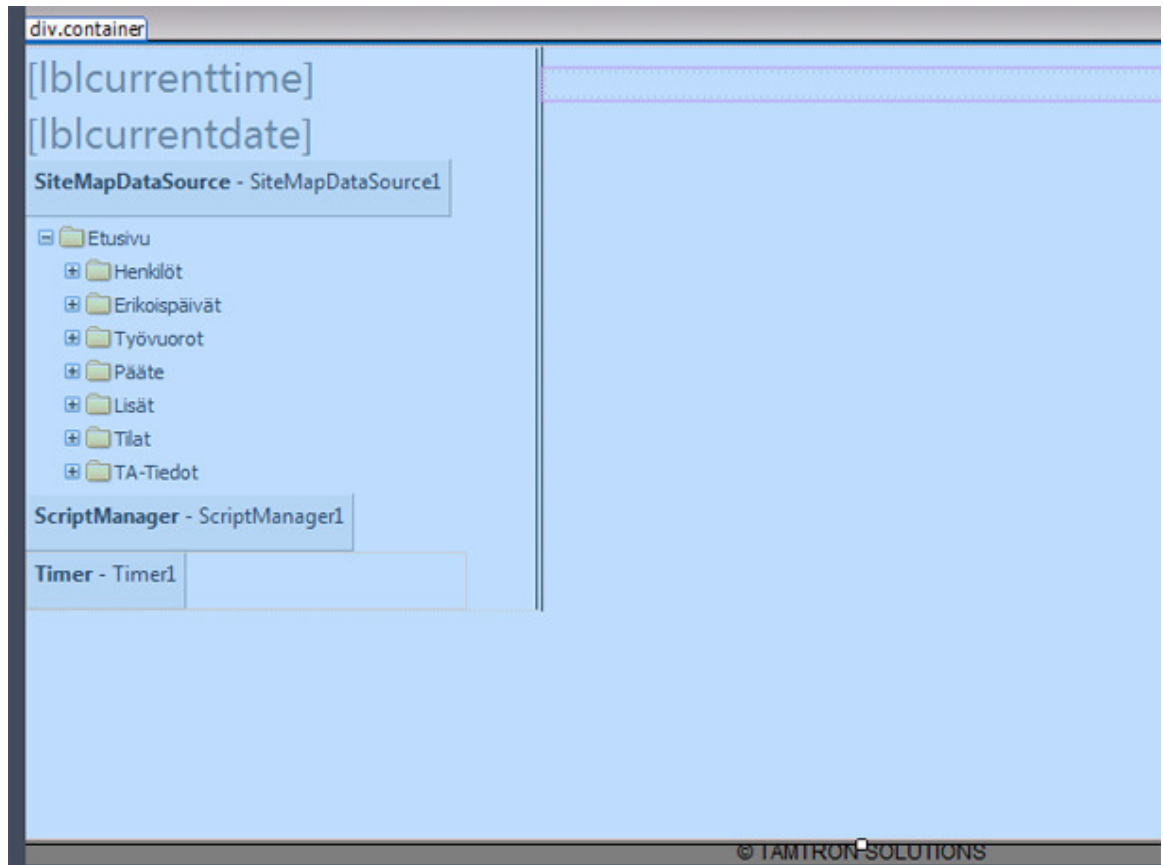


Kuva 8 Sivuston yläpalkki sisältää div.header- ja div.sitemap osiot

Yläpalkin jälkeen sivun rakenteessa tulee div.container, joka toimii "säiliönä" div.menu ja MainContent-osioille. Menu eli valikko näytetään aina sivun vasemmassa laidassa, joten navigointi onnistuu sijainnista huolimatta. Valikon ylälaidassa näytetään nykyinen kellonaika sekä päivämäärä, jotka ovat hyvä olla esillä, koska on kyse työajanseurannasta. Itse valikko on toteutettu TreeView-komponentilla, joka mahdollistaa selkeän mallin valikkorakenteen esittämiseen. Valikon sivut on jaoteltu kansio- ja datasisivuihin, joita kuvaavat vastaavat ikonit. Ideana on, että kun käyttäjä valitsee kansiosivun, aukeaa MainContent-osioon tarkemmat tiedot kyseisen osion mahdollisuuksista. Kansiosivulta ei siis varsinaisesti pysty muuttamaan tai lukemaan järjestelmän tietoja, vaan se on tehty sivuilla navigoinnin helpottamiseksi. Datasivut sen sijaan mahdollistavat järjestelmän tietojen hallinnan ja käyttäjä voi halutessaan siirtyä niihin suoraan valikon kautta. Valikon tietoja voidaan tarvittaessa pienentää tasojen mukaan -/+ -merkistä, joka saattaa helpottaa valikon rakenteen hahmottamista.

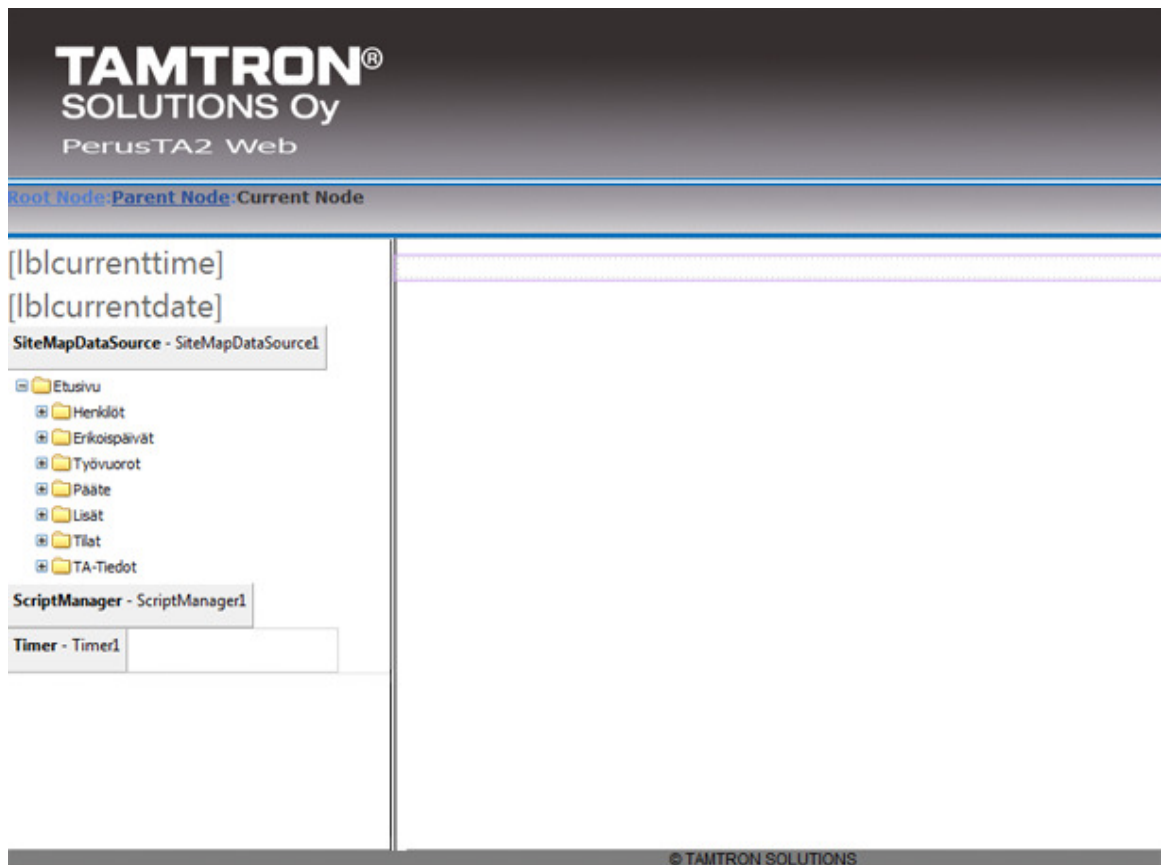
Div.container-osion toinen puoli sisältää ContentPlaceHolder-kontrollin, nimeltä MainContent. Tämä on tärkeä kontrolli sivun kannalta, sillä kaikki sisältö aukeaa sen merkittävään paikkaan. Kuvassa 9 on esitelty Div.containerin sisältö, ja tämä kontrolli näkyy siinä liilan värisenä neliönä. Käytännössä tämä toimii siten, että kun käyttäjä valitsee valikosta sivun, sen sisältö näytetään kyseisessä kohdassa ja muu sivusto säilyy muut-

tumattomana. Toki valikkoon ja sivupolkuun päivitetään nykyinen sijainti, mutta rakenne pysyy samana. Sivuston alalaidassa on vielä yksi div-osio, nimeltään footer, joka on ohut palkki, josta ilmenevät kopio-oikeudet.



Kuva 9 Div.container sisältää valikon ja järjestelmän sisällön

Sivuston rakenne koostuu siis viidestä div-osiosta. Osiot on suunniteltu siten, että ne näkyisivät oikein mahdollisimman monilla erilaisilla resoluutioilla. Tämä toimii periaatteessa siten, että ainoastaan yläpalkki on vakiokokoinen, eli sen koko on asetettu pikseleinä. Tilaa se vie kuitenkin vain vähän. Valikko- ja sisältöosiot on asetettu kooltaan prosenttiosuutena. Näin ollen niiden suhde pysyy samana erilaisissa tarkkuuksissa, ja jos se menee liian pieneksi, tulevat näkyviin vierityspalkit, joten kaikki sisältö on nähtävissä. Kuvassa kymmenen on näytetty vielä järjestelmän yleiskatsaus. Seuraavissa kappaleissa käydään läpi toteutettujen sivujen toteutustapa sekä käyttömahdollisuudet. Tämä työ on vielä kehitysvaiheessa oleva kaupallinen tuote, niin käyttömahdollisuudet joudutaan kuvailemaan yleisellä tasolla.



Kuva 10 Sivuston rakenne

5.1 Henkilöt

Henkilöt-osio mahdollistaa kaikki työntekijöihin liittyvät toimenpiteet. Näitä ovat työvuoroasetukset, kulkukorttien määrittäminen, tapahtumien hallinnointi ja tilatietojen selailu. Osio koostuu viidestä erillisestä sivusta, joista yksi on henkilöt-kansio, joka sisältää lisäinformaatiota muista sivuista.

5.2 Erikoispäivät

Erikoispäivät-osio mahdollistaa erikoispäiviin liittyvät toimenpiteet. Näitä ovat erikoispäivien määrittäminen eli päivien lisääminen, poistaminen ja muokkaaminen. Erikoispäivän määrittely koostuu päivän nimestä, esimerkiksi Joulujoulu ja siihen liittyvästä päivämäärästä, eli tässä tapauksessa 24.12. Erikoispäivistä voidaan myös tehdä ryhmiä, eli

voidaan tehdä kaikki tarvittavat erikoispäivät valmiiksi ja sijoittaa ne ryhmään, jolloin erikoispäivien kohdistaminen esimerkiksi työvuoroihin on helppoa.

5.3 Työvuorot

Työvuoro-osio mahdollistaa kaikkien työvuoroihin liittyvien tekijöiden käsittelyn. Osiossa voidaan luoda uusia työvuoroja järjestelmään sekä poistaa niitä. Työvuoroihin voidaan asettaa siinä sallitut lisät, aloitus aikaikkuna, viikon työajat päiväkohtaisesti, erikoispäiväryhmien työvelvoitteet sekä automaattiset muokkaukset.

5.4 Pääte

Pääte-osiossa voidaan määrittää itse päätteeseen liittyviä asetuksia. Näitä ovat esimerkiksi mahdollisuus näyttää tai piilottaa erilaisia leimaustapoja, kuten sisään- tai ulos syillä –leimaus. Lisäksi voidaan määrittellä päätteen näytöllä näkyvät ja valittavissa olevat tilat. Voidaan myös määrittellä päätteellä näytettävät syykoodit sekä erikseen valittavissa olevat syykoodit.

5.5 Lisät

Lisät-osiossa voidaan hallita lisiin liittyviä asetuksia. Siinä voidaan luoda uusia lisiä tyhjältä tai käyttää olemassa olevaa lisää pohjana. Lisiä voidaan myös poistaa. Lisän määrittely koostuu sen nimestä ja kertymisen ehdoista. Kertymistä voidaan rajoittaa esimerkiksi kellonajan tai päivän työajan mukaan tai se voidaan sallia vain tietyissä päivyissä tai tiloissa. Voidaan esimerkiksi määrittää, että lisiä ei kerry jos henkilö on sairaana tai vaikkapa kahvilla.

5.6 Tilat

Tilat-kohdassa voidaan määrittellä erilaisten tilojen asetuksia. Tiloihin voidaan määrittellä tilan nimi ja tapahtuma, joka aktivoi sen. Tiloihin voidaan myös määrittellä kuuluuko se työaikaan ja kasvattaako se henkilön saldoa. Voidaan myös tehdä automaattitapahtumia esimerkiksi kahvitauko, joka liipaistetaan kun henkilö leimaa itsensä kahvitaulle. Tällöin henkilö on määritellyn ajan kahvitaulla, jonka jälkeen varsinainen työaika jatkuu automaattisesti. Tiloja voidaan tällä sivulla lisätä tyhjältä tai käyttämällä vanhaa tilaa pohjana. Tiloja voidaan myös poistaa käytöstä.

5.7 TA-Tiedot

TA-tiedot –kohdassa voidaan selailla työaikatietoja henkilökohtaisesti. Hakua voidaan rajoittaa halutulle aikajaksolle alkupäivämäärän ja loppupäivämäärän perusteella. Saldoa voidaan tässä kohtaa myös muuttaa manuaalisesti eli se voidaan suoraan asettaa haluttuun arvoon tai sitä voidaan kasvattaa tai vähentää haluttu määrä. Voidaan myös suorittaa saldon uudelleenlaskenta, jolloin ohjelma laskee uuden saldon tehtyjen muutosten perusteella. Lisäksi tässä kohtaa voidaan lisätä lisiä päiväkohtaisesti.

6 Johtopäätökset

6.1 Yhteenveto

Tässä työssä pohdittiin ratkaisuja hyvän ja käytettävän web-käyttöliittymän toteuttamiseen työajanseurantajärjestelmään. Käytiin läpi työajanseurantajärjestelmien vaiheita niiden historiassa sekä tutkittiin erilaisia laitteita ja ratkaisuja. Esiteltiin myös Tamtron Solutions Oy ja siellä nykyisin tämän työn kannalta oleellinen käytössä oleva järjestelmä eli PerusTA2-ohjelmisto oheislaitteineen.

Työn ongelmana oli saada toteutettua helposti käytettävä, selkeä ja graafisesti miellyttävä järjestelmä, joka toiminnoiltaan vastaa PerusTA2-järjestelmää. Työssä käytiin läpi käyttöliittymän suunnitteluun liittyvää teoriaa sekä esiteltiin menetelmät, joilla teorian mukainen järjestelmä saataisiin mahdollisimman hyvin kasattua. Työssä päädyttiin käyttämään ASP.NET-tekniologiaa sen mahdollistamien menetelmien vuoksi ja myös siksi, että yrityksen muissa järjestelmissä on sama tekniologia käytössä. Tämä helpotti järjestelmän toteuttamista sekä myös mahdollistaa tulevaisuudessa erilaisten järjestelmien integroimisen keskenään.

ASP.NET-tekniologia ja Visual Studio 2010-ympäristö mahdollistavat sulavan ja visuaalisen tavan web-järjestelmien tekoon. Navigointi pystyttiin toteuttamaan hyvin käyttämällä site map-kuvausta ja yhdistämällä se tree view-rakenteeseen. Sivustosta saatiin yhtenäisen näköinen käyttämällä master pages –mallia ja sivun laajuista CSS-

tyylimäärittelyä. Järjestelmä toteutettiin monitasoarkkitehtuurin mukaisesti eli ulkoasu ja datakerrokset olivat määriteltä erikseen. Tiedon esittämiseen käytettiin ASP.NET:in komponentteja GridView, DetailsView ja FormView.

Sivuston päärakenteesta tuli lopulta kolmetasoinen rakenne, jossa etusivun jälkeen on seitsemän alitasoa joiden jälkeen toiminnallisuus on pilkottu erillisille sivuille. Näin saavutettiin mielestäni käytön kannalta optimaalinen rakenne, jossa tietoa on sopivasti sivukohtaisesti.

Toiminnallisuuden kannalta tämän työn aikana saatiin toteutettua henkilöihin liittyvät toiminnot lähes kokonaan. Uusia henkilöitä pystytään lisäämään järjestelmään ja olemassa olevia henkilöitä pystytään muokkaamaan tai poistamaan. Henkilöt voidaan myös listata niin, että kaikki perustiedot ja henkilön nykyinen tila näkyvät näppärästi yhdessä kohtaa. Henkilölle voidaan myös asettaa työvuoro, osasto ja yritys. Myös henkilökohtaisien kulkukorttien lisääminen, muokkaaminen ja poistaminen on toteutettu.

Kiireellisen aikataulun johdosta kaikkea PerusTA2-järjestelmän toiminnallisuutta ei ehditty tämän työn puitteissa toteuttaa, vaan puuttuvat osiot on tarkoitus lisätä tulevaisuudessa tämän työn palauttamisen jälkeen. Tässä työssä saatiin kuitenkin toteutettua pohja sivustolle ja tarvittavat komponentit ja logiikat käytiin läpi antaen hyvät mahdollisuudet lopunkin toiminnallisuuden lisäämiseen.

6.2 Jatkokehityssuunnitelmat

Tämän työn tuloksena syntyneen web-käyttöliittymän kehitystä on tarkoitus jatkaa, jotta siitä saataisiin mahdollisimman toimiva järjestelmä Tamtron Solutions Oy:n asiakkaiden käyttöön. Työ jäi sellaiseen vaiheeseen, että ensimmäinen tavoite on luonnollisesti toteuttaa olemassa olevan PerusTA2-ohjelmiston toiminnallisuus loppuun asti myös web-käyttöliittymän kautta käytettäväksi. Tämän työn tuloksena saatiin aikaiseksi hyvä pohja ja valmiudet hyvän käytettävyyden ja selkeän ulkoasun myötä lopunkin toiminnallisuuden toteuttamiseen käyttämällä samaa logiikkaa ja menetelmiä jatkossakin.

Kun kaikki toiminnallisuus on saatu järjestelmään toteutettua ja toiminnallisuus optimoitu mahdollisimman selkeäksi ja tehokkaaksi, voidaan alkaa pohtia muita jatkokehityssuunnitelmia. Käytettävyyden kannalta asiakkaita parhaiten palvelisi luultavasti monimutkaiset hakutoiminnot esimerkiksi koko sivuston laajuisesti. Tämän voisi toteuttaa laittamalla hakulaatikko suoraan sivuston päävalikon yhteyteen, josta käyttäjä voisi helposti hakea haluamaansa toiminnallisuutta ja järjestelmä ohjaisi käyttäjän sivulle, josta toiminnallisuus löytyy. Hakua voisi muutenkin käyttää esimerkiksi henkilöiden valinnan helpottamiseen, sillä tällä hetkellä järjestelmä toimii niin, että henkilöitä voi vain selata listasta. Hakutoiminnoilla siis saataisiin sivuston käytöstä helpompaa ja nopeampaa.

Tamtron Solutions Oy:llä on olemassa CiDWebXS-niminen kulunvalvontajärjestelmä, joka on tämän työn tapaan toteutettu ASP.NET-tekniikalla. Jatkossa ehkä voitaisiin yhdistää PerusTA2 web-käyttöliittymä osaksi CiDWebXS-järjestelmää ja näin saavuttaa kattava järjestelmä, joka kattaisi työajanseurannan lisäksi kulunvalvonnan. CiDWebXS:ssä on myös valmiina toteutettu käyttäjien tunnistaminen käyttäjänimen ja salasanan perusteella, joten nämä järjestelmät yhdistäessä saadaan sisäänkirjautuminen toteutettua helposti myös tämän työn järjestelmään niin, että jotkut ominaisuudet vaatisivat tietynlaiset käyttöoikeudet.

Tällä hetkellä sivuston käyttöliittymä on saatavilla pelkästään suomeksi. Yksi hyvä jatkokehitysidea olisi siis tehdä tuki monikielisyydelle ja lisätä järjestelmään ainakin mahdollisuus vaihtaa kieli englanniksi tai ruotsiksi.

Sivustossa käytössä oleva CSS-tyylitiedosto mahdollistaisi hyvin erilaisten teemojen käytön sivun pohjana. Periaatteessa riittäisi tehdä useampi tyylitiedosto ja sitten antaa käyttäjälle mahdollisuus valita käytössä oleva tiedosto. Tämä tuo käyttäjälle vaihtoehtoja ulkoasun suhteen ja hän voi tarvittaessa valita itseään eniten miellyttävän vaihtoehdon.

Eräs kehitysmahdollisuus olisi myös erillisen mobiiliversion kehittäminen järjestelmästä. Nykyisellään sen käyttäminen esimerkiksi älypuhelimien pieneltä näytöltä ei ole täysin käyttäjäystävällistä. Mobiiliversion voisi olla kompaktimpi kooltaan ja ominaisuuksiltaan ja kattaa jotain vain puhelimen kannalta tärkeiksi havaittuja toimintoja. Yksi tällainen

voisi olla esimerkiksi helppo ja nopea sisäänleimaus. Mobiilisovelluksessa voitaisiin myös hyödyntää widgettejä, joilla kaikkein yleisimmät toiminnot saataisiin helposti esiin.

Tulevaisuudessa kun järjestelmä otetaan käyttöön voidaan käyttöliittymää parantaa vielä palautteen mukaan. Kehitettävää varmasti löytyy ja tavoitteena on saada mahdollisimman paljon näistä ideoista toteutettua.

Lähteet

1. MacDonald, Freeman & Szpuszta, Apress. Pro ASP.NET 4 in c# 2010 USA ISBN-N 978-1-4302-2529-4.
2. Microsoft Corporation. What is ASP.net? [online] saatavissa: <http://wiki.asp.net/page.aspx/1332/aspnet/>.
3. Järvenpää, Juha-Pekka. Ohjeita web-sivuston tekijälle [online] saatavissa: <http://www.sivut.org/sivusto/vinkit/selkeys.php> 3.
4. Microsoft Corporation. Introduction to ASP.NET Dynamic Data [online] saatavissa: <http://www.asp.net/dynamicdata>.
5. Lonergan, Tom. The Evolution of Time & Attendance Systems [online] saatavissa: http://www.hrri.ie/hrri/index.php?option=com_content&view=article&id=103:evolution-timne-attendance-systems&catid=41:technology&Itemid=63.
6. Pihlava, Jorma. Taksvärkkikapula oli torpparin kellokortti [online] Turun Sanomat [Viitattu 04.04.2011] Saatavissa: <http://www.ts.fi/kotimaa/?ts=1,3:1002:0:0,4:2:0:1:2004-06-18,104:2:232241,1:0:0:0:0:>.
7. Tamtron Solutions Oy. T900 esite [online] [viitattu 10.4.2011] saatavissa: <http://www.tamtronsolutions.fi/solutions/sites/tamtron.leka.net.solutions/files/solutions/pdf/T900.pdf>.
8. 2kmediat. CSS-opas [online] [viitattu 25.4] Saatavissa: <http://www.2kmediat.com/css/johdanto.asp>.
9. Mantere, Timo. Automaation tekniikka [online] viitattu [25.4] Saatavissa: <http://lipas.uwasa.fi/~timan/AUTO2020/auttt3.pdf>.
10. Rouhiainen, Eeva-Kaisa. Käyttöliittymän visuaalinen suunnittelu. [online] Saatavissa: <http://www.mit.jyu.fi/opiskelu/seminaarit/bak/kayttoliittyma/>.
11. Wikipedia. MVC-arkkitehtuuri [online] Saatavissa: <http://fi.wikipedia.org/wiki/MVC-arkkitehtuuri>.

12. Hyvönen, Eero. Www-ohjelmointi 1 Merkkäuskielet [online] saatavissa:
<http://www.cs.helsinki.fi/u/eahyvone/courses/muunto01/WWW1Merkkäuskielet.PDF>.
13. SearchSOA.com. Definition. [online] saatavissa:
SGML<http://searchsoa.techtarget.com/definition/SGML>.
14. Wikipedia. HTML. [online] Saatavissa: <http://fi.wikipedia.org/wiki/HTML>.
15. Wikipedia. SGML. [online] Saatavissa: <http://fi.wikipedia.org/wiki/SGML>.
16. Saarikumpu, Osmo. Mitä on validointi? [online] saatavissa:
<http://weppipakki.com/avut/tekstit/validi.htm>.

Esite: T900-tiedonkeruupääte**TAMTRON SOLUTIONS****T900****Tehokas kulunvalvonta-, työajanseuranta- ja tiedonkeruupääte.**

T900-pääte on monipuolisilla liitännöillä varustettu pääte tiedonkeruu-, työajanseuranta- ja kulunvalvontakäyttöön. Pääte on laadukas teollisuuskäyttöön tehty paneeli-PC, joka on erittäin kestävä ja luotettava. Kulunvalvonnassa päätteeseen voidaan liittää useita lukijoita, joiden avulla ohjataan ovien lukkoja sekä kerätään hälytystietoja.

Työajanseurannassa parametroitava käyttöliittymä voidaan muokata asiakkaan tarpeiden mukaiseksi. Työajan ja työvuorojen lisäksi on mahdollista seurata myös työn tekemistä projektikohtaisesti.

Tiedonkeruukäytössä päätteen IO-liittymien ja sarjaporttien avulla voidaan seurata työvaiheita ja koneita. Tieto säilyy myös sähkökatkojen aikana.

Edut ja hyödyt

- Luotettava ja erittäin turvallinen
- Monipuoliset tunnistusmenetelmät: RFID, Biometriset tunnisteet, GSM-puhelin, PIN-koodi
- Skaalautuu asiakkaan tarpeiden mukaisesti
- Joustavat kytkentämahdollisuudet, myös langattomasti (WLAN/GPRS)
- Helppokäyttöiset toiminnot kosketusnäppäimillä
- Tukeva runko, ei liikkuvia osia

Vakio-ominaisuudet

- 8,4-tuuman värillinen kosketusnäyttö
- Kytkentä Ethernet, WLAN tai GPRS yhteyksillä yrityksen verkkoon
- Viivakoodiliitäntä
- Kaksi monikäyttöistä RS232-liitäntää
- Monipuoliset mahdollisuudet parametroida työajanseurannan toimintoja

T900

TEKNISET TIEDOT

T900-pääte on monipuolisilla liitännöillä varustettu pääte tiedonkeruu-, työajanseuranta- ja kulunvalvontakäyttöön. Pääte on laadukas teollisuuskäyttöön tehty paneeli-PC, joka on erittäin kestävä ja luotettava. Kulunvalvonnassa päätteeseen voidaan liittää useita lukijoita, joiden avulla ohjataan ovien lukkoja sekä kerätään hälytystietoja.

Mallit:

- T900 (kulunvalvonta-, työajanseuranta- ja tiedonkeruupääte)
- T900-W (WLAN-yhteydellä)
- T900-G (GPRS-yhteydellä)
- T900-WG (WLAN- ja GPRS-yhteyksillä)

Kotelo:

- IP64-suojaluokitus
- Koko: 234mm * 184mm * 42mm
- Paino: 0,8 kg
- Ei tuulettimia

Sertifioinnit:

- CE, FCC Class B
- RoHS

Lämpötila-alue:

- Toiminta: $\pm 0 \dots +50^{\circ}\text{C}$
- Säilytys: $-20 \dots +60^{\circ}\text{C}$

Liittimet:

- 2 * RJ45 Ethernet
- 2 * DB9 RS232
- 2 * USB
- DC Input

Virtalähde:

- Ulkoinen virtalähde, 45W

Komponentit:

- Max 1 Gt RAM
- Kiintolevynä CF-kortti (1-4 Gt)
- 500 MHz CPU

Näyttö:

- 8,4" TFT LCD
- Resisttiivinen kosketusnäyttö
- 800 * 600 resoluutio
- 262K värisyvyys
- Kirkkaus: 220 cd/m²
- Katselukulma: $-55 \dots +65^{\circ}$

Tiedustelut ja myynti

Tamtron Solutions Oy ja valtuutetut jälleenmyyjät

Esite: RFID-tunnisteet

TAMTRON SOLUTIONS

RFID -TUNNISTEET

Yhdistävät käyttäjän ja lukijan.



Kosketuksettomaan passiiviseen RFID -etälukutekniikkaan perustuvat tunnisteet soveltuvat helppokäyttöisyytensä ja toimintavarmuutensa ansiosta kaikenlaisiin tunnistusta vaativiin sovelluksiin.

Kestävyytensä ansiosta tunnisteita voidaan käyttää vaativissakin olosuhteissa. Tunnisteissa ei ole paristoja tai muita kuluvia osia, joten ne ovat erittäin pitkäikäisiä.

Yhdeksän numeroinen ID -numero mahdollistaa noin neljä miljardia eri numeroyhdistelmää taaten täten erittäin korkean turvallisuustason.

Käytännössä etätunnistus toimii siten, että käyttäjän tunnisteiden ID -numero ohjelmoidaan kulunvalvonta- tai työajanseurantaohjelmaan. Käyttäjä näyttää omaa henkilökohtaista RFID -tunnistettaan kulunvalvonta- tai työajanseurantalukijalle ja lukija tunnistaa käyttäjän ID -numeron avulla ja, esimerkiksi kulunvalvontajärjestelmässä, avaa oven.

Edut ja hyödyt

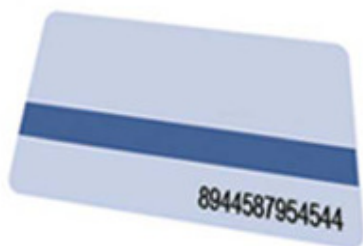
- Kestävät kulutusta
- Pitkäikäisiä ja uudelleen käytettäviä
- Helppokäyttöisiä hyvän lukeutaisuuden ansiosta
- Turvallisia yksilöllisen ID -numeron ansiosta

Vakio-ominaisuudet

- Eivät sisällä paristoja tai muita kuluvia osia
- Jokaisessa tunnistimessa yhdeksän numeroinen ID -numero laserkaiverrettuna
- Tunnisteina avaimenperä, ranneke, ID -kortti tai kotelomallinen tunniste

RFID -TUNNISTEET

TUNNISTEVAIHTOEHDOT



ID 410 Avaimenperätunniste, Pisara

- Kulutusta kestävä muovirakenne, PCB epoksi IP -suojaluokka 67
- Lukuetäisyys noin 5 cm
- Käyttölämpötila -70 °C - +150 °C
- Värinä musta, paksuus 1,5 mm
- Erittäin kestävä

Ranneke

- Lasikuitulujuitettu epoksi, IP -suojaluokka 67
- Lukuetäisyys noin 5 cm
- Käyttölämpötila -40 °C - +85 °C
- Värinä mattamusta

ID 410 Card ID -kortti

- Tavallisen luottokortin kokoinen ja näköinen etätunnistuskortti
- ISO 7816 mukainen termotransfer-tulostukseen soveltuva ID -kortti
- Mahdollisuus varustaa magneettiraidalla (HICO) 4000 Oe
- Lukuetäisyys noin 8 cm
- Käyttölämpötila -30 °C - +50 °C
- Värinä valkoinen, koko: 85,6 mm x 54 mm x 0,76 mm

Kotelomallinen tunniste

- Kestävä muovirakenne, polykarbonaatti
- Pidikeosa mitoitettu ISO -standardin mukaan
- Useita kiinnitysvaihtoehtoja
- Lukuetäisyys noin 5 cm
- Käyttölämpötila -40 °C - +132 °C
- Värinä musta

Kiinnitysvaihtoehdot

- Neulakiinnike
- Ketjukiinnike
- Irroitinkiinnike

Tiedustelut ja myynti

Tamtron Solutions Oy ja valtuutetut jälleenmyyjät