



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Mikael Öst

AJAX – INTRODUKTION OCH IMPLEMENTERING I PRAKTIKEN

Företagsekonomi och turism

2010

ABSTAKT

Författare	Mikael Öst
Lärdomsprovets titel	Ajax – Introduktion och implementering i praktiken
År	2010
Språk	Svenska
Sidantal	42
Handledare	Kimmo Paulaharju

Detta lärdomsprov behandlar utvecklandet av en webbtjänst åt Vasa stad *senior-nätet*. Lärdomsprovet börjar med att beskriva teknikerna som webbtjänsten är baserad på och speciellt AJAX principen. Planeringsstadiet för projektet beskrivs med layout och databasmodell. Webbtjänsten beskrivs sedan med funktionsförklaring och hur AJAX används i detta projekt i praktiken.

ABSTRACT

Author Mikael Öst
Title Ajax – An Introduction and Implementation in Practice
Year 2010
Language Swedish
Pages 42
Name of Supervisor Kimmo Paulaharju

This thesis studies the development of a web based service for the seniornetti web site which is maintained by the town of Vaasa. The thesis begins with describing the techniques which the web service is based upon especially the AJAX principle. The planning stage for the project is described with layout and database models. The web service is later described with an explanation of the functions and how AJAX was used in practice.

Keywords AJAX, PHP, Web Service, Project

INNEHÅLLSFÖRTECKNING

1. INLEDNING	4
1.1 Syfte.....	5
1.2 Avgränsningar	5
2. BETÄCKNINGAR OCH FÖRKORTNINGAR	6
3. OPEN IT LAB OCH VASA STAD SENIORNÄTET.....	7
3.1 Open IT LAB.....	7
3.2 Vasa stad seniornätet	7
4. TEKNIKBEKRIVNING.....	8
4.1 PHP.....	8
4.2 MySQL.....	9
4.3 AJAX.....	11
5. KOSTNADSKALKYLATOR- PLANERINGSSTADIET	14
5.1 Planering och inledande möte	14
5.2 Layout.....	16
5.3 Databas	18
6. KOSTNADSKALKYLATOR – ANVÄNDNING OCH FUNKTIONER.....	20
6.1 Navigering och grundläggande funktioner.....	20
6.1.1 Hemvård - Kommunal hemservice och hemsjukvård.....	22
6.1.2 Hemvård – Servicesedel	24
6.2 Uträkningshistorik	26
6.3 Skriv ut och exportera till PDF.....	28
6.4 Administratörssektion	29
6.4.1 Ändra översättningar	30
6.4.2 Ändra värden	32
7. PROGRAMMERINGSTEKNISKT FÖRKLARING AV UTRÄKNING	34
8. SLUTLEDNING	37
9. KÄLLFÖRTECKNING	39

1. INLEDNING

Detta lärdomsprov kommer att behandla utvecklingen av en webbtjänst för Vasa stads *seniornätet* under sommaren och hösten 2010. Arbetet har utförts som projektarbete för Open IT Lab vid Vasa yrkeshögskola. Arbetet med webbtjänsten har från planeringsstadiet till färdigställning genomförts av mig. Det praktiska arbetet uppskattas till ca 250 h och arbetet utfördes under sommaren 2010. Webbtjänstens syfte är att fungera som en kalkylator som skall underlätta för seniorer eller deras anhöriga att uppskatta kostnader för hemvården.

Den teoretiska delen kommer att ge en närmare beskrivning av begreppet AJAX som webbtjänsten till stora delar baserar sig på. I den teoretiska delen beskrivs även de övriga teknikerna som webbtjänsten baserar sig på som PHP och MySQL i kortare sammandrag.

Planeringen för webbtjänsten behandlas tillsammans med layout och databasplanering. De olika funktionerna som webbtjänsten innehåller kommer att beskrivas skilt för sig i närmare detalj. En inblick i vilken information som uträkningarna baserar sig på samt en generell översikt hur vissa funktioner fungerar programmeringstekniskt kommer även att beskrivas.

1.1 Syfte

Syftet med detta lärdomsprov har varit att utveckla en komplett webbtjänst som kan användas publikt av seniorer eller dess anhöriga för att beräkna eller uppskatta sina kostnader för hemvården. Syftet med den teoretiska delen är att ge läsaren en inblick i teknikerna som ligger till grund för webbtjänsten så att läsaren kan se sambandet och hur de har blivit implementerade i praktiken då webbtjänstens funktionalitet beskrivs senare i lärdomsprovet. Syftet är också att ge läsaren en inblick i hur planeringsskedet fortlöpte för webbtjänsten.

1.2 Avgränsningar

Detta lärdomsprov kommer att ge grundläggande beskrivningar av teknikerna som denna webbtjänst är uppbyggd av så att läsaren skall kunna se implementeringen i den praktiska delen. Några detaljerade beskrivningar av teknikerna kommer inte att ges. Användning och funktioner på webbtjänsten kommer att beskrivas så att både en läsare som är och inte är insatt i programmering kan uppskatta beskrivningen. Den kompletta källkoden kommer inte att ges i detta lärdomsprov men exempel på källkoden finns i den programmeringstekniska beskrivningen i kapitel 7.

2. BETECKNINGAR OCH FÖRKORTNINGAR

Java Script	Klientbaserat objektorienterat programmeringsspråk som kan användas för att utföra diverse interaktiva funktioner på en webbsida.
CSS	Språk som kan användas för att skapa en layout och bestämma t.ex. teckenstorlek.
XML	Ett märkspråk som används för att transportera och lagra text i ett universellt format.
Session	Temporär information som sparas på webbservern för den aktuella klienten.
Cookie	Webbsida sparar en temporär fil på användarens dator för att underlätta vid återkommande besök.
Variabel	Minnesplats som används vid programmering.
Vektor	Likt en variabel men innehåller flera minnesplatser. Mängden minnesplatser kan specificeras vid deklarerings.
Klass	Lagar metoder och funktioner vid objektorienterad programmering.
XMLHttpRequest	Ett programmeringsgränssnitt för webbutveckling som används för att skicka HTTP eller HTTPS trafik från klientdatorn till webbservern och få tillbaka informationen som t.ex. XML format. [12]
XHTML	En utvecklad variant av standard HTML som har striktare regler för hur den tolkas eftersom det är tänkt att användas tillsammans med XML. [13]

3. OPEN IT LAB OCH VASA STAD SENIORNÄTET

3.1 Open IT LAB

Open IT Lab är en del av Vasa yrkeshögskola och drivs av Kimmo Paulaharju. Inom Open IT Lab får studerande arbeta med projektarbeten som är verkliga uppdrag. Uppdragen kan komma från företag och organisationer inom nejden men också utanför samt projekt inom skolan. Projektens uppdrag kan vara att utveckla eller skräddarsy webbsidor, CRM eller CMS system men också andra ICT behov som en uppdragsgivare kan ha. Projekten strävar till att som namnet säger att använda sig av öppna källkodslösningar.[10]

3.2 Vasa stads seniornätet

Vasa stads *seniornätet* är en portal som riktar sig till seniorer, anhöriga och yrkesfolk. Webbtjänsten erbjuder sådan serviceinformation som är relevant för den äldre befolkningen som boende, socialskydd och hälso- och sjukvårds information. Besökaren får kännedom om både kommunala, privata och andra organisationers tjänster samt länkar till dessa. Förutom serviceinformation finns även aktuella händelser om evenemang och aktiviteter som riktar sig till seniorer samt information och länkar till föreningar med liknade verksamhet. [9]

4. TEKNIKBESKRIVNING

4.1 PHP

PHP: Hypertext processor är idag är ett av de populäraste programmeringsspråken som används inom webbutveckling. PHP är fritt att använda och modifiera eftersom det är lanserat som öppen källkod, den senaste versionen av PHP är version 5. PHP är ett såkallat server-side språk vilket innebär att källkoden körs på webbservern tillskillnad från t.ex. programmeringsspråket JavaScript som körs på klientens dator. PHP kod skall skrivas enligt denna syntax för att tolkas korrekt: [1] [3. s.1-5]

```
<?php
for ($i=1; $i<=3; $i++)
{
    echo "Talet är: " . $i . "<br/>";
}
?>
```

Talet är: 1
Talet är: 2
Talet är: 3

Bild1. PHP kodexemplet, exemplet räknar till tre.

PHP kod kan bäddas in i vanlig HTML kod och kan på så sätt enkelt kombineras med HTML. Ett möjligt användningsscenario kan då se ut så att man använder ett PHP script för att hämta information ur en databas och sortera den hämtade informationen i en önskad ordning. Informationen presenteras sedan visuellt med en HTML tabell eller alternativt CSS. [3. s.1-5]

4.2 MySQL

MySQL är en databashanterare som baserar sig på frågespråket SQL. MySQL skapades som ett internt företagsprojekt av finlandssvenske Michael Widenius och David Axmark och släpptes publikt 1995. Företaget MySQL AB grundades senare i Uppsala Sverige för att vidareutveckla produkten. MySQL såldes 2008 till Sun Microsystems som idag äger och underhåller produkten. MySQL existerar idag under två olika licensmodeller GNU General Public License som är fri att använda och vidareutveckla men också ett licenserings alternativ där support ingår. [3. s. 511-512]

En databas som är skapad med SQL språket lagrar information i tabeller. I varje tabell finns ett antal kolumner som kan konfigureras att lagra en speciell informations typ t.ex. siffror med två decimaler. Informationen kan sedan hämtas ur databasen med en SQL fråga.

SELECT * from Personer WHERE Förnamn = 'Mikael'

Denna fråga hämtar alla personer från tabellen personer med förnamnet Mikael och listar alla kolumner som finns i tabellen.

PersonID	Förnamn	Efternamn	Födelsedat
	Mikael	Öst	300987
*	(New)		

Bild 2. Resultatet av SQL frågan.

SQL tabeller kan också använda sig av relationer vilket innebär att två eller flera tabeller är beroende av varandra. För att en relation skall kunna upprättas så måste en gemensam identifierare finnas i båda tabellerna t.ex. ett id nummer. Då kan tabellerna Personer och Adresser kopplas ihop så personuppgifter finns i tabellen Personer och adressuppgifterna i tabellen Adresser. För att en adress skall kunna registreras i tabellen Adress så måste först en personen som adressen skall kopplas till registreras i tabellen Personer

Relationen som visas i figuren nedan är en relation som har egenskapen en till många vilket innebär att det får endast finnas en person i tabellen Personer som har samma ID nummer men denna person kan ha flera olika adresser i tabellen Adresser med samma ID nummer. Relationen kan konfigureras så att om Mikael raderas från tabellen Personer så raderas också Mikael's alla adresser ur tabellen Adresser.

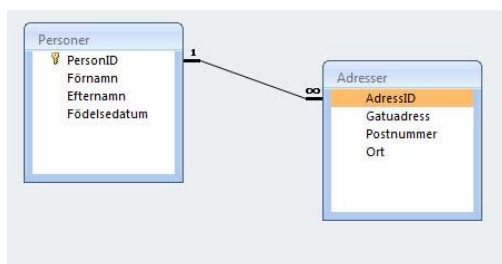


Bild 3. En till många relation mellan tabellen Personer och Adresser.

4.3 AJAX

AJAX (*Asynchronous JavaScript and XML*) är ett relativt nytt begrepp inom webbutveckling. Förkortningen AJAX uppstod 2005 av Jesse James Garrett hos Adaptive Path i USA i en artikel då han beskrev detta begrepp. AJAX är inte en ny teknik utan kan snarare beskrivas som ett nytt sätt att se på webbutveckling. AJAX har heller ingen upphovsman eller någon som kan göra anspråk på det.

AJAX är en samling av existerande tekniker som sammanbetar med varandra. Några av dessa tekniker är XMLHttpRequest, JavaScript, XML, XHTML samt CSS. AJAX kan beskrivas som en metod att dynamiskt hämta innehåll från webbservern och databasen utan att stoppa upp interaktionen på en webbsida för användaren då webbsidan laddas om med nytt innehåll. [4]

Den traditionella principen då man har utvecklat webbsidor har länge varit att då användaren klickar på en länk så skickas en förfrågan till webbservern som behandlar den och som i sin tur hämtar det önskade materialet från databasen. Innehållet från databasförfrågan sätts in i en designmall och innehållet skickas tillbaka till användaren och webbläsaren laddar om webbsidan med det nya innehållet. Användarens interaktion med webbsidan bryts dock och användaren blir tvungen att vänta på att det nya innehållet skall hämtas. Detta är ett synkront sätt att hantera webbförfrågningar.

Denna grundprincip har använts ända sedan Tim Berners-Lee gjorde det första utkastet till World Wide Web 1990. Principen fungerar också idag utmärkt för webbsidor vars innehåll är väldigt statiska och inte strävar att vara särskilt interaktiva för användaren. [14]

AJAX är ett asynkront sätt att göra webbförfrågningar. Istället för att stoppa interaktionen för användaren då ny data hämtas så hämtas data dynamiskt utan användarens vetskap. Detta uppnås med en så kallad AJAX motor som egentligen är ett Java Script objekt eller funktion. Då användaren klickar på en länk görs en förfrågan till AJAX motorn som i sin tur gör förfrågan till webbservern. Webbservern hämtar önskat innehåll från databasen och returnerar innehållet i XML format som AJAX motorn kan tolka. Informationen kan sedan presenteras för användaren utan att webbsidan behöver laddas om. Detta innebär också att minimalt med data behövs överföras mellan webbserver och webbläsaren eftersom bara nödvändig data behövs skickas av webbservern till webbläsaren. AJAX motorn agerar således som ett slags extra lager som används som kontakt till webbservern. [2. s. 1-7]

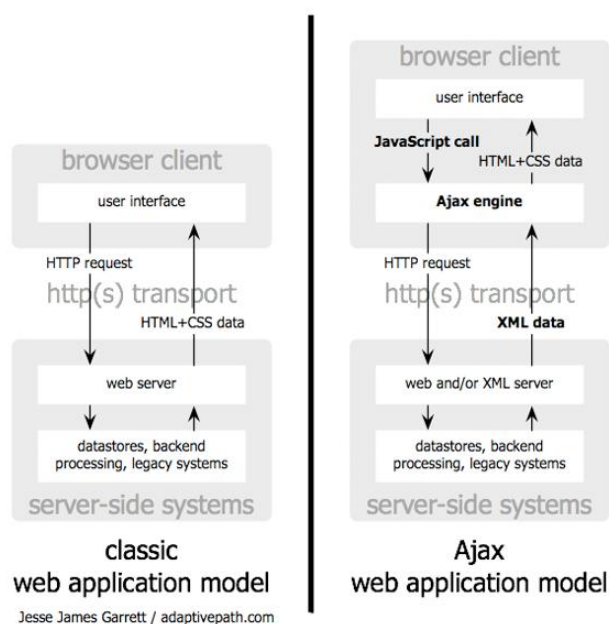


BILD 4. Modellen visar skillnad mellan traditionella webbapplikationer och AJAX baserade. [4]

Detta flyttar också gränserna för vad en webbsida kan ha för syfte. Genom att dynamiskt kunna hämta data och inte behöva ladda om sidan kan man uppnå sådan interaktion som tidigare enbart fanns i applikationer som är installerade lokalt på datorn. Ett exempel är Google docs som är en ordbehandlare i webbläsaren och har samma grundfunktionalitet som man förväntar sig att en ordbehandlare skall innehålla.

AJAX är dock inte den första lösningen för att hämta data dynamiskt och minska omladdningar av webbsidor. Ett exempel på detta är en teknik som kallas IFrames. Denna teknik introducerades redan med Microsoft Internet Explorer 4 och kunde kombineras med Java Script för att uppnå dynamiskt hämtning från webbservern. [2][4][11]

5. KOSTNADSKALKYLATOR- PLANERINGSSTADIET

5.1 Planering och inledande möte

Projektet inleddes med att Vasa Stad önskade en webbtjänst till den nya portalen *seniornätet*. Webbtjänsten skulle vara en kostnadskalkylator som riktar sig till seniorer eller deras anhöriga. Kalkylatorn syfte skulle vara att räkna ut kostnader för hemvård enligt ett antal parametrar.

Projektet inleddes för min del 11 juni 2010 med ett möte vid äldrevårdscentret i Vasa med kontaktpersonerna Carola Hannula, Maija-Liisa Tiilikka, Anna Björninen. Under mötet fick jag ta del av papper med vad uträkningarna baserar sig på och en del information om vad de önskade av tjänsten. Utöver detta så fick jag rätt stor frihet att utforma tjänsten som jag tyckte att det skulle passa. Utgången av mötet var att jag under sommaren då de flesta av kontaktpersonerna var på semester så skulle jag konstruera en prototyp med ett layoutförslag och de flesta av funktionerna implementerade.

Som teknisk grund i webbtjänsten skulle jag använda mig av PHP och som databas hanterare MySQL. För att bygga upp designmallen använde jag mig av CSS. Java Script planerade jag också använda speciellt vid inmatningsvalidering. Jag hade använt denna kombination vid flertal projekt tidigare så jag var bekant med den och visste att jag kunde lösa uppgiften med denna kombination.

Webbtjänstens innehåll skulle förstås finnas på både inhemska språken. Vid tidigare projekt inom Open IT Lab hade vi använt oss av en lyckad lösning då ett värde för valt språk lagras i en session. Jag bestämde mig för att implementera systemet också i detta projekt eftersom jag visste att det fungerade bra och var lätt att implementera.

Jag valde också att använda mig av dynamisk hämtning och skrivning till databasen för att öka interaktionen för användaren och minska omladdningar av hemsidan. Till detta skulle jag använda mig av AJAX principen. Jag hade experimenterat med denna princip tidigare men inte implementerat den fullt ut i ett projekt tidigare. De funktioner där AJAX skulle implementeras skulle vara rätt grundläggande mot vad som är möjligt att åstadkomma med AJAX men skulle ge en viss erfarenhet inom principen samt att det skulle göra att webbtjänsten skulle kännas mera interaktiv.

En idé som jag fick rätt så tidigt i projektet och som gjordes lättare att genomföra med AJAX var en uträkningshistorik. Den skulle automatiskt uppdateras med nya uträkningar allt eftersom de utförs. Eftersom ingen registrering skulle krävas för denna hemsida så blir möjligheterna med att spara historik begränsade så jag bestämde mig att historiken skulle spara uträkningar som gjordes så länge som webbläsaren var öppen genom sessioner.

Uträkningarna baseras på procentsatser och olika värden som kan komma att ändras från år till år. Det skulle därför vara en nödvändighet att dessa tal skulle kunna administreras i efterhand så jag planerade för en administratörssektion från början i projektet. Där skulle det också finnas möjlighet att göra ändringar i de finska och svenska översättningssträngarna.

5.2 Layout

Genom de uträkningsexempel jag hade till pappers kunde jag se att uträkningarna delades upp i två olika typer kommunal hemservice och hemsjukvård samt uträkning av servicesedel. Jag bestämde mig för att använda mig av ett tabbaserat system där jag lade ett uträkningsformulär under varsin tabb för att dela upp tjänsten så att den blir så enkel att använda som möjligt då målgruppen för tjänsten kanske inte har så stor erfarenhet av datorer.

I planeringskedet gjorde jag ett ganska enkelt utkast till layout för hemsidan. Överst på sidan finns logotypen med kombinerad tabbmeny. Två olika tabbar finns och det skall vara möjligt att se vilken tabb som är vald. I den övre delen av webbsidan finns också språkväljaren där användaren kan byta språk mellan finska och svenska om så önskas.

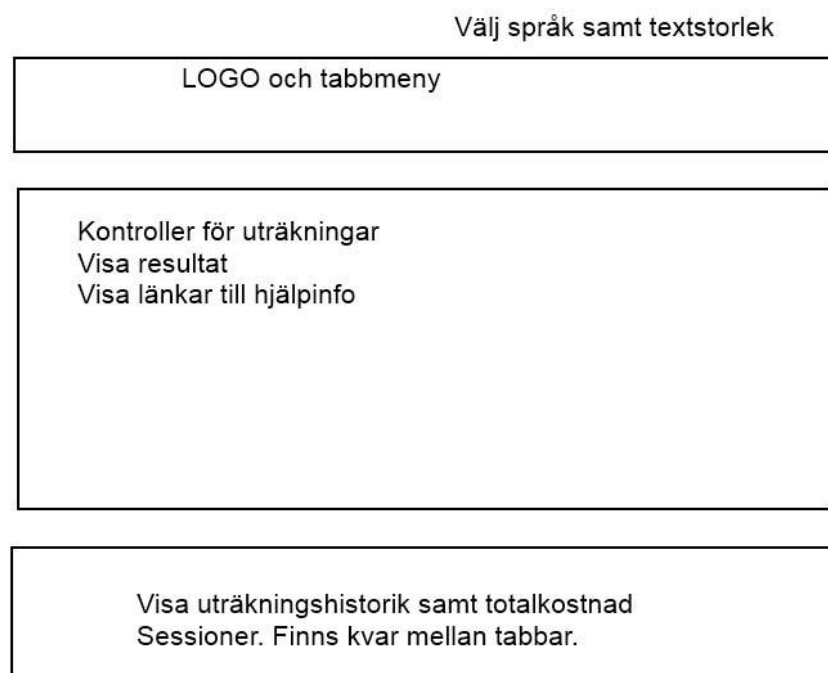


BILD 5. Layout vid planeringsstadiet i mitten av juni 2010.

I den mittersta sektionen visas uträkningsformuläret baserat på vilken tabb som användaren har valt. Upplägget på de två olika formulären är rätt lika och enbart inmatningsalternativ skiljer dessa åt. Bredvid varje inmatningsfält så finns också en länk som erbjuder möjlighet för användaren att få mera information om vad respektive inmatningsfält betyder. Under uträkningsformuläret presenteras resultatet då användaren trycker på räkna knappen. Resultaten presenteras dynamiskt utan att webbsidan behövs laddas om eftersom jag använder mig av AJAX principen.

I den nedersta sektionen av webbsidan finns uträkningshistoriken. Den är uppdelad i varsin del för kommunal hemservice och hemsjukvård samt servicese- del. I respektive del visas tidigare uträkningar som användaren har gjort. Under historiken visas dessutom en sidfot med information om upprätthållaren av webbtjänsten.

5.3 Databas

Databasen för detta projekt körs på MySQL och består av fem tabeller. Tabellerna har inga relationer mellan varandra då jag inte ansåg att detta var nödvändigt för funktionaliteten för webbtjänsten samt att ingen registreringsfunktion av användare existerar. Tabellernas uppgift är att erhålla fasta värden så att uträkningarna kan utföras. Förutom detta så har en tabell som uppgift att lagra alla textsträngar på finska och svenska.

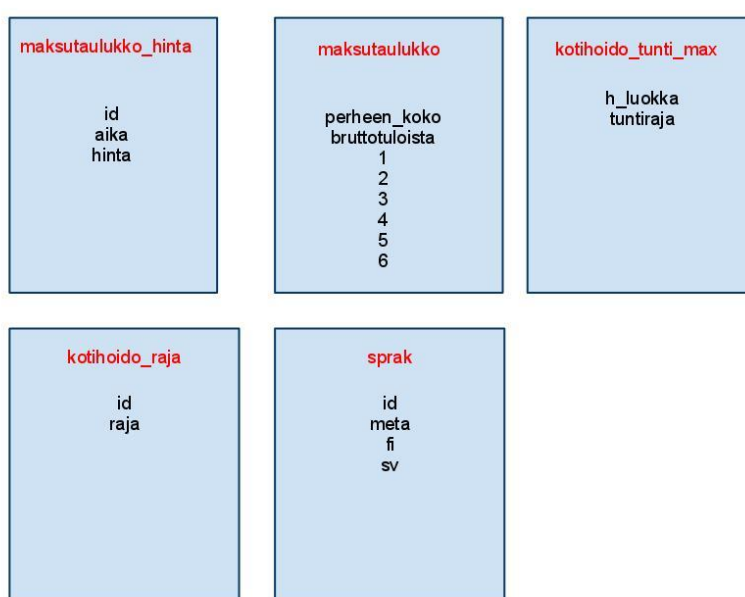


Bild 6. Databasmodell för projektet

Tabellen maksutaulukko_hinta används till uträkningar för servicesedel. Tabellen innehåller två värden som används då olika kostnader gäller för vardag samt för kväll/helg. Kolumnen ID används för att skapa en unik identifikationsnummer åt de lagrade värdena. I Kolumnen aika finns en textsträng den används för att veta vart värdet hör som lagras i följande kolumn hinta.

I tabellen maksutaulukko används till både kommunal hemservice och servicesedel. Tabellen innehåller värden som används av alla uträkningar. I kolumnen perhreen_koko listar olika stolekar för boendestorlek(1-6 personers boende). I nästa kolumn bruttotuloista lagras ett inkomstvärde som ökar ju större boendet är. De följande kolumnerna är numrerade från ett till sex. Dessa används som sex olika vårdklasser för varje boendestorlek. Dessa vårdklasser är baserade på hur många timmar man behöver hjälp i månaden. Vårdklasserna innehåller procenttal som sedan används vid uträkningarna.

Tabellen kotihoido_tunti_max används till uträkningarna för kommunal hemservice och hemsjukvård. Tabellen uppgift är att tillsammans med tabellen kotihoido_raja räkna ut timpriset och jämföra det med den lagstadgade max- och minimitaxan för timpriset. I kolumnen h_luokka finns de sex olika vårdklasser som också finns i tabellen maksutaulukko. I nästa kolumn tuntiraja finns en timgräns där maxantalet timmar för den aktuella vårdklassen definieras.

Tabellen kotihoido_raja används till uträkningar för både kommunal hemservice och servicesedel. Tabellen innehåller den lagstadgade gränsen för vad som är max- och minitaxan för respektive uträkning. Kolumnen ID används som identifierare för de två olika gränserna och kolumnen raja innehåller värdena.

I tabellen sprak innehåller alla textsträngar på både finska och svenska. För att identifiera varje sträng finns en meta kolumn med ett nyckelord på engelska för att den skall vara universell för båda språken. Själva textsträngen finns i kolumnerna fi och sv.

6. KOSTNADSKALKYLATOR – ANVÄNDNING OCH FUNKTIONER

6.1 Navigering och grundläggande funktioner

Användaren navigerar mellan de två uträkningsformulären genom tabbar. Formuläret under den första tabben innehåller uträkningar för kommunal hemservice och hemsjukvård. Uträkningsformuläret för servicesedel finns under den andra tabben.

The screenshot shows the Vasa City website interface for a calculator. At the top left is the VASA logo. In the top right corner, there are links for 'Help', 'Finska', and 'Svenska'. Below this, there are two tabs: 'Hemvård - Kommunal hemservice och hemsjukvård' (selected) and 'Hemvård - Servicesedel'. The main content area is titled 'Hemvård - Kommunal hemservice och hemsjukvård' and contains three sections: 'Inkomster' with a text input for 'Ange din bruttolöns per månad' and a 'Utörligare information' link; 'Familiens storlek' with a dropdown menu set to '2 pers. (959 €)' and a 'Utörligare information' link; and 'Timmar' with a dropdown menu set to '< 8 h (12,57 %)' and a 'Utörligare information' link. A 'Räkna' button is located at the bottom of these sections. Below the calculator, there are two sections for 'Uträkningshistorik för kommunal hemservice och hemsjukvård' and 'Uträkningshistorik för servicesedlar', both with explanatory text and links. At the bottom of the page, it says 'Vasa Stad 2010'.

BILD 7. Det färdiga utseendet för kalkylatorn under tabben kommunal hemservice och hemsjukvård.

Uppe i högra hörnet så finns möjlighet att ändra språk mellan finska och svenska. Det finns dessutom en länk som leder till en hjälpfil i PDF format. Beroende på vilket språk som är valt visas hjälpfilen antingen på finska eller svenska. I den nedre sektionen visas uträkningshistoriken och längst nere visas en sidfot.

Layouten byggs upp av tre PHP filer. Index.php innehåller den grafiska botten för webbtjänsten med hjälp av en CSS fil style.css. Då ett uträkningsformulär är valt så inkluderas PHP filen som innehåller formuläret t.ex. sid1.php om kommunal hemservice och hemsjukvård är valt. Filen history.php inkluderas också till index.php för att alltid aktuell historia skall följa med användaren oberoende vilket uträkningsformulär användaren befinner sig på.

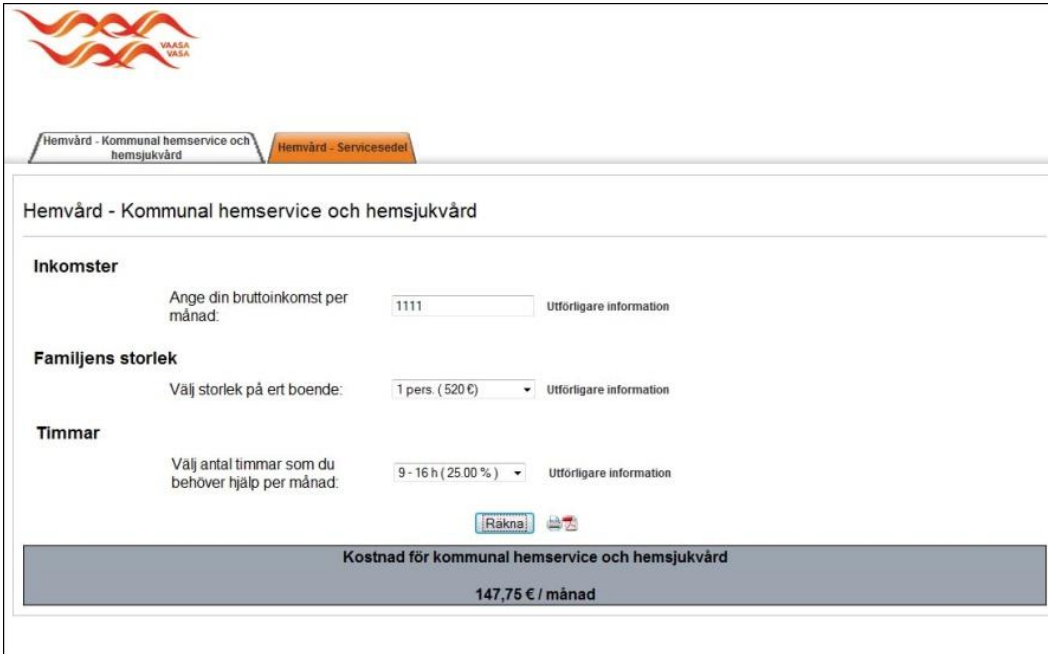
Webbtjänsten laddas enbart om då användaren växlar mellan tabbar och byter språk. Information som skickas från formuläret på webbservern till klientdatorn samt information som hämtas från databasen hämtas dynamiskt med AJAX principen. Detta inkluderar presentation av resultat och visning av uträkningshistorik.

Båda uträkningsformulärens använder sig dessutom av varsitt Java Script för att validera data som användaren matar in. Fälten där enbart numeriska tal skall tillåtas att matas in kontrolleras så att fältet inte är tomt och att det enbart innehåller numeriska värden. Kontroller på rullgardinslisterna och kryssboxar görs också för att kontrollera att ett värde har valts.

6.1.1 Hemvård - Kommunal hemservice och hemsjukvård

Det första uträkningsformuläret används för att räkna ut kostnader för kommunal hemservice och hemsjukvård. För att utföra en uträkning måste användaren mata in uppgifter i tre olika inmatningsfält inkomst, familjens storlek vilket också kan ses som hur många som bor i hushållet och antal timmar som hjälpen behövs per månad.

Inkomster skrivs in som en numerisk kombination och decimaler är tillåtna. Användaren måste välja familjens storlek före antal timmar kan väljas. Då användaren har valt familjens storlek så hämtas procentsatserna som är unika för den valda familjens storlek. Dessa hämtas dynamiskt och visas i en ny rullgardinslista. Användaren kan då välja hur många timmar hjälp som behövs per månad.



The screenshot shows a web-based calculation form for home care services. At the top left is the logo for VASA VASA. Below it are two tabs: 'Hemvård - Kommunal hemservice och hemsjukvård' (selected) and 'Hemvård - Servicesedel'. The main heading is 'Hemvård - Kommunal hemservice och hemsjukvård'. The form is divided into three sections: 'Inkomster' with a text input for 'Ange din bruttointkomst per månad:' containing '1111' and a link for 'Utförligare information'; 'Familjens storlek' with a dropdown menu for 'Välj storlek på ert boende:' set to '1 pers. (520 €)' and a link for 'Utförligare information'; and 'Timmar' with a dropdown menu for 'Välj antal timmar som du behöver hjälp per månad:' set to '9 - 16 h (25.00 %)' and a link for 'Utförligare information'. Below these sections is a 'Räkna!' button with a calculator icon. At the bottom, a grey box displays the result: 'Kostnad för kommunal hemservice och hemsjukvård' and '147,75 € / månad'.

BILD 8. Uträkningsformlär för kommunal hemservice och hemsjukvård

Formeln för uträkningen ser ut på detta sett: (inkomst – kostnad för boendestorlek) * procenttal för antal timmar. Detta resultat ger månadskostnaden. Resultatet divideras sedan med max antal timmar för den timklass som användaren tillhör. Detta görs för att kontrollera om användaren överstiger maxtaxan för timkostnaden. Om användaren överstiger maxtaxan ändras resultatet till maxtaxans värde som är 46,57 € per timme då detta skrivs. Resultatet omvandlas till månadskostnad och presenteras för användaren under räkna knappen i en avskild ruta, om maxtaxan används så ges en beskrivning för användaren vad som har hänt och varför. Samtidigt så uppdateras uträkningshistoriken med den utförda uträkningen. Uträkningshistoriken beskrivs närmare i kapitel 6.2.

6.1.2 Hemvård – Servicesedel

Det andra uträkningsformuläret används för att räkna ut värdet på en servicesedel som en senior kan använda sig av hos olika vårdföretag. Användaren matar in sina inkomster per månad och väljer därefter familjens/boendets storlek. Tidpunkten på dygnet när servicesedeln skall vara giltig måste också väljas. Här finns det två alternativ vardag eller kväll/helg, användaren måste välja minst ett av dessa men kan kryssa i båda alternativen.

The screenshot shows a web-based calculation form for home care services. At the top, there is a logo for 'VÅASA VASA' and two tabs: 'Hemvård - Kommunal hemservice och hemsjukvård' and 'Hemvård - Servicesedel'. The main heading is 'Hemvård - Servicesedel'. Under the heading, there are three sections: 'Inkomster' with a field for 'Ange din bruttoinkomst per månad' containing '1300'; 'Information för beskrivningar av servicesedelns värde' with a field for 'Välj storlek på ert boende' containing '1 pers. (520 €, 35.00 ?'; and 'Tid på dygnet' with two checked checkboxes for 'Vardag' and 'Kväll/helg'. A 'Räkna' button is located below the checkboxes. At the bottom, a grey box displays the results: 'Servicesedelns värde är: Vardag - 19,45 € / timme, Kväll/Veckoslut - 27,45 € / timme'.

Bild 9. Uträkningsformulär för servicesedel.

Formeln för uträkning av en servicesedel ser ut på detta sätt: (inkomst – kostnad för boendestorlek) * procenttal för boendestorlek. Den mellansumma som fås från denna uträkning divideras därefter med talet 60. Resultatet från detta subtraheras sedan från talet som finns bakom tidpunkten på dygnet som servicesedeln skall gälla t.ex. 24 för vardag för att få slutresultatet.

Slutresultatet kontrolleras sedan så att värdet ligger över gränsen för minimivärdet för en för vad en servicesedel minst får vara vilket är 7 € då detta skrivs. Om slutresultatet ligger under gränsen så justeras resultatet till minigränsen. Om användaren har kryssat i att den vill ha en uträkning för både vardag och kväll/helg så körs uträkningen två gånger för att få fram värdet för två servicesedlar som gäller för både vardag och kväll/helg. Resultatet presenteras för användaren och om minivärdet har använts som resultat i någon av uträkningarna så ges en beskrivning för användaren vad som har hänt och varför. Samtidigt uppdateras uträkningshistoriken med de utförda uträkningarna, vilket beskrivs närmare i nästa kapitel 6.2.

6.2 Uträkningshistorik

Förutom att användaren får ta del av resultatet av sina uträkningar lagras alla uträkningar som användaren har utfört under tiden webbläsaren har varit öppen. Uträkningshistoriken lagras i varsin session för båda uträkningsformulären och har också varsin sektion i den nedre tredje delen på hemsidan. Informationen i sessionerna används också då användaren skall skriva ut uträkningar eller exportera som PDF format. Detta beskrivs i nästa kapitel 6.3.



Bild 10. Denna text visas om ingen uträkningshistorik hittas.

Båda uträkningsformulären lagrar uträkningarna i sina sessioner på samma vis, det som skiljer dem åt är mängden information som lagras. För uträkningsformuläret servicesedel lagras information på detta vis i sessionen service. Alla uträkningar sparas som en textsträng med ”|” som ett separeringstecken som skiljer uträkningarna åt. Mellan separeringstecknet så finns detaljerna som uträkningarna baseras på. Detta är informationen som användaren har matat in samt procenttalen som ligger till grund för uträkningarna. Här lagras också resultatet för uträkningen. Detaljerna separeras också med ett separeringstecken från varandra vilket är ”#”.

En uträkning i sessionen service kan då se ut på detta sätt: |#Inkomst: 1250,54 €#Boendestorlek kostnad: 520 €#Procenttal för boendestorlek: 35,00#Kostnad för vardag eller kväll/helg: 24,00 €#Resultat: 19,74 €|. Om det finns flera uträkningar lagrade så följer dessa uträkningar efter varandra.

Då historiken skall visas för användaren så måste först uträkningarna i sessionen separeras från varandra. Detta sker genom att textsträngen körs genom en funktion som returnerar en vektor med varje uträkning i en skild minnesplats. Varje uträkning separeras sedan ytterligare en gång så att varje detalj t.ex. resultatet får en egen post i en ny vektor.

Uträkningshistorik för kommunal hemservice och hemsjukvård	
Uträkning 1 - Total månadskostnad:	118,2 €
Uträkning 2 - Total månadskostnad:	147,75 €
Uträkningshistorik för servicesedlar	
Uträkning 1 - Servicesedelkostnad/timme:	19,45 €
Uträkning 2 - Servicesedelkostnad/timme:	27,45 €
Uträkning 3 - Servicesedelkostnad/timme:	20,55 €
Uträkning 4 - Servicesedelkostnad/timme:	28,55 €

Vasa Stad 2010

Bild 11. Historik då uträkningshistorik hittas.

När historiken presenteras för användaren listas inte alla detaljer som ingår i uträkningen direkt. Resultatet för uträkningarna listas samt vilken uträkning i ordningen det handlar om. Om användaren vill se en mera detaljerad beskrivning av vad en uträkning baseras på kan användaren hålla musen ovanför resultatet för önskad uträkning. En ruta visas då som innehåller ytterligare detaljer.

Uträkningshistorik för kommunal hemservice och hemsjukvård	
Hit kommer dina uträkningsresultat att visas skilt för sig så att det blir enkelt att få en översikt.	Bruttoinkomst: 1300 €
Uträkningshistorik för servicesedlar	Boendekostnad: 520 € (35,00 %)
Uträkning 1 - Servicesedelkostnad/timme: 19,45 €	Vardag kostnad: 24,00 €
Uträkning 2 - Servicesedelkostnad/timme: 27,45 €	Servicesedelkostnad/timme: 19,45 €

Vasa Stad 2010

Bild 12. Ytterligare detaljer visas om muspekaren hålls ovanför resultatet i uträkningshistoriken.

6.3 Skriv ut och exportera till PDF

Användaren kan skriva ut uträkningsresultaten eller exportera dessa till PDF format. Den information som skrivs ut eller exporteras hämtas från sessionen för det aktuella uträkningsformulär som användaren befinner sig på. Textsträngen från sessionen separeras med samma funktioner som beskrevs tidigare i kapitel 6.2.

Tid på dygnet

vilken tid på dygnet som hjälpen behövs

Vardag	<input checked="" type="checkbox"/>	Utförligare information
Kväll/helg	<input checked="" type="checkbox"/>	

Räkna  

Servicesedelns värde Exportera som PDF

Vardag - 19,45 € / timme
Kväll/Veckoslut - 27,45 € / timme

Bild 13. Ikonerna för skriv ut och exportera till PDF.

Skriv ut läget använder sig av ett utskriftvänligt format och listar alla uträkningar som finns i sessionen. Alla uträkningsdetaljer som lagras i sessionen visas också i utskriftsformatet. PDF exporteringen baseras på en färdig klass som är utvecklad av R&OS och är tillgänglig under öppen källkod [8]. PDF exporteringen har samma layout och skriver ut samma detaljer som skriv ut funktionen.

6.4 Administratörssektion

Administratörssektionen används för att fasta värden som uträkningarna baserar sig på skall kunna editeras i framtiden av någon som är ansvarig för webbtjänstens drift utan att en person som kan hantera MySQL databaser behöver anlitas. Det finns också möjligheter att editera alla textsträngar på finska och svenska som visas för besökare. Administratörssektionen använder samma layout som den publika sektionen med uppdelning genom två stycken tabbar och samma färgschema.

För att skydda administratörssektionen från otillbörliga besök skyddas sektionen med användarnamn och lösenord. AJAX principen som används i den publika sektionen används också i administratörssektionen så att data både hämtas och skrivs till databasen utan omladdningar. Förutom detta så används Java Script för att göra inmatning av informationen mera interaktiv.

6.4.1 Ändra översättningar

Under ändra översättningar kan en person med administratörsrättigheter enkelt editera textsträngar på både finska och svenska. Textsträngarna är uppdelade på rader så att på en rad visas rubriken för textsträngen och till höger innehållet på finska och svenska. Administratören editerar textsträngen genom att klicka på skiftnyckeln bredvid den rad som önskas editeras. De båda textrymderna för finska och svenska blir då öppna och editierbara. Då en rad med textsträngar är editierbar går det inte att klicka på en skiftnyckel för en annan rad, dessa göms tills administratören sparar informationen för den aktuella rad som editeras.



	Ändra översättning	Ändra värden
Rubrik	Finska - Suomi	Svenska - Routsii
welcome_msg	Tervetuloa	Hej
help	Apua	Hjälp
	Tulokset oli:	Servicesedelns värde är:

Bild 14. Utseende för administratörssidan och tabben Ändra översättningar.

För att spara en rad med textsträngar som har blivit editerad så klickar administratören på spara ikonen som nu syns istället för en skiftnyckel längst till höger på raden. Egentligen så körs funktionen som sparar informationen till databasen redan då användaren klickar sig ur textrymden men jag lade in en spara ikon för att administratören skulle få en bättre respons från gränssnittet att informationen faktiskt har sparats. Samtidigt som administratören klickar på spara ikonen så låses raden med textstängarna och sidan går tillbaka till ursprungsläget där skiftnycklar visas bredvid varje rad igen.

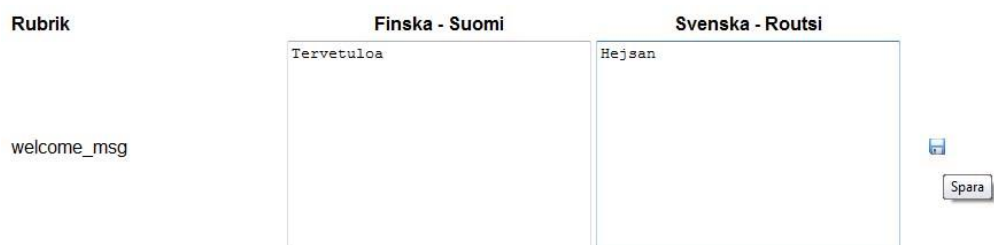


Bild 15. Skiftnyckeln byts ut mot en spara knapp.

6.4.2 Ändra värden

Administratören kan under ändra värden tabben ändra sådana fasta procentsatser och värden som uträkningarna baserar sig på. De befintliga värdena visas under tre olika rubriker för varje tabell som informationen hämtas från: maksutaulukko, maksutaulukko_hinta och kotihoido_raja. Editeringen av informationen fungerar efter samma princip som under tabben Ändra översättningar, närmare beskrivning gavs under föregående rubrik 6.4.1. Administratören klickar på skiftnyckeln bredvid raden som den vill editera vilket gör raden editierbar och värdena sparas då administratören klickar på spara ikonen vilket också låser den aktuella raden.



Ändra översättning
Ändra värden

Värden som används till både servicesedel och kommunal hemvård

Perheen koko	Bruttotuloista 1	2	3	4	5	6	
1	<input type="text" value="520"/>	<input type="text" value="20,00"/>	<input type="text" value="25,00"/>	<input type="text" value="31,00"/>	<input type="text" value="33,50"/>	<input type="text" value="34,50"/>	<input type="text" value="35,00"/>
2	<input type="text" value="959"/>	<input type="text" value="12,57"/>	<input type="text" value="15,71"/>	<input type="text" value="19,48"/>	<input type="text" value="21,06"/>	<input type="text" value="21,69"/>	<input type="text" value="22,00"/>
3	<input type="text" value="1504"/>	<input type="text" value="10,28"/>	<input type="text" value="12,85"/>	<input type="text" value="15,94"/>	<input type="text" value="17,23"/>	<input type="text" value="17,74"/>	<input type="text" value="18,00"/>
4	<input type="text" value="1860"/>	<input type="text" value="8,57"/>	<input type="text" value="10,71"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
5	<input type="text" value="2252"/>	<input type="text" value="7,43"/>	<input type="text" value="9,28"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
6	<input type="text" value="2585"/>	<input type="text" value="6,28"/>	<input type="text" value="7,86"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Värde för tidpunkt (servicesedel)

arki

itita/viikonloppu

Max och minigräns / timme

kotihoido max

palveluseteli min

Bild 16. Utseende för administratörssidan och tabben ändra värden.

Alla värden förutom de som finns i tabellen maksutaulukko och kolumnen bruttotuloista kräver två decimaler. Om administratören inte specificerar dessa decimaler så läggs automatiskt två nollor som decimal, detta sköts automatiskt från databasen. Decimaltecken som accepteras av databasen är enbart det engelska decimaltecknet ”.”, om administratören skriver in den nordiska motsvarigheten ”,” så bytts detta tecken ut mot den engelska varianten före informationen sparas i databasen med en strängutbytningsfunktion genom PHP.

7. PROGRAMMERINGSTEKNISK FÖRKLARING AV UTRÄKNING

I detta kapitel kommer jag ytterligare att beskriva hur uträkning av resultatet går genom hela kedjan av funktioner från att användaren matar in information tills att resultatet visas för användaren. Uträkningen görs för räkneformuläret kommunal hemservice och hemsjukvård, uträkning för servicesedel fungerar enligt samma princip med några ändringar. Uträkningar för servicesedel kommer inte att beskrivas. Beskrivningen förutsätter att läsaren är någorlunda insatt i programmering.

I räkneformuläret för kommunal hemservice och hemsjukvård kan man likt i uträkningen för servicesedel välja storlek på boendet men för kommunal hemservice skall antal timmar som man antas behöva hemhjälp också väljas. Värdet bakom antal timmar är ett procenttal som ändras beroende på storlek på boendet. Detta innebär att man först måste välja storlek på boendet.

Rullgardinslistan `select_familj_storlek` har ett `onchange` attribut som kör Java Script funktionen `showTimmar()` då man väljer ett värde. Detta Java Script hämtar id värdet för storleken på familjen från rullgardinslistan man har valt och lägger värdet i en GET sträng som skickas till PHP filen `timmar.php`.

```
function showTimmar()
{
var familj_storlek = document.getElementById('select_familj_storlek').value;

if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  { // code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.onreadystatechange=function()
{
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
  {
  document.getElementById("timmar").innerHTML=xmlhttp.responseText;
  }
}

xmlhttp.open("GET","timmar.php?familj_storlek="+ familj_storlek,true);
xmlhttp.send("");
}
```

Bild 17. Boendestorleken hämtas och skickas till `timmar.php`.

Då timmar.php körs så hämtas procentvärdena från databasen som hör till boende storleken som fås från id värdet i GET strängen. En HTML tabell och en rullgardinslista med timmarna och procentalen byggs upp och visas i en <Div> tagg för användaren. Då användaren har matat in alla sina värden och trycker på räkna så körs först en valideringsfunktion valideraKommunalHemservice() som kontrollerar så att endast numeriska värden har matats in som bruttoinkomst och att ett värde har valts ut varje kontroll. Om valideringsfunktionen hittar ett fel så rapporteras detta med ett felmeddelande bredvid felaktigt ifyllt inmatningsfält och uträkningen avbryts.

Om inga fel hittas så körs ett Java Script showKommunalHemservice() som hämtar all informationen som har matats in av användaren. Alla hämtad information sätts i en GET sträng som skickas till kommunal_hemservice.php. I filen kommunal_hemservice.php hämtas bruttotuloista värdet från tabellen maksutaulukko baserat på boendestorleken. Själva räknefunktionen rakna_kommunal_hemservice() som finns i funktioner.php körs och alla hämtade värden skickas med.

```
//Check that only numbers have been entered
txt[0]=document.getElementById("brutto").value;

for (i = 0; i < txt[0].length && IsNumber == true; i++)
{
  Char = txt[0].charAt(i);
  if (ValidChars.indexOf(Char) == -1)
  {
    IsNumber = false;
  }
}
```

BILD 18. Kontrollerar så att endast numeriska värden har matats in.

I räknefunktionen så hämtas en gräns som används som maxtaxa för timkostnaden för Kommunal hemservice från tabellen kotihoido_rajaa. Från det valda antal timmar som personen behöver hjälp(t.ex. 30-40) så tas max antal timmar med i beräkningen(t.ex. 40).

Detta värde hämtas från tabellen `kotihoido_tunti_max`. Detta görs för att få fram en timkostnad från månadskostnaden eftersom maxtaxan skall jämföras på timbasis.

```
$summa = ($stal-$familj_kostnad)*($row_tunti[$stimmar]/100);
$summa_timme = $summa / $row_tunti2["tuntiraja"];

//If the "Kommunal hemservice" is above a specific value that are collected from the database, the
if ($summa_timme > $row_rajaja["raja"])
{
    $summa_timme = $row_rajaja["raja"];
    $summa = $summa_timme * $row_tunti2["tuntiraja"];
    echo str_replace(".", "", round($summa,2)) . " € ";
    echo $Sample->translate("kommunal_resultat_text", $_SESSION['lang']) . $Sample->display() . "<br>"
    echo $Sample->translate("hour", $_SESSION['lang']) . $Sample->display() . "<br/>";
    echo $Sample->translate("kommunal_limit1", $_SESSION['lang']) . $Sample->bold() . $Sample->display
}
else
{
    echo "<b>" . str_replace(".", "", round($summa,2)) . " € ";
    echo $Sample->translate("kommunal_resultat_text", $_SESSION['lang']) . $Sample->display();
    echo "</b>";
}
}
```

Bild 19. Uträkningen, kontroll av maxtaxa och utskrivning av resultat.

Då resultaten skall presenteras så körs först en kontroll om slutsumma ligger under maxtaxan som hämtades från tabellen `kotihoido_rajaja`. Om resultaten ligger över gränsen så justeras detta genom att resultaten blir maxtaxan. I presentationen nämns också att denna justering har gjorts.

8. SLUTLEDNING

Detta lärdomsprov har beskrivit mitt arbete med kostnadskalkylatorn för seniorer för Vasa stad *seniornät* samt teknikerna som de baseras på. Arbetet utfördes under sommaren 2010. Arbetet har förflutit relativt smidigt under projektets gång. Eftersom detta inte är mitt första projekt inom Open IT Lab där webbtjänster har utvecklats tidigare har de erfarenheterna kunnat nyttjas från tidigare projekt och implementera dem i detta projekt.

Detta har dock varit det första projektet där jag har utvecklat allt själv från grunden. Av denna orsak var det viktigt att tänka på vilka uppgifter jag tog på mig och hur jag tänkte implementera dessa inom en rimlig tidsram. För några funktioner hade jag redan en lösning i princip klar som jag kunde implementera från tidigare projekt medan andra funktioner och lösningar krävde mera arbete som att arbeta med AJAX samt uträkningshistoriken.

Den planering som jag gjorde i början av projektet som inkluderar databasmodell och layout har hållit sig relativt intakt ända tills webbtjänsten slutfördes. Några ändringar har förstås tillkommit och några funktioner har fallit bort då de inte har tillfört något till slutresultatet.

Som webbtjänsten ser ut idag är den klar att tas i bruk. Om det finns behov att i framtiden vidareutveckla den med flera uträkningsformulär eller andra funktionaliteter så är detta nog möjligt eftersom webbtjänsten har en ganska modular uppbyggnad. Varje formulär består av endast en PHP fil och ett nytt uträkningsformulär kan läggas till som en ny tabb. För att sköta eventuella uträkningar måste förstås funktioner skapas men för någon med erfarenheter i PHP och Java Script borde det inte vara några större problem.

Det som kunde ha gjorts annorlunda eller bättre under projektet gång skulle handla om feedback samt ett problem som uppstår då man jobbar på ett projekt själv. Eftersom webbtjänsten riktar sig till seniorer eller deras anhöriga är det en stor chans att målgruppen inte har så stor erfarenhet av hur man använder sig av webbtjänster eller datorer. Då man ensam utvecklar en webbtjänst som inte direkt riktar sig till den målgrupp eller ålderkategori som man själv tillhör kan det vara svårt att sätta sig in i deras situation.

Jag har försökt sätta mig in i deras situation och följa de standarder som finns angående gränssnitt. Jag har dessutom försökt göra allt lite extra tydligt och hålla layouten så ren från extra text och grafik som möjligt. Men då man utvecklar något själv eller också i en liten grupp har jag lagt märke till att man blir fort blind för brister som en utomstående person kanske skulle uppmärksamma. Under detta projekt har jag nog fått feedback då jag behövt från handledarna på Vasa stad men webbtjänsten har hittills enbart testats av oss på Open IT Lab och handledarna på Vasa stad. Jag och handledarna på Vasa stad är nöjda med slutresultatet men det skulle varit en bra idé att inkludera en eller helst flera personer från den målgrupp som faktiskt skall använda webbtjänsten redan från projektets början.

I början av projektet skulle de kunnat komma med råd och tips om vad de tycker är viktigt och hur de önskar att layouten skulle se ut så att webbtjänsten blir så lätt att använda som möjligt. Under projektets gång så kunde gruppen med testpersoner få se regelbundna versioner av webbtjänsten och komma med åsikter och synpunkter. Detta är dock något som borde implementeras från början i projektet och ju senare de sker desto svårare och tidkrävande blir det att göra ändringar i layout och funktionalitet.

Förutom detta så är jag nöjd med slutresultatet och det har varit mycket lärorikt att arbeta med detta projekt och det skall bli intressant att följa med då webbtjänsten tas i bruk.

9. KÄLLFÖRTEKNING

1. PHP.NET 2007. Usage Stats for Juli 2007. [Uppdaterad i juli 2007].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument:
<URL:<http://php.net/usage.php>>
2. Zakas, C Nicholas, McPeak, Jeremy, Fawcett, Joe 2006. Professional AJAX. 1 upplagan. Indianapolis, Indiana, USA . Wiley Publishing Inc.
3. Gilmore, Jason W. 2004. Beginning PHP 5 and MySQL: From Novice to Professional. 1 upplagan. USA . Apress.
4. Jesse James Garrett, 2005. Ajax: A New Approach to Web Applications. [Uppdaterad i 18.2.2005].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument:
<URL:<http://www.adaptivepath.com/ideas/essays/archives/000385.php>>.
5. Ajax Tutorial. Creating client-side dynamic Web pages. [Uppdaterad 2006].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument: <URL:
<http://www.xul.fr/en-xml-ajax.html>>
6. Aaron Swartz, 2005. A Brief History of Ajax. [Uppdaterad 22.12.2005].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument:
<URL: <http://www.aaronsw.com/weblog/ajaxhistory>>
7. PHP.NET. What is PHP?. [Uppdaterad 2010].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument: <URL: <http://www.PHP.net>>
8. R&OS. PDF class[Uppdaterad 2006].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument: <URL: <http://www.ros.co.nz/pdf/>>
9. SeniorNätet. Startside[Uppdaterad 2010].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument: <URL:
http://seniorinetti.vaasa.fi/Pa_Svenska/Framsidan>

10. Open IT Lab. Startsidea[Uppdaterad 2010].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument: <URL: <http://www.openitlab.fi>>
11. Google dokument. Skapa och dela ditt arbete online med Google Dokument [Uppdaterad 2010].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument: <URL: <http://docs.google.com>>
12. XMLHttpRequest. W3C Candidate Recommendation [Uppdaterad 3.8.2010].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument: <URL: <http://www.w3.org/TR/XMLHttpRequest/>>
13. XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition). A Reformulation of HTML 4 in XML 1.0 [Uppdaterad 1.8.2002].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument: <URL: <http://www.w3.org/TR/xhtml1/>>
14. Dan Connolly, 2000. A Little History of the World Wide Web [Uppdaterad 2000].[hänvisning 11.11.2010]. Tillgänglig i form av www-dokument: <URL: <http://www.w3.org/History.html>>