



TAMPEREEN TEKNILLINEN YLIOPISTO

*Tietotekniikan koulutusohjelma*

HANNU KUKKONEN

MONIMUTKAISEN WEB-HAKULOMAKKEEN KORVAAMINEN  
TEKSTIHAULLA

Diplomityö

Tarkastaja: professori Hannu Jaakkola  
Tarkastaja ja aihe hyväksytty  
Tieto- ja sähkötekniikan tiedekunta-  
neuvoston kokouksessa 5. marraskuu-  
ta 2014

## TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tietotekniikan koulutusohjelma

**KUKKONEN, HANNU:** Monimutkaisen web-hakulomakkeen korvaaminen tekstihaulla

Diplomityö, 44 sivua

Lokakuu 2014

Pääaine: Ohjelmistotekniikka

Tarkastaja: professori Hannu Jaakkola

Avainsanat: web-hakulomake, tekstihaku, vaihtoautot, JavaScript, PHP, PostgreSQL, JSON, Ruby on Rails

Tekstihaun käyttö graafisten käyttöliittymien rinnalla on yleistymään päin. Tässä työssä tutkitaan, onko monimutkaisen web-hakulomakkeen korvaaminen mahdollista pelkällä tekstihaulla ja onko hakujen tekeminen nopeampaa ja helpompaa, kun hakuehdot kirjoitetaan. Tätä työtä varten luotiin verkkopalvelu, jossa vaihtoautoja voi hakea käyttämällä sekä web-hakulomaketta että tekstihakua. Hakutapojen vertailua varten luotiin käyttäjätesti molemmille lomakkeille. Käyttäjätestien tulosten perusteella voidaan todeta, että yhtä suuri osa testihenkilöistä piti graafista hakulomaketta ja tekstihakua helpompana ja nopeampana tapana suorittaa hakuja. Graafisen hakulomakkeen korvaaminen tekstihaulla siis onnistui. Jatkotoimenpiteenä voidaan suositella lisätä monimutkaisten graafisten web-hakulomakkeiden rinnalle tekstihakua, koska palvelua jatkuvasti käyttävät oppivat käyttämään sitä ajan myötä ja hyötyvät sen tuomasta käyttönopeudesta. Samalla ne, jotka pitävät graafisesta hakulomakkeesta voivat käyttää sitä edelleen.

## ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

**KUKKONEN, HANNU:** Replacing a long and complex GUI web search form with a text search

Master of Science Thesis, 44 pages

October 2014

Major: Software Engineering

Examiner: Professor Hannu Jaakkola

Keywords: GUI Web search form, text search, used cars, JavaScript, PHP, PostgreSQL, JSON, Ruby on Rails

The use of text only search in parallel to graphical web search forms is increasing. In this thesis the replacing of a complex web search form with text only search form is studied and the speed and ease of use of text search is compared to a complex graphical web form search. A web service for used cars was created where a user can search with graphical web search form or text search. A user test was created that covers both the graphical web search form and the text search form. The user test was then performed by test users. Based on the results of user testing one half of the test users considered the graphical web form to be better and easier to use and the other half considered the text only search to be easier and faster to use. This also means that text search form could replace the complex web search form. Based on the results, it is suggested that there should be a text search along side with complex web search form, because people will learn to use the text search with time and will benefit greatly from the speed and easy of use it brings. It is also possible for the people who like the complex web search form to continue using it.

## ALKUSANAT

Idea tämän diplomityön tekemiseen syntyi, kun selasin vaihtoautoja eräässä kotimaisessa vaihtoautojen hakupalvelussa. Koin haussa käytettävän lomakkeen pituuden ja monimutkaisena ja hitaana käyttää. Ohjelmistosuunnittelijan työssäni olen oppinut hyödyntämään enemmän näppäimistöä kuin hiirtä tietokoneen graafisen käyttöliittymän ohjaamiseen ja eri ohjelmien käynnistämiseen. Näppäimistön käyttö on nopeampi tapa ohjata tietokonetta. Näppäimistön jouhevaan käyttöön on kuitenkin kulunut paljon aikaa. Mieleeni tuli ajatus, että voisiko pelkkä tekstikenttä korvata monimutkaisen graafisen webhakupomakkeen.

On ollut haastavaa löytää aikaa ja tarmoa viedä tämä työ päätökseen päivätyön ja perheen ohella. Onnekseni vierellä on ollut ihmisiä, joiden tuella työ on saatu päätökseen.

Haluan kiittää professori Hannu Jaakkolaa asiantuntevista kommentteista ja ohjauksesta. Kiitän erityisesti vaimoani tuesta ja kannustamisesta. Kiitän myös vanhempiani, ystäviä ja työkavereita tuesta ja ymmärryksestä.

Ylöjärvellä 22.10.2014

Hannu Kukkonen

# SISÄLLYS

1	Johdanto .....	1
2	Verkkopalvelun hakutavat .....	3
2.1	Historiaa – tietokoneen käyttöliittymät .....	3
2.2	Verkkopalvelu .....	6
2.3	HTML .....	7
2.4	Selaimet .....	11
2.5	HTML-lomakkeen elementit .....	12
2.6	Vaihtoehtoiset käyttöliittymät .....	16
2.6.1	Kosketuskäyttöliittymä .....	16
2.6.2	Puhekäyttöliittymä .....	17
2.7	Graafinen hakulomake .....	18
2.8	Tekstihaku .....	19
2.9	Graafinen web-hakulomake tekstihakuun verrattuna .....	20
2.10	Vaihtoautohaun tietojärjestelmä .....	21
3	Hakulomakkeiden toteutus .....	23
3.1	Palvelin .....	23
3.2	WWW-palvelinohjelma .....	25
3.2.1	Nginx .....	25
3.2.2	Apache HTTP Server .....	26
3.3	PHP ja muita skriptikieliä .....	26
3.4	Tietokanta .....	29
3.5	Ruby on Rails .....	30
3.6	JavaScript .....	31
3.7	Monimutkainen web-hakulomake .....	32
3.7.1	HTML .....	32
3.7.2	AJAX .....	33
3.7.3	CSS .....	33
3.8	Tekstihakulomake .....	34
3.8.1	Toiminta .....	35
3.8.2	JSON .....	35
3.8.3	Säännöllinen lauseke .....	35
3.8.4	Sumea hakutekniikka .....	36
3.8.5	Ennakoiva ja passiivinen haku .....	36
4	Käyttäjätetit .....	38
4.1	Testin valmistelu .....	38
4.2	Testin kuvaus .....	38
4.3	Käyttäjäprofiilit .....	39
5	Tulokset ja niiden tarkastelu .....	40
6	Johtopäätökset .....	41
	Lähteet .....	43

## Lyhenteet ja merkinnät

GUI	Graphical User Interface, graafinen käyttöliittymä
WWW, WEB	World Wide Web, verkkopalvelutyyppi maailmanlaajuisessa tietoverkossa
HTML	HyperText Markup Language
CSS	Cascading Stylesheets, tyylimäärittäminen
PHP	PHP: Hypertext Preprocessor. Kommentoarjakieli, jota käytetään web-sovellusten tekoon.
JSON	JavaScript Object Notation, tiedon tallennusmuoto
RegExp	Regular Expression, säännöllinen lauseke
SQL	Structured Query Language, tietokannan kyselykieli
NoSQL	Not only SQL, tietokantatyyppi

# 1 Johdanto

Graafinen käyttöliittymä, GUI, suunniteltiin helpottamaan tietokoneen käyttöä. Graafisessa käyttöliittymässä tietokoneen käsitteet ovat nähtävissä näytöllä. Tiedostot, kansiot ja esimerkiksi roskakori piirretään näytölle kuvakkeina, jolloin käyttäjälle muodostuu konkreettisuus tietokoneen toiminnasta. Tietokoneen käyttäjä ohjaa hiirellä näytöllä näkyvää osoitinta ja hiiren näppäintä klikkaamalla tietyissä paikoissa pystyy tekemään valintoja ja käynnistämään ohjelmia. Kun käyttäjä haluaa poistaa tiedoston, raahaa hän tiedoston ikonin osoittimen avulla roskakoriin. Ennen graafista käyttöliittymää tietokoneetta ohjattiin kirjoittamalla komennot tai syöttämällä komennot reikäkortin avulla tietokoneeseen. Komentojen syöttäminen reikäkortilta tai komentojen kirjoittaminen tekstikäyttöliittymään vaati järjestelmän syvällisempää tuntemusta. Tiedoston poistaminen komentoriviltä on hankalampaa - käyttäjän tulee osata siirtyä hakemistoon, missä poistettava tiedosto on ja sen jälkeen syötettävä komento virheettömästi tiedoston poistamiseen. Jos tiedoston nimi sattuu olemaan pitkä tai sisältää erikoismerkkejä, kuten välilyönnin, voi käyttäjältä kulua paljonkin aikaa ennen kuin poisto onnistuu. Tekstikäyttöliittymä saattaa aiheuttaa käyttäjän mielessä kysymyksiä: ”Miten rajaan hakua, mitkä ovat komennot, mistä löydän ohjeet?”.

Graafiset käyttöliittymät ovat helposti hahmotettavissa ja ovat selkeitä käyttää. Toisinaan vastaan tulee graafisia käyttöliittymiä, jotka pituudellaan ja koollaan saattavat sekoittaa ja hämmentää käyttäjää. Yksi esimerkki tällaisesta käyttöliittymästä on vaihtautojen hakupalvelu tietoverkossa. Autoissa on paljon ominaisuuksia ja toiminnallisuksia, minkä takia hakukriteereitä on paljon. Kun hakukriteereistä luodaan omat graafiset elementit hakusivulle, tulee hakusivusta hyvin pitkä ja ensi katsomalla vaikeaselkoinen. Vaikka hakusivua käyttäisikin hyvin usein, saattaa hakujen tekeminen tuntua työläältä, kankealta ja hitaalta, koska monen hakukriteerin valinta vaatii monen graafisen elementin käsittelyä hiirtä ohjaamalla ja klikkaamalla, sekä pitkähkön hakulomakkeen tapauksessa sivulla siirtymistä sivupalkkia liikuttamalla.

Tämä tutkielma on tehty, koska eräässä verkkopalvelun hakutoiminnon käyttämisessä havaittiin ongelmia käytön nopeuden ja selkeyden suhteen. Sama ongelma voi vaivata myös muiden verkkopalvelujen hakutoimintoja, joissa on käytössä pitkä ja monielementtinen graafinen hakulomake. Toiminnan nopeuttamisen ja selkiyttämisen takia haluttiin tutkia, olisiko mahdollista selkiyttää ja nopeuttaa tiedonhakua verkkopalvelussa.

Alankomaalaisessa Twenten yliopistossa on tutkittu tekstihaun käyttämistä vaihtoehtoisena tapana web-hakulomakkeelle matkansuunnittelusovelluksessa. Tutkijoiden tuloksista ilmenee, että käyttäjät pitivät tekstihakua parempana tapana käyttää sovellusta ja haun suorittaminen tekstihaulla oli keskimäärin 9 % nopeampaa. (Tjin-Kam-Jet et al. 2011)

Tämän työn tavoitteena on selvittää, voiko monimutkaisen, graafisen hakulomakkeen korvata tekstihaulla ja onko hakujen suorittaminen nopeampaa ja helpompaa käyt-

tämällä tekstihakua. Tutkielmassa käsitellään tiedonhakua web-käyttöliittymässä ja vertaillaan graafista hakulomaketta tekstihakulomakkeeseen.

Tutkimuskohteena käytetään vaihtoautojen hakupalvelua, jonka käyttöä arvioidaan empiirisellä testillä. Testihenkilöiden käyttöön luodaan vaihtoautojen hakupalvelu, jossa on sekä graafinen hakulomake että tekstihaku. Testihenkilöt, joilla on kokemusta tietokoneen ja Internet-selaimen käytöstä, suorittavat hakuja ennalta laaditun testiohjeen mukaisesti. He käyttävät, testaavat ja arvioivat palvelun hakulomakkeet.

Tutkielman rakenne on seuraavanlainen: luvussa kaksi on tutkielman teoreettinen osuus ja luvuissa kolme ja neljä on työosuus. Luvussa kaksi käydään läpi graafisen käyttöliittymän historiaa, HTML-kielen ominaisuuksia, esitellään graafisen hakulomakkeen ja tekstihaun teoriaa, sekä esitellään muita käyttöliittymiä. Luvussa kolme käydään läpi työn tekemiseen liittyviä teknologioita ja varsinaista työn tekemistä. Luvussa neljä esitellään, miten vaihtoautopalvelun hakulomakkeita testataan, miten testi on valmisteltu, testitapaukset ja testaajaprofiilit. Luvussa viisi esittelen testin tulokset ja pohdin niiden merkitystä. Luvussa kuusi pohdin tutkielman onnistumista ja esitellään jatkokehitystarpeet.



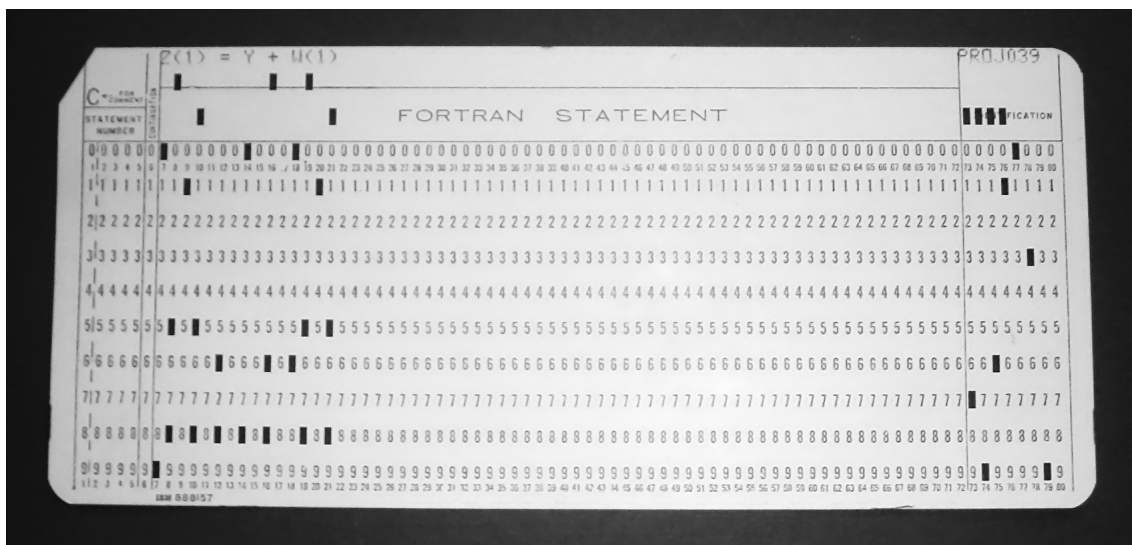
## 2 Verkkopalvelun hakutavat

Graafiset käyttöliittymät ovat helppoja käyttää, koska graafiset elementit muodostavat näyttöruudulle selkeän ja tutun käyttöliittymän. Tutkielma lähtökohtana on, että joissain tapauksissa on luotettu liikaa graafisen käyttöliittymän helppouteen ja käyttöselkeyteen. Graafisen käyttöliittymän toimivuuden voi helposti pilata panemalla samalle sivulle liikaa graafisia elementtejä, jolloin sivusta tulee pitkä ja hankala käsittää. Vaikka palvelua käytettäisiin toistuvasti ja käyttöliittymä tulisi käytön myötä hyvin tutuksi, niin käyttäjälle olisi helpompaa ja nopeampaa syöttää hakuehdot näppäimistöllä yhteen tekstilaitteeseen.

Tässä luvussa käydään läpi tietokoneiden käyttöliittymän historiaa, verkkopalvelun mahdollistavia tekniikoita, esitellään Twenten yliopistossa tehty vastaava tutkimus (Tjin-Kam-Jet et al. 2011) ja käydään läpi vaihtoehtoisia tietokoneen käyttöliittymiä.

### 2.1 Historiaa – tietokoneen käyttöliittymät

Ensimmäisiä tietokoneita käytettiin syöttämällä komennot reikäkortilta. Se oli hidasta ja hankalaa. Tietojen syöttämistä ja tulkitsemista varten käyttäjän piti opetella ja tietää paljon yksityiskohtaista tietoa. Kuvassa 1 on tietokoneen ohjelmoimiseen tarkoitettu kortti. Reikäkorteissa ja -nauhoissa olevat reiät, tai pikemmin reikärivit sisälsivät tietokoneiden ymmärtämiä ohjeita.



Kuva 1: Reikäkortti

Reiät kortteihin ja nauhaan luotiin erillisellä rei'itinlaitteella. Reikäkortit syötettiin lukijalaitteeseen, joka luki korttien sisältämän ohjelman ja tiedon keskusmuistiin. Reikänauha syötettiin tietokoneen reikänauhan lukijaan, joka luki ohjelman keskusmuistiin. Tietokone suoritti ohjelman ja tulos tulostettiin paperille. Kuvassa 2 on reikänauha, jota käytettiin tiedon ja ohjelmien syöttämiseen tietokoneeseen.



**Kuva 2: Reikänauha**

Reikänauhaa ja -korttia edistyneempi tapa on päätekäyttöliittymä, Command Line Interface. Päätekäyttöliittymä kehittyi 1960- ja 1970-lukujen vaihteessa kun kaukokirjoittimesta (Teleprinter) pystyttiin muodostamaan yhteys tietokoneeseen. Lyhenne TTY, mikä tarkoittaa Teletype-yhtiön valmistamaa laitetta, on edelleen käytössä Unix-käyttöjärjestelmissä. Teletype-yhtiö valmisti kaukokirjoittimia, joita käytettiin paljon myös päätelaitteena. Kaukokirjoittimia oli käytetty tiedon välitykseen jo aiemmin. Laitteella voidaan muodostaa yhteys ohjelman suorittavaan tietokoneeseen ja syöttää komentoja näppäimistöltä, mikä nopeutti paljon ohjelmien suorittamista. Laitteilla näppäily teksti ja ohjelmien tulokset tulostuivat paperille. Paperia kului paljon, kunnes 1970-luvun puolivälissä päätelaitteeseen yhdistettiin näyttö. Paperille tulostamisen sijaan kirjoitettu teksti ja ohjelman tulos näkyi näytöllä.

Kuvassa 3 on kuvakaappaus tekstikäyttöliittymästä. Siinä ei ole muuta kuin tekstiä ja tekstikursori, joka ilmaisee, mihin kohtaan teksti ilmaantuu kirjoittaessa. Tekstissä voi olla useita eri värejä.

```

$ cat > wikipedia.pas
program wikipedia;

uses SysUtils;

begin
  WriteLn('Wikipedia is so cool');
end.
cat:

Felipe@mr-69ypraxr6d97 ~
$ fpc wikipedia.pas
Free Pascal Compiler version 2.0.0 [2005/05/08] for i386
Copyright (c) 1993-2005 by Florian Klaempfl
Target OS: Win32 for i386
Compiling wikipedia.pas
Linking wikipedia.exe
7 Lines compiled, 0.3 sec

Felipe@mr-69ypraxr6d97 ~
$ ./wikipedia.exe
Wikipedia is so cool

Felipe@mr-69ypraxr6d97 ~
$ =

```

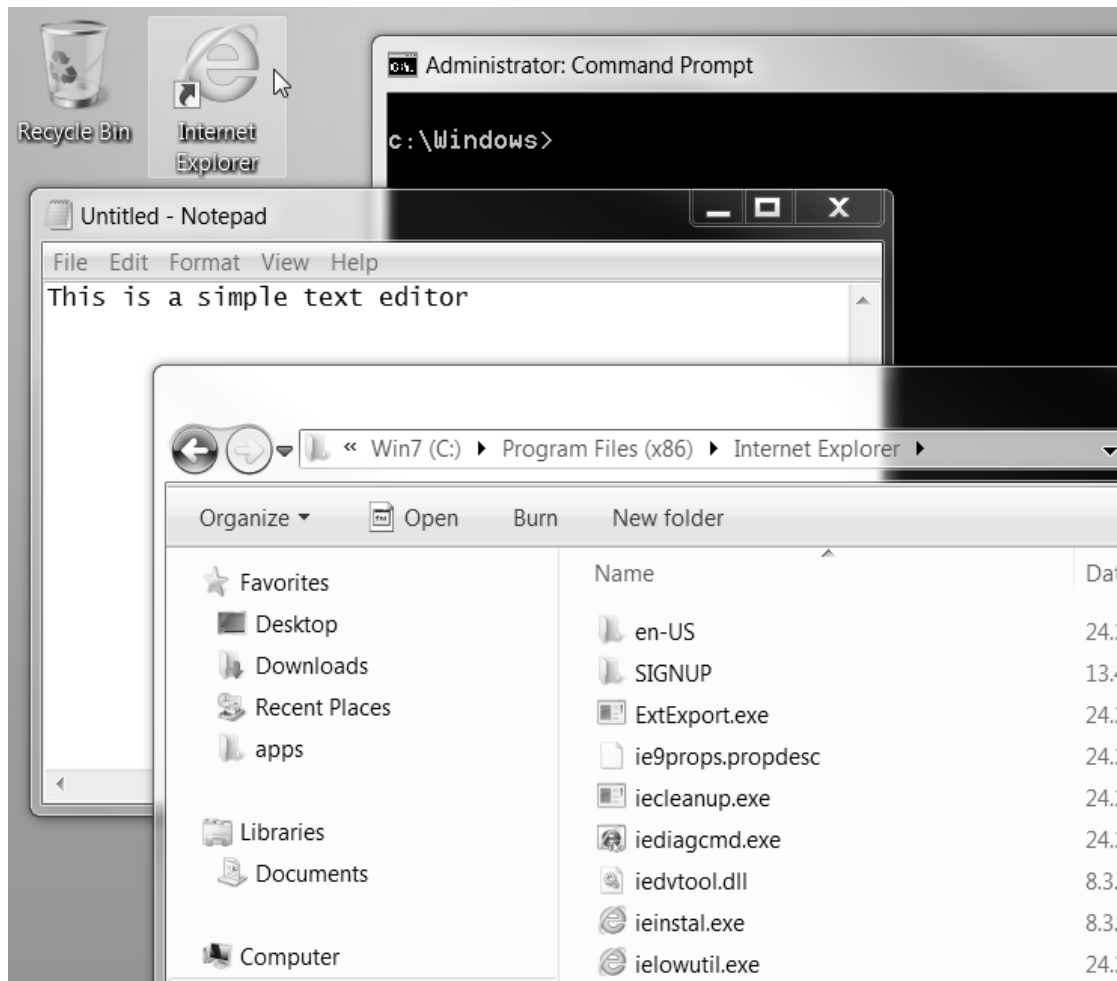
Kuva 3: Tekstikäyttöliittymä

Tekstikäyttöliittymä koostuu näppäimistöstä ja näytöstä. Käyttäjä kirjoittaa komennot ja tiedot näppäimistöltä ja näkee kirjoittamansa tekstin ja tietokoneen palauttavat ilmoitukset näytöltä. Koska käyttöliittymä ei suoraan tarjoa mitään tietoa järjestelmän toimintoista tai käsitteistä, on käyttäjän vastuulla tietää, miten järjestelmää käytetään, eli mitä komentoja järjestelmä tukee ja mikä niiden muoto ja syntaksi ovat. Koska käyttöliittymä ei tarjoa mitään visuaalista apua ja käyttäjän pitää opetella koneen käyttöä erikseen, tekstikäyttöliittymiä pidetään hankalina käyttää.

Tietokoneen käyttöä nopeuttamaan keksittiin graafinen käyttöliittymä, Graphical User Interface (GUI). Ensimmäinen tietokone, johon se toteutettiin, oli Xerox Alto, joka julkaistiin vuonna 1973. Xerox Altossa toteutui graafinen käyttöliittymä näyttöruudulla ja käyttöliittymää käytettiin hiirellä ja näppäimistöllä.

Graafinen käyttöliittymä tarjoaa hyvin intuitiivisen tavan tietokoneen käyttämiseen. Tietokoneen ohjaukseen käytetään näyttöpäätettä, hiirtä ja näppäimistöä. Ruudulla hiiren liikkeitä vastaa osoitin. Käsitteet, kuten tiedosto ja hakemisto, pyritään ilmentämään monitorille kuvakkeina ja ikkunoina.

Kuvassa 4 on nykyaikaisen Microsoft Windows 7 –käyttöjärjestelmän graafisen käyttöliittymän sovellusikkunoita, ikoneita, hiiriosoitin ja komentorivikehote. Graafisen käyttöliittymän yleistymisen myötä kynnys oppia käyttämään muita GUI-laitteita, esimerkiksi tablettitietokoneita, on alempi.

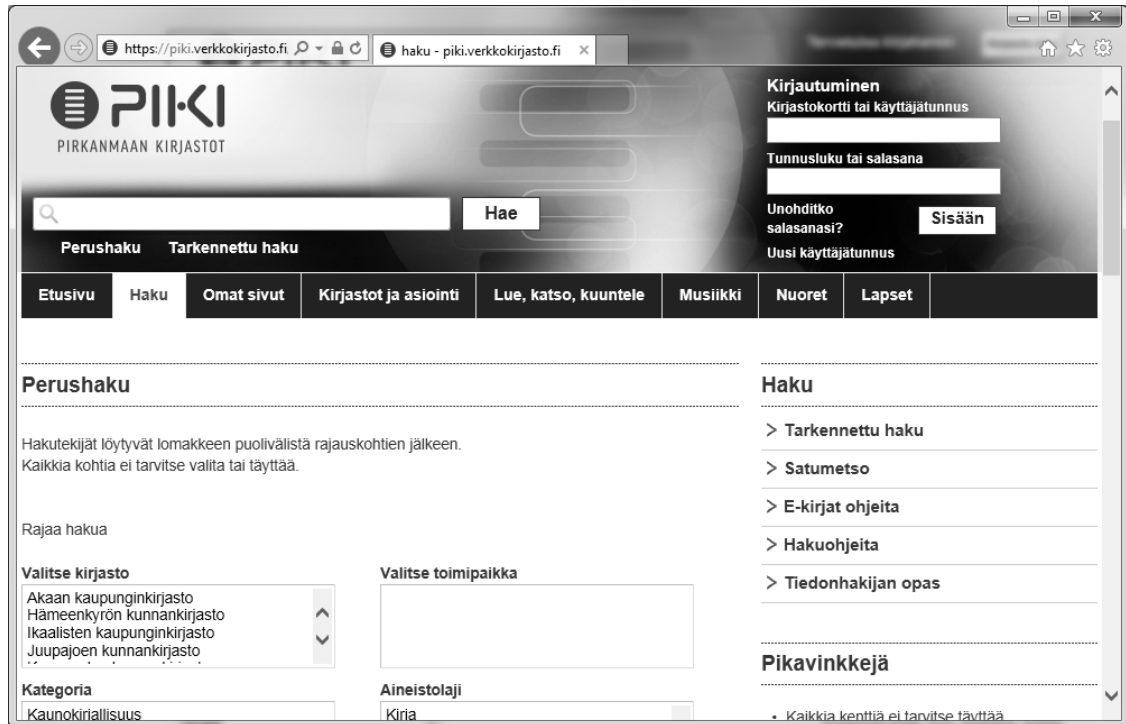


Kuva 4: Windows 7 -käyttöjärjestelmän graafinen käyttöliittymä

## 2.2 Verkkopalvelu

Laite, joka on yhdistetty tietoverkkoon ja joka tarjoaa palvelun, esimerkiksi hypertekstijärjestelmän, tarjoaa verkkopalvelun tietoverkon käyttäjille. Verkkopalvelusta käytetään yleisimmin nimityksiä www-sivut, nettisivut ja kotisivut. Verkkopalvelulla on tiettyjä ominaisuuksia, jotka ovat yhteisiä kaikille verkkopalveluille. Verkkopalvelulla on osoite, kohdeyleisö ja julkaisija. Kohteena voivat olla vain yrityksen omat käyttäjät, jolloin verkkopalvelu on saatavilla ainoastaan yrityksen sisäisessä verkossa. Jos palvelun kohteena ovat kaikki ihmiset, sijaitsee palvelu silloin julkisessa Internetissä. Tyypillisiä verkkopalveluja ovat sähköposti, Internet-puhelut, pankkiasiointi, viranomaispalvelut ja sähköinen kaupankäynti.

Kuvassa 5 nähdään kuvakaappaus Pirkanmaan kirjaston WWW-verkkopalvelusta, jossa on mahdollista hakea kirjaston aineistoja, katsoa aineiston lainaustietoja, tehdä varauksia aineistoon, uusia lainoja, nähdä maksut ja katsoa omia tietojaan.



Kuva 5: Pirkkanmaan kirjaston verkkopalvelu

## 2.3 HTML

HTML on kuvauskieli, jolla voidaan toteuttaa rakenteellisia dokumentteja. Tavallisesti ne ovat web-sivuja, joista voidaan koostaa verkkopalvelu. HTML-spesifikaatiossa on määritelty tunnisteet (tavallisemmin tagit), joilla voidaan määrittää dokumentin osia tietynlaisiksi, esimerkiksi otsikko ja leipäteksti voidaan erottaa tunnisteilla. HTML-dokumentti koostuu tunnisteista ja tekstistä. HTML on staattinen kieli, siinä ei ole dynaamisia ominaisuuksia. Dokumentti pysyy aina samanlaisena. Dokumentteihin voidaan lisätä toiminnallisuutta ja vuorovaikutusta selainten tukemien skriptikielten, kuten JavaScript, avulla. HTML-kielessä on myös tekstin muotoiluun liittyviä tunnisteita, kuten `<font>`, mutta suositeltava käytäntö on käyttää tekstin muotoilussa CSS-tyylikieltä, Cascading Stylesheets, jolla dokumenttien tekstiä ja sivun asettelua voidaan muotoilla. HTML-dokumenttien ulkoasu ja tyyli on helppo yhtenäistää ottamalla käyttöön CSS-tyyliohjetiedosto monessa HTML-dokumentissa. Samalla tyylin päivittämisestä tulee helppoa, koska muutokset tarvitsee tehdä vain yhteen paikkaan. On kuitenkin mahdollista sisällyttää tyylitiedostot HTML-dokumentin sisään. Se kuitenkin hankaloittaa ylläpitoa ja tekee HTML-dokumentin rakenteesta siivottomamman.

HTML:n ensimmäinen versio julkaistiin Tim Berners-Leen toimesta vuonna 1991. HTML:n viimeisin virallinen versio on 4.01, joka julkaistiin vuonna 1999. (W3C 1999). HTML:n seuraavan version, HTML5, työstäminen on aloitettu vuonna 2008 (W3C 2014) ja sen odotetaan valmistuvan vuonna 2014.

HTML5 spesifikaatio sisältää uusia ominaisuuksia HTML-kuvauskieleen. Uusiin ominaisuuksiin kuuluu

- piirtoalusta (canvas),
- video ja ääni,
- sijaintitieto (geolocation),
- offline-sovellukset, selaimen muisti,
- semanttiset elementit,
- lomakkeiden uudet piirteet. (Korpela 2011)

Canvas on piirtoalusta, johon voi toteuttaa piirtotoimintoja JavaScriptillä. Videoita ja äänitiedostoja voi toistaa selaimessa ilman lisäsovellusten asentamista, ne toiminnot on sisällytetty selaimen. Sijaintitieto mahdollistaa sisällön muokkauksen sijaintiin liittyen, esim. yrityksen WWW-sivuilta voi nähdä omaan sijaintiinsa nähden lähimmän toimipisteen. Se on hyödyllinen toiminto etenkin mobiilikäyttäjille. Sovellusvälimuisti mahdollistaa selainsovelluksen käytön, vaikka Internet-yhteys olisi tilapäisesti poikki. Yhteyden palattua tiedot synkronoidaan palvelimelle. Semanttiset elementit ovat kehitetty siitä, että on tunnistettu lukuisilla web-sivuilla käytettävien tietyn tyyppisiä div-elementtejä dokumentin rakenteen muodostamiseen ja niistä on muodostettu omat elementtinsä. Esimerkiksi on käytetty paljon sivunavigointialueena `<div id="nav">` ja lopputunnistealueena `<div id="footer">`. Niistä on luotu `<nav>` ja `<footer>` -elementit. Niin ollen ne selventävät dokumentin lähdekoodia.

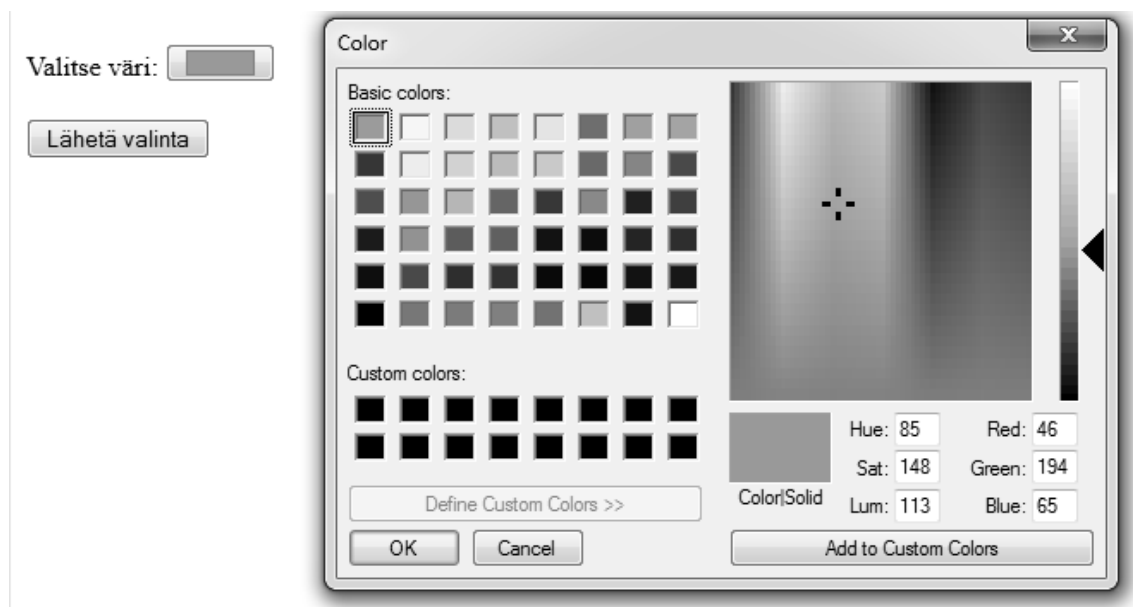
Lomakkeille on luotu uusia piirteitä, uudentyyppiset tietokentät ja tiedon tarkistus. (Ekonoja et al. 2009). Uudentyyppisiä input-elementtejä ovat

- *tel*,
- *search*,
- *url*,
- *email*,
- *datetime*,
- *date*,
- *month*,
- *week*,
- *datetime-local*,
- *number*,
- *range*,
- *color*.

*tel* on lyhennetty sanasta telephone, eli sen tyyppiseen kenttään syötetään puhelinnumero. *search* on hakukenttä, *url*-kenttään syötetään WWW-palvelinosoite, *email*-kenttään syötetään sähköposti, *datetime*-kenttään syötetään päivämäärä ja kellonaika, *date*-kenttään päivämäärä, *month*-kenttään kuukausi, *week*-kenttään viikko, *datetime-local* -kenttään paikallinen päivämäärä ja kellonaika ja *number*-kenttään numero. *range*-tyypillä määritetään syöttökentäksi liukuvalitsin, jota vetämällä arvo määritellään. *color* määrittää syöttökentäksi painikkeen, jossa näkyy valittu väri. Kun painiketta klik-

kaa, aukeaa käyttöjärjestelmäikkuna, jossa värivalinnan voi suorittaa. Selain ilmoittaa lomaketta lähetettäessä, jos annettu syöte ei vastaa kentälle määritetyn tyyppin syntaksia esimerkiksi, jos *tel*-kentässä on tekstiä, mikä ei ole sallittu. Tässä tutkielman työsuo-  
dessa käytetään HTML5-spesifikaatiota, vaikka HTML4.01 on viimeisin virallinen. Voidaan kuitenkin ajatella HTML5:n olevan tarpeeksi hyvin toteutettu jo nykyisissä selaimissa, joten valinta on perusteltu.

Kuvassa 6 nähdään välivalintaelementti `<input type="color">`. Kun värivalin-  
tanappia klikataan, avautuu käyttöjärjestelmän värivalintaikkuna. Kun klikataan *OK*,  
väri tulee valituksi. Tämä helpottaa värin valintaa. Aiemmin väriarvo on syötetty lo-  
makkeella joko RGB (punainen, vihreä, sininen)- tai heksadesimaaliarvona tekstikent-  
tään.



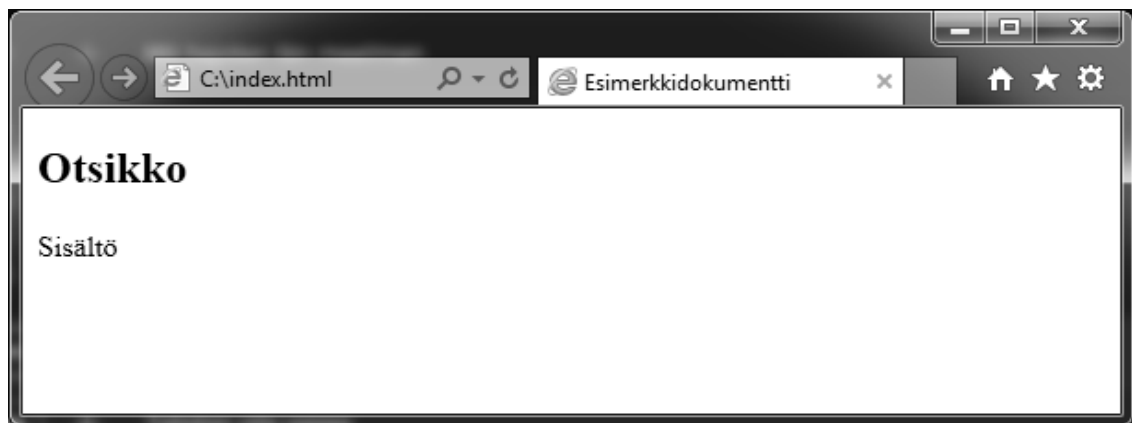
**Kuva 6: HTML-lomakkeen värivalintaelementti**

Uudet input-elementit on kehitetty eritoten mobiililaitteiden Internet-selaimia varten, jolloin tiedon syöttö olisi nopeampaa ja helpompaa kuin kirjoittamalla.

Seuraavassa listauksessa on esimerkki HTML5-dokumentista.

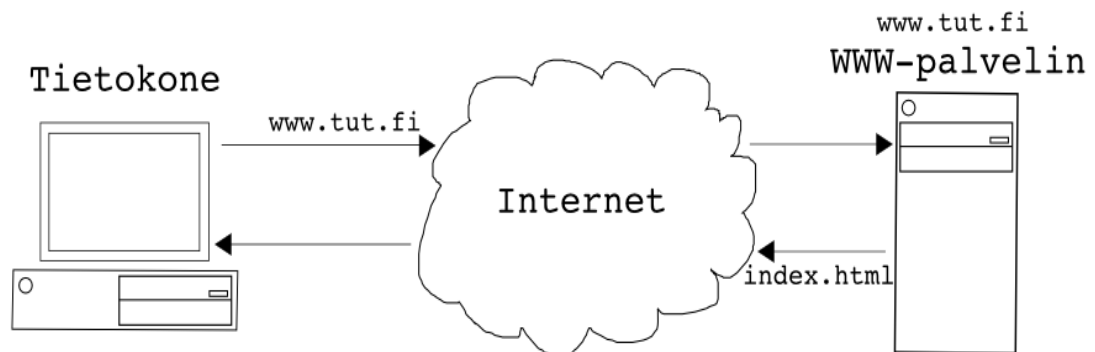
```
<!DOCTYPE html>
<html>
<head><meta charset="utf-8"><title>Esimerkkidokumentti</title></head>
<body>
  <h2>Otsikko</h2>
  <p>Sisältö</p>
</body>
</html>
```

Dokumentissa tulee olla tietyt tunnisteet, jotta se vastaisi HTML5-määrittystä. Dokumentti alkaa dokumenttityypin määrittelyllä `<!DOCTYPE html>`, joka ilmoittaa sen olevan HTML5-dokumentti. Dokumentissa tarvitsee olla vielä muitakin elementtejä, jotta se vastaa HTML5-määrittystä. Elementin `<head>` sisään määritetään dokumenttia kuvaavia tietoja; niitä voidaan antaa `<meta>`-elementillä. Merkitököoodaus on pakollinen ja se määritetään `<meta charset="UTF-8">`. `<body>`-elementti on myös pakollinen. Sen sisään tulee dokumentin selaimessa näkyvä osa, kaikki tekstit ja kuvat. Tekstin otsikko on määritelty `<h2>`-tagin sisään, jolloin se muotoutuu selaimessa olevan määritellyn asetuksen mukaan. Kuvassa 7 näkyy edellisessä listauksessa esitetty yksinkertainen HTML-dokumentti Internet Explorer 11-selaimella avattuna.



Kuva 7: HTML-dokumentti selaimessa

Tässä tapauksessa HTML-dokumentti on avattu saman tietokoneen C-levyltä kuin miltä Internet Explorer on käynnistetty. Tätä tapaa käytetään yleensä kun ollaan toteuttamassa dokumenttia ja halutaan testata, miltä sivu näyttää. Kun dokumenttiin on valmis, niin se siirretään WWW-palvelimelle, josta sivu haetaan. WWW-palvelimella tulee olla Internet-osoite, esimerkiksi `www.tut.fi`, ja WWW-palvelinohjelmisto, joka huolehtii selainten dokumenttipyyntöistä. Kuvassa 8 on esitetty HTML-dokumenttien lataus WWW-palvelimelta.



Kuva 8: HTML-sivujen lataaminen palvelimelta Internetin kautta



Tietokoneen selaimen kirjoitetaan palvelimen osoite ja lataus käynnistetään. Tietokone välittää haun Internetin kautta pyydetylle palvelimelle, joka käsittelee pyynnön. WWW-palvelin palauttaa pyydetyn dokumentin Internetin kautta tietokoneen selaimelle, joka käsittelee palautetun dokumentin ja esittää sen selaimessa.

## 2.4 Selaimet

HTML-dokumenttien katsomiseen oikeassa muodossa käytetään Internet-selainta. Yleisimmät HTML-dokumenttien selaimet ovat Internet Explorer (IE), Google Chrome, Firefox, Safari ja Opera. Eniten käytetyimpien selainten markkinaosuus (NetMarketShare, 2014) oli huhtikuussa 2014 seuraavan taulukon mukainen:

Selain	Osuus prosentteissa
Internet Explorer	58 %
Chrome	18 %
Firefox	17 %
Safari	6 %
Opera	1 %

Taulukko 1: Suosituimpien Internet-selainten osuus huhtikuussa 2014

Microsoftin Windows-käyttöjärjestelmän mukana tuleva Internet Explorer oli käytetyin, 58 prosentin osuudella kaikista selaimista. Googlen Chrome oli toiseksi käytetyin, n. 18 prosentin osuudella. Mozillan Firefox oli kolmantena hieman Chromesta jääneenä 17 prosentin osuudella. Enimmäkseen Applen OSX-käyttöjärjestelmän mukana tuleva, mutta myös Windowsille saatavilla oleva Safari on neljänneksi käytetyin kuuden prosentin osuudella. Norjalainen Opera-selain on viidenneksi suosituin yhden prosentin osuudella.

Internet Explorer, IE, on vanhimpia Internet-selaimia. Se julkaistiin Windows 95 käyttöjärjestelmän Plus!-lisäpaketissa vuonna 1995. IE toimii vain Windows-käyttöjärjestelmässä. Uusin versio on 11.0, joka julkaistiin Windows 7, 8 ja 8.1 – käyttöjärjestelmille vuonna 2013.

IE on yleisin Internet-selain, mutta syy siihen lienee, että se tulee Windows-käyttöjärjestelmän mukana. Windows-käyttöjärjestelmä on maailman käytetyin PC-tietokoneiden käyttöjärjestelmä. IE on saanut maineen hitaasti päivittyvänä, tietoturvatomana, hitaana ja se kärsii uusien web-standardien tuen puutteesta.

IE käyttää ActiveX-teknologiaa lisäosien asentamiseen. Lisäosat ovat sovelluksia, joiden avulla voidaan laajentaa Internet Exploreria sovellusalustana. Esim. peli voi olla lisäosa. Lisäosat asennetaan ActiveX-muodossa, mikä antaa niille tietokoneen resurssit saataville. Se on aiheuttanut paljon tietoturvaongelmia.

Google julkaisi Chrome-selaimensa vuonna 2008. Se toimii Windows-, Linux-, Android- ja OS X-käyttöjärjestelmissä. Uusin versio on 35, joka julkaistiin 20.5.2014.

Chromessa on sisäänrakennettu sovelluksen päivitystoiminto. Päivitys ladataan taustalla automaattisesti, eikä käyttäjältä kysytä mitään päivitykseen liittyvää. Uusi versio tulee käyttöön kun selain käynnistetään uudelleen. Päivityksessä tulleet isommat muutokset esitellään käyttäjälle, jotta käyttäjät olisivat niistä tietoisia. Googlen Chrome-kehitystiimi julkaisee uusia versioita usein. Google Chromea on arvosteltu siitä, että siinä on yksityisyyttä vaarantavia tekijöitä, koska joitakin tietoja lähetetään Googlen palvelimille. Hyviä puolia ovat sovelluksen nopeus, JavaScriptin suoritusnopeus, sisäänrakennettu Flash-soitin ja selainmerkkien tallennus Google-tunnuksella. Lisäohjelmia (laajennuksia) on saatavilla Chromeen Web Store sovelluksen kautta. Aiemmin sovelluksia pystyi asentamaan myös muilta web-sivuilta, mutta ei enää.

Mozilla julkaisi Firefox-selaimen vuonna 2004. Se toimii Windows, Linux, OS X, Firefox OS ja Android -käyttöjärjestelmissä. Se on monen Linux-jakelun mukana tuleva oletusselain. Uusin versio, 29, julkaistiin 21.4.2014. Aiemmin versioiden julkaisu oli hitaampaa, mutta kehitystiimi otti mallia Google Chromen kehitystiimistä ja alkoi päivittää versionumeroa tiheämpään tahtiin. Myös päivitystoiminto on ottanut mallia Google Chrome -selaimen päivitystoiminnosta – päivitys asennetaan selaimen kautta ja uudelleenkäynnistys ottaa uuden version käyttöön. Myös ulkoasu on saanut vaikutteita Chromesta (välilehdet ovat siirtyneet otsikkopalkkiin). Lisäohjelmat on helppo etsiä ja asentaa selaimen ingroidun lisäosan asennusosion kautta.

## 2.5 HTML-lomakkeen elementit

Seuraavaksi käydään läpi HTML:n lomake-elementti `<form>` ja sen sisältämät käyttöliittymäelementit. HTML-lomakkeen etuna on nopea sovelluskehitys. Niillä toteutetaan graafinen hakulomake. Lomakkeen tarkoitus on kerätä käyttäjän syöte (teksti, erilaiset valinnat) ja välittää se käsittelijäohjelmalle jatkokäsittelyä varten. Esimerkkejä lomakkeesta voisi olla vaikka yhteystietojen keräyslomake, jossa on syöttökentät nimelle ja puhelinnumerolle. Lomake määritellään alkavaksi elementillä `<form>` ja lomake lopetetaan elementillä `</form>`. Kaikki elementit, jotka ovat määriteltäviä näiden elementtien sisällä, kuuluvat lomakkeeseen.

Lomakkeella tulee olla määriteltynä käsittelijäohjelma. Se määritetään attribuutilla *action*. Käsittelijä ottaa vastaan lomakkeen tiedot ja prosessoi ne. Useimmiten lomakkeen syötteet validoidaan ja tallennetaan tietokantaan. Tiedon tallentamiseen tarvitaan sitten jo palvelinpäähän käsittelijäohjelma, jolla on tietokantayhteys. Esimerkkinä mainittakoon PHP-skriptikieli.

Kuvassa 9 nähdään esimerkki lomakkeesta, joka toteutettu HTML-kielillä. Lomakkeessa on syöttökentät nimelle ja puhelinnumerolle. Toimintopainikkeilla voi joko lähettää tiedot tai tyhjentää jo syötetyt tiedot. Kentissä on harmaalla tekstillä ehdotukset syötettävän tiedon muodosta, minkä HTML5 mahdollistaa.

## Yhteystietolomake

Nimi

Puhelin

**Kuva 9: Esimerkki HTML-lomakkeesta, jossa on otsikko, kaksi kenttää ja toimintopainikkeet**

Seuraavassa tarkastellaan lyhyesti lomakkeen rakentamisessa käytettävissä olevia erilaisia elementtityyppejä.

### Elementin otsikko ja tekstikenttä

Lomake-elementille voidaan määrittää otsikko tai kuvaus käyttämällä `<label>`-elementtiä. Sen tarkoitus on selventää syötettävän tiedon merkitystä. Se sijoitetaan yleensä lomake-elementin yläpuolelle tai vasemmalle puolelle. Sen tarkoitus on tarkentaa, mikä tieto elementtiin liittyy. Alla on toteutettuna otsikko vuosimallin syöttökentälle.

Vuosimalli

**Kuva 10: Tekstikentälle on annettu otsikko ”Vuosimalli”**

Vuosimalli kirjoitetaan tekstikenttään. Tekstikenttä-elementti on yhden rivin korkuinen, mutta sisältötyypiltään ja pituudeltaan rajoittamaton elementti.

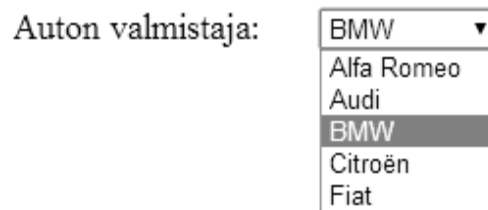
### Valintalista

Valintalista näyttää listan asioista ja siihen voi tehdä yhden tai useamman asian valinnan. Alla on toteutettuna auton valmistajan valinta. Monivalinnalla voidaan hakuun sisällyttää monta arvoa. Kuvassa 11 nähdään valintalista, jossa on auton valmistajia.

Auton valmistaja:

**Kuva 11: Valintalista-elementti**

Valintalistan voi myös näyttää vetovalikkona ja siitä voi valita yhden arvon. Tätä voi käyttää silloin kun halutaan rajoittaa valintojen määrä yhteen. Kuvassa 12 nähdään vetovalikko, jossa on auton valmistajia.



Kuva 12: Vetovalikko (selectbox)

## Valintanappi

Valintanappielementillä on mahdollisuus esittää monta vaihtoehtoa, joista vain yksi on mahdollista valita. Kuvassa 13 näkyvällä valintanappielementillä voi rajoittaa vaihtoehtojen hakua valitsemalla tietty vaihdelaatikkotyyppi. Tähän tietoon sopii hyvin valintanappivalinta, koska autossa on vain yksi vaihteistotyyppi.

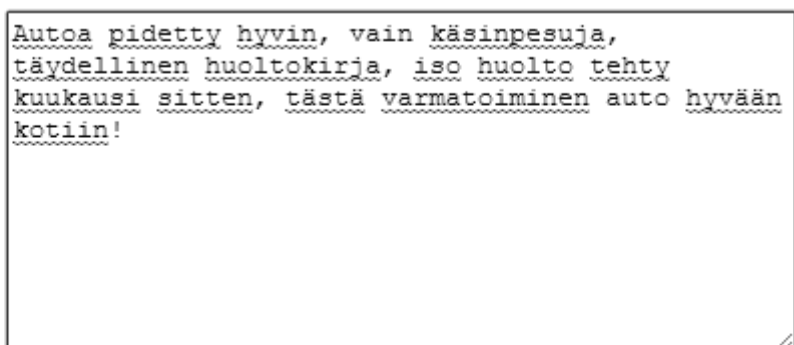


Kuva 13: Esimerkki valintanappi-elementistä

## Laajempi tekstikenttä

Laajempi tekstikenttä -elementtiin voi kirjoittaa useamman rivin tekstiä. Se sopii hyvin vapaamuotoisen ja pitkäkhön tiedon syöttämiseen kuten kuvassa 14 näkyvään vaihtoautopalvelun autotietojen tarkemman kuvauksen antamiseen. Elementin leveys ja korkeus on käyttäjän määritettävissä

### Tarkempi kuvaus



Kuva 14: Tekstialue (textarea) -elementti

## Valintaruutu

Valintaruutu on neliö, jonka sisään ilmestyy ruksi, kun se valitaan. Valintaruutua käytetään yleensä silloin kun vaihtoehtoja on valittavissa useampi. Kuvassa 15 on esitettyä vaihtoautopalvelun graafisen web-hakulomakkeen vaihtoauton lisävarustevalinta.

Lisävarusteet	<input checked="" type="checkbox"/> Vetokoukku	<input type="checkbox"/> Kattoluukku
	<input checked="" type="checkbox"/> CD-soitin	<input checked="" type="checkbox"/> Ilmastointi
	<input type="checkbox"/> Audio-in liitin	<input checked="" type="checkbox"/> Vakionopeuden säädin
	<input checked="" type="checkbox"/> Huoltokirja	<input checked="" type="checkbox"/> Isofix
	<input type="checkbox"/> Peruutustutka	<input type="checkbox"/> Navigaattori

**Kuva 15: Auton lisävarusteet valittavissa valintaruutujen avulla**

Käyttäjä voi sisällyttää lisävarusteen hakuun klikkaamalla valintaruutua. Esimerkiksi jos käyttäjä valitsee ”Ilmastointi”-vaihtoehdon, kaikissa hakutuloksen vaihtoautoissa on ilmastointi. Valintaruudun voi helpoimmillaan toteuttaa seuraavasti:

```
<input type="checkbox" name="extra" value="Vetokoukku">Vetokoukku</input>
```

Valintalogiikka useamman valinnan kohdalla voi toimia kahdella eri tavalla: AND tai OR. AND tarkoittaa, että hakutuloksen tulee sisältää kaikki valitut arvot. OR tarkoittaa, että hakutulos sisältää jonkin valituista arvoista. Sovelluskehittäjä päättää kumpaa tapaa käytetään.

## Piilokenttä

Piilokenttää käytetään välittämään tietoja lomakkeen mukana käyttäjältä piilotettuna, esimerkiksi oletusarvo jollekin kentälle tai JavaScriptillä haettu arvo, esimerkiksi käyttäjän IP-osoite. Piilokenttä tehdään määrittämällä syöttökenttä hidden-tyyppiseksi:

```
<input type="hidden" name="ip" value="193.127.151.163">
```

## Painike

Painiketta (button) käytetään tyhjentämään lomakkeelle syötetyt tiedot (Reset) tai lähettämään tiedot (Submit). Painikkeen voi luoda myös itse, jolloin toiminnon joutuu toteuttamaan myös itse. Painikkeiden tekstin voi myös itse määrittää. Painikkeiden sijainti on yleensä lomakkeen lopussa. Kuvassa 16 on esimerkit tyypillisistä painikkeista.

<input type="button" value="Lähetä"/>	<input type="button" value="Tyhjennä"/>
---------------------------------------	---

**Kuva 16: Lomakkeen tyypilliset painikkeet**

Kuvassa 16 olevat painikkeet on luotu HTML-lomakkeeseen sisäänrakennetuilla `<input type="submit">` ja `<input type="reset">` syöttöelementeillä. Kun halutaan luoda painike erikseen, käytetään `<button>`-elementtiä.

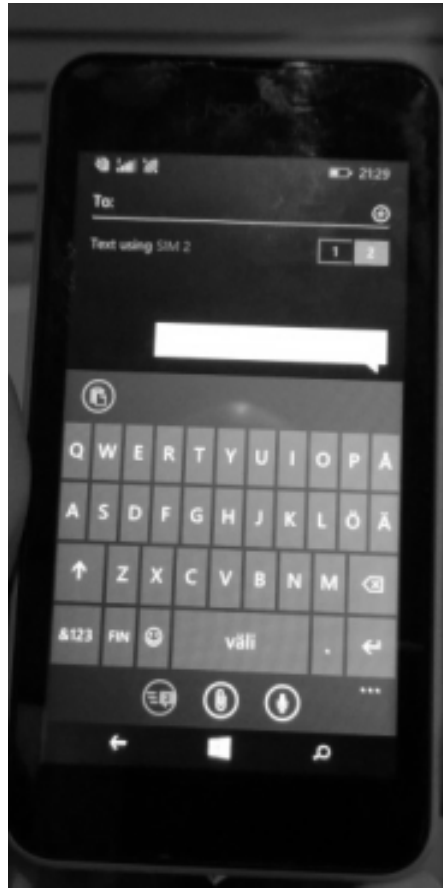
Syöttökentän tyypiksi voidaan määrittää myös puhelinnumero, päivämäärä, sähköposti ja väri. Tällöin selain validoi syötteen syntaksin, että se on varmasti oikeaa muotoa. Erityisesti näistä on hyötyä silloin kun palvelua käytetään mobiiliselaimella. Mobiiliselaimet muuntavat syöttökomponenttinsa näyttämään vain niitä merkkejä, joita on mahdollista kirjoittaa. Mobiililaitteilla kirjoittaminen ei suju yhtä nopeasti kuin oikealla näppäimistöllä, joten tämä helpottaa ja nopeuttaa tiedon syöttämistä.

## **2.6 Vaihtoehtoiset käyttöliittymät**

Tässä kohdassa kerrotaan vaihtoehtoisista tietokoneen käyttöliittymistä. Niitä ei kuitenkaan käytetä tutkielmassa.

### **2.6.1 Kosketuskäyttöliittymä**

Viime vuosina ovat kosketuskäyttöliittymälliset laitteet yleistyneet nopeasti, koska sen mahdollistava tekniikka on halventunut oleellisesti. Laitteessa on kosketuksen tunnistava näyttö ja laitetta ohjataan koskettamalla ruutua sormella tai kosketusnäyttökynällä. Laitteessa voi olla fyysinen näppäimistö tai sitten näppäimistö sisältyy laitteen ohjelmistoon. On kustannustehokkaampaa toteuttaa näppäimistö ohjelmallisesti. Myös näppäinmekaniikkaan liittyvät vikaantumiset jäävät pois. Ruudulla näkyvä käyttöliittymä poikkeaa jonkin verran perinteisestä hiirellä ja näppäimistöllä käytettävästä käyttöliittymästä. Esimerkiksi osoitinta ei ole, kuten tietokoneen näytöllä. Tekstin syöttö tapahtuu näpyttämällä ruudulla näkyvää tekstielementtiä, jolloin näppäimistö ilmestyy ruudulle. Tällaisia laitteita ovat mm. Apple iPhone ja iPad, Android-älypuhelimet ja -tablettitietokoneet sekä Windows 8 -tablettitietokoneet. Eritoten kannettavissa tietokoneissa kosketusnäytöt ovat osaksi korvanneet hiirellä ja näppäimistöllä suoritettavaa ohjausta. Kuvassa 17 nähdään Windows Phone -älypuhelimien kosketusnäyttökäyttöliittymä. Ruudulla näkyy ohjelmallinen näppäimistö, koska käyttäjä on luomassa tekstiviestä, eikä laitteessa ole fyysistä näppäimistöä.



Kuva 17: Kosketusnäyttökäyttöliittymä

## 2.6.2 Puhekäyttöliittymä

Laitteille on kehitetty myös puhekäyttöliittymiä. Laitteessa on mikrofoni, jonka avulla laite voi tallentaa ääntä muistiin ja käsitellä sitä. Tavallisesti se on vain ohjelma, joka käsittelee mikrofonin kautta tulevia ääniä ja yrittää tulkita äänet komennoiksi tai sisälöksi.

Applen iOS-käyttöjärjestelmä tukee puheohjausta iPhone ja iPad –laitteissaan. Siri-puheentunnistusohjelma käynnistetään nappia pitkään painamalla kun halutaan käyttää puhetta laitteen ohjaamiseen. Toimintoja ovat mm. sääennusteen kysyminen, sovelluksen käynnistäminen, tekstiviestin laadinta, sanelu ja lähettäminen, tweetin tekeminen, muistutuksen luominen kalenteriin ja tiedon etsiminen Internetistä. (Apple, 2013). Vastaavanlainen puheohjaustoiminto on myös Windows Phonen Cortana-ohjelmassa (Microsoft, 2014)- ja Android-älypuhelimissa.

Google Glass on laite, joka muistuttaa silmälasia. Sitä voi myös käyttää normaalien silmälasien kanssa. Sitä ohjataan puhekomennolla tai laitteessa olevasta näppäimestä. Google Glass –laite sisältää pienen näyttöruudun, joka tulee käyttäjän näkökenttään kun laite puetaan silmien eteen. Näyttöruudulla näkyy laitteen tila ja käyttöliittymä.



**Kuva 18: Google Glass –laite käytössä**

Kuvassa 18 on henkilö, jolla on Google Glass –laite puettuna. Käyttöliittymä näkyy pienessä ruudussa ja laitetta ohjataan puheella. Kun käyttäjä puhuu: ”Ok glass, take a picture”, niin laite ottaa valokuvan. Muita toimintoja on videon tallentaminen, tiedon haku Internetistä ja tekstin sanelu. Omien web-sovellusten tekeminen on mahdollista Mirror-rajapinnan avulla (Google Developers, 2014). Laite vaikuttaa erittäin hyödylliseltä moniin tarkoituksiin. Laitteen huono puoli on akun huono kestävyys. Koska laite puetaan kasvoille, pitää laitteen olla kevyt ja silloin akku ei saa olla kovin painava. Ras-kasta laitetta ei ole mukava pitää kasvoilla. Google Glass saattaa aiheuttaa ongelmia myös tilanteissa, joissa valokuvaus tai videokuvaaminen ei ole sallittua, kuten yksityisyyttä loukkaavissa tilanteissa. Vaikka käyttäjä ei tallentaisikaan tilannetta millään tavalla, jo laitteen yllä pitäminen aiheuttaa tilanteita, joissa laite koetaan ongelmaksi tai jopa uhkaavaksi.

Puhekäyttöliittymä on hyödyllinen esimerkiksi autoa ajaessa. Tekstiviesti voidaan sanella laitteelle niin, ettei tarvitse kuin aktivoida laitteen puhetunnistus. Puheentunnistus on vielä kuitenkin uusi tekniikka, ja sen käytössä on vielä haasteita. Laitteet eivät osaa vielä tunnistaa kaikkea puhetta ja täydellinen puheentunnistus tuntuu mahdottomalta, koska ihmiset puhuvat eri tavoilla. On murteita, ääntämiseroavaisuuksia ja ää-mentasojen erilaisuuksia. Myös kovaääninen taustamelu haittaa sekä puheentunnistuksessa että palautteen kuulemisessa.

## **2.7 Graafinen hakulomake**

HTML-hakulomakkeella käyttäjä määrittää hakuehdot, joilla hän etsii tietoa. Web-sivun hakulomakkeelle on mahdollista liittää monenlaisia graafisia elementtejä. HTML-kuvauskieli tarjoaa seuraavat elementit: vetolista, painike, tekstikenttä, tekstialue, radionappi, valintaruutu. Syöttökentän tyypiksi voi määrittää mm. hakutekstin, numeron,



puhelinnumeron, WWW-palvelun osoitteen, sähköpostiosoitteen, värin, päivämäärän ja ajan. Näitä käydään läpi tarkemmin kohdassa 2.5.

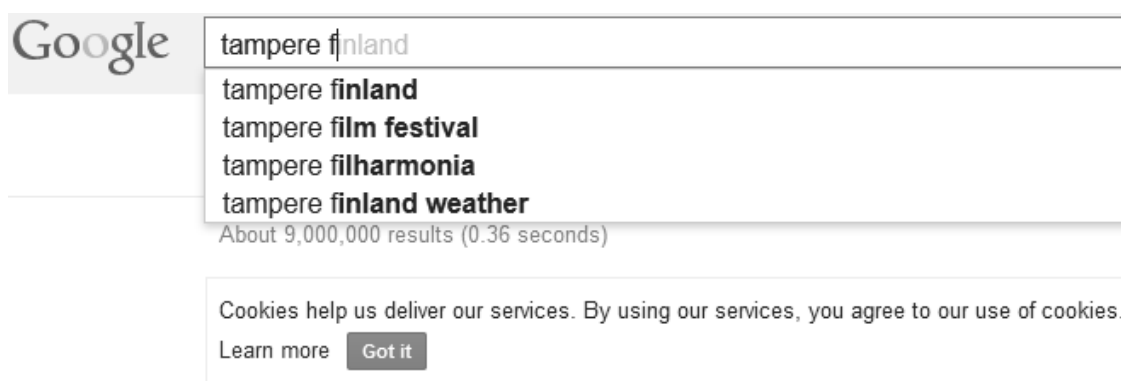
Lomakkeelle on myös mahdollista liittää itse tehtyjä elementtejä. Yleisimmät niistä on toteutettu JavaScript-kielellä. Itse tehtyjen elementtien käyttöönotto tapahtuu sisällyttämällä sen JavaScript-kirjasto `<script>`-tagilla lähdekoodiin. Tällaisia kirjastoja on mm. jQuery-UI. Itse tehtyjä elementtejä ei käytetty tässä tutkielmassa.

## 2.8 Tekstihaku

Tekstihaun toteuttamiseen tarvitaan vain tekstikenttä, johon hakuehdot syötetään. Kun käyttöliittymässä ei ole edes painiketta, jota painamalla haku käynnistyisi, pelkkä Enter-näppäimen painaminen käynnistäisi haun. Selkeyden vuoksi lisätään tavallisesti nappi, jota klikkaamalla haku käynnistyy.

Tekstihakua varten käyttäjän pitää tietää hakusanojen syntaksi, missä muodossa hakuehdot syötetään niin, että järjestelmä osaa tunnistaa ne. Asia vaatii hieman ohjeiden lukemista. Ohjeet tulisivatkin tarjota käyttäjälle selkeästi hakulaatikon viereen.

Hakutulosten näyttäminen voi olla ennakoivaa tai passiivista. Ennakoivalla tarkoitetaan hakuja, jossa haku suoritetaan taustalla kun käyttäjä pitää pienen tauon hakusanojen kirjoittamisessa tai kun selain tunnistaa, että nyt on syötetty uusi hakusana, esim. välilyöntiä on painettu. Käyttäjälle näytetään hakutulokset taustahaun jälkeen ja hakutulokset päivittyvät, aina kun on suoritettu uusi haku taustalla. Passiivisella haulla tarkoitetaan hakuja, jossa haku ei tapahdu taustalla, vaan käyttäjä aktivoi haun, yleensä klikkaamalla haun käynnistävää nappia. Kuvassa 19 nähdään Googlen haku, joka käyttää ennakoivaa hakutaktiikkaa.



### Tampere - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/Tampere ▾

**Tampere** is a city in Pirkanmaa, southern **Finland**. It is the most populous inland city in any of the Nordic countries. The city has a population of 220,609, growing ...

Category:Tampere - University of Tampere - Tampere University of ...

**Kuva 19: Ennakoiva haku**

Käyttäjän kirjoittaessa hakuehtoja, sovellus yrittää tunnistaa niitä ja listaa ne hakuken-  
tän alapuolelle valintalistaksi. Kuvassa 19 näkyy, että hakutulokset on haettu ensimmäi-  
seen sanan perusteella. Hakutulokset ja hakusanojen ehdotukset päivittyvät sitä mukaa  
kun hakuehtoja kirjoitetaan.

Hakuehtojen tunnistamiseen voidaan käyttää myös sumeaa hakua, mikä tarkoittaa,  
että hakusanojen ei tarvitse aivan täysin täsmätä. Hakutuloksia saadaan, vaikka hakusa-  
nan yksi tai kaksi kirjainta on erilaisia tai puuttuvat. Esimerkiksi, jos hakusanasta ”Vol-  
vo” olisi näppäilyvirheen vuoksi jäänyt yksi kirjain pois, hakusana ”Vlvo” palauttaisi  
hakutuloksessa Volvo-merkkiset autot. Tai jos yksi kirjain olisi näppäilyvirheen vuoksi  
väärin, esimerkiksi ”Volvp”, niin haku palauttaisi silti Volvo-merkkiset autot, koska  
sumea haku palauttaa samankaltaiset sanat.

## **2.9 Graafinen web-hakulomake tekstihakuun verrattuna**

On paljon eroja, miten graafista web-hakulomaketta ja tekstihakua käytetään. Graafi-  
sessa hakulomakkeessa joudutaan keskittymään moneen elementtiin ja ne voivat olla  
vuorovaikutuksessa keskenään (esimerkiksi yhden valinnan teko päivittää seuraavan  
elementin sisällön). Tekstihaussa tarvitsee tietää vain syntaksi, jolla hakua voidaan ra-  
joittaa. On selvää, että moneen graafiseen elementtiin keskittyminen ja hiiren osoittimen  
siirtely, käden siirtely näppäimistöltä hiirelle ja takaisin vievät enemmän aikaa kuin  
pelkän tekstin kirjoittaminen. On otettava huomioon myös, että tekstihaun syntaksin  
opettelu vie alussa jonkin verran aikaa. Jos käyttäjä käyttää palvelua usein ja tekstihaun  
syntaksi tulee tutuksi, säästyy paljon aikaa.

Twenten yliopistossa on tehty tutkimus monimutkaisen web-lomakehaun ja teksti-  
haun käytöstä (Tjin-Kam-Jet et al. 2011). Tutkijat esittävät työssään, että graafisen lo-  
makkeen hakuehdot voidaan antaa myös kirjoittamalla tekstinä yhteen syöttökenttään.  
Tutkijat loivat web-palvelun matkan suunnitteluun mukailleen Hollannin rautateiden  
verkkopalvelua. Heidän lähtökohtansa oli samanlainen kuin tässä tutkielmassa: moni-  
mutkainen web- hakulomake vie paljon aikaa tulkita ja tietojen täyttäminen on hidasta.  
Tutkijat halusivat selvittää seuraavat asiat:

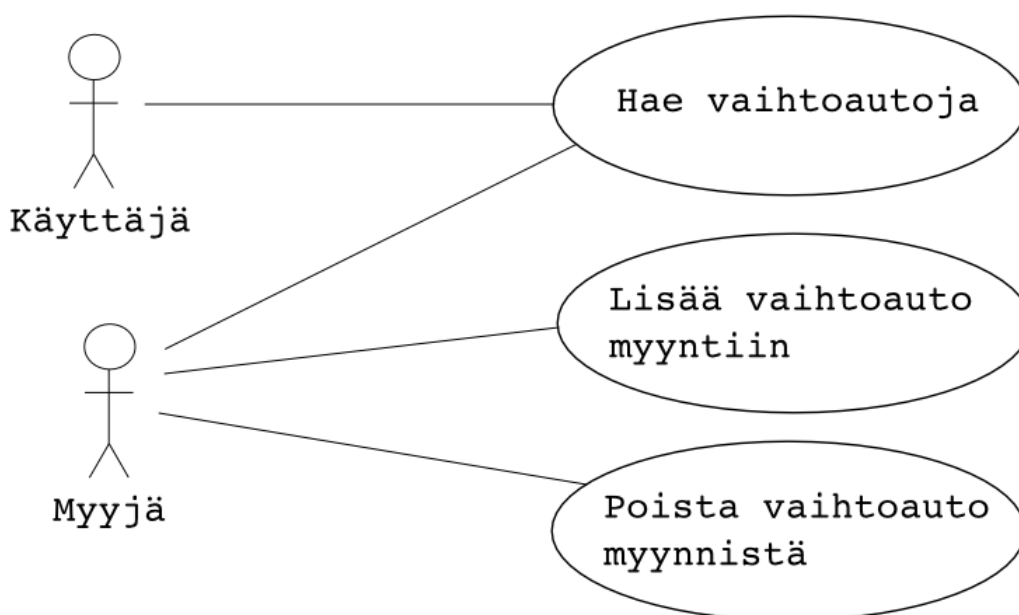
- kumpaa hakua ihmiset haluavat käyttää
- onko tekstihaku nopeampi käyttää
- onko hakusanoissa paljon vaihtelua
- ovatko ihmiset johdonmukaisia hakuehtoja antaessaan
- miksi tekstihaku on parempi tai huonompi.

Heidän käyttäjätestiinsä osallistui 17 henkilöä. Keskimäärin tekstihakua pidettiin nope-  
ampana, mukavampana, nopeampana ja parempana hakutapana. Tätä tulosta tukivat  
myös aikamittaukset. Monimutkaista web-hakulomaketta käyttäen aikaa hakuihin kului  
keskimäärin 7,3 minuuttia kun tekstihakua käyttäen kului vain 6,7 minuuttia. Testikäyt-  
täjät olivat siis noin 9% nopeampia tekstihakua käyttäen. Yleensä ottaen testikäyttäjät

olivat hyvin johdonmukaisia hakuetoja antaessaan, vaikkakin testikäyttäjien välillä oli suuria eroja.

## 2.10 Vaihtoautohaun tietojärjestelmä

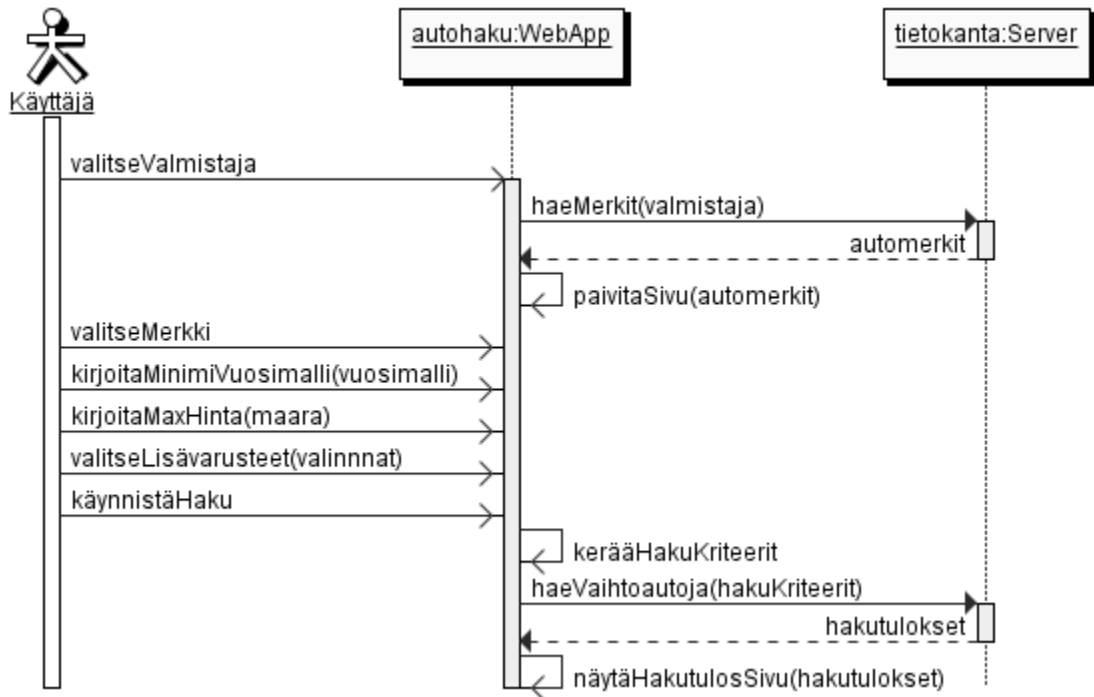
Vaihtoautohaun tekstihakulomakkeen toteutusta lähdettiin miettimällä mitä käyttäjiä ja toimintoja järjestelmässä olisi. Käyttötapauskaavio kuvaa järjestelmän käyttäjät ja toiminnot. Kuvassa 20 esitetään vaihtoautopalvelun käyttötapauskaavio.



Kuva 20: Vaihtoautopalvelun käyttötapauskaavio

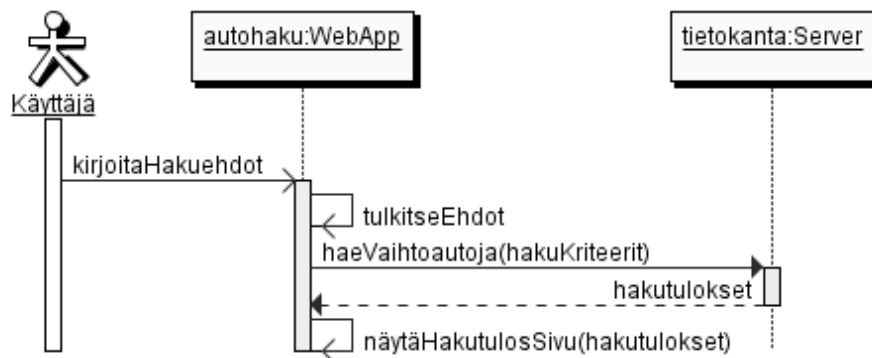
Kuvan 20 käyttötapauskaaviosta nähdään, ketkä käyttävät vaihtojen hakupalvelua ja millä tavalla sitä käytetään. Käyttäjä etsii vaihtoautoja tekemällä hakuja vaihtoautopalvelussa. Myyjä lisää ja poistaa vaihtoautoja palvelusta. Sen lisäksi myyjä voi seurata yleistä hintatasoa tekemällä vaihtoautohakuja.

Sekvenssikaavioilla kuvataan järjestelmän komponenttien välisiä vuorovaikutuksia ajan kuluessa. Kuvassa 21 on sekvenssikaavio vaihtoautohaun tekemisestä graafisella hakulomakkeella. Kaavio kuvaa käyttäjän, vaihtoautosovelluksen ja tietokannan välisiä viestejä ja toimintaa.



**Kuva 21: Sekvenssikaavio vaihtoautohaun tekemisestä monimutkaisella web-hakulomakkeella**

Vertailemalla kaaviota kuvan 21 sekvenssikaaviioon, joka kuvaa vastaavasti vaihtoautohaun tekemistä tekstihakulomakkeella, huomataan heti, että käyttäjän tekemiä toimintoja on huomattavasti vähemmän tekstihakulomaketta käytettäessä. Kuvan 21 kaaviossa on käytetty arviolta keskimääräinen määrä hakuehtoja.



**Kuva 22: Sekvenssikaavio tekstihakulomakkeen toiminnasta**

## 3 Hakulomakkeiden toteutus

Tässä luvussa käydään läpi hakulomakkeiden tekninen toteutus palvelimen asennuksesta ja konfiguroinnista alkaen.

### 3.1 *Palvelin*

Palvelin on laitteisto, joka koostuu suorittimesta, keskusmuistista, massamuistista ja verkkoyhteydestä. Palvelimen käyttöjärjestelmä suorittaa WWW-sovelluspalvelinohjelmaa, joka puolestaan mahdollistaa hakulomakkeiden saatavuuden. Web-palvelu koostuu useasta sovelluksesta, joita suoritetaan palvelimella. Työn tekemiseksi asennettiin palvelin ja siihen käyttöjärjestelmä.

Käyttöjärjestelmä on tietokoneohjelma, joka huolehtii tietokoneen resurssien jakamisesta muiden ohjelmien käyttöön. Käyttöjärjestelmä siis mahdollistaa useamman ohjelman suorittamisen tietokoneessa samanaikaisesti. Palvelinkäyttöjärjestelmä eroaa normaalista käyttöjärjestelmästä siten, että asennuksen mukana ei tule tiettyjä sovelluksia, esim. pelejä tai graafista käyttöliittymää, resurssien ja levytilan käytön optimoimiseksi. Hallinnointiohjelmat voivat sisältää enemmän säätömahdollisuuksia kuin normaali käyttöjärjestelmä.

Microsoft on yksi palvelinkäyttöjärjestelmien valmistajista. Microsoftin viimeisin palvelinohjelmisto, Windows Server 2012 R2, julkaistiin lokakuussa 2013. Sen pääpiirteisiin kuuluu IIS8.0 WWW-palvelinohjelmisto. Käyttöjärjestelmän käyttöönottoa varten tarvitsee ostaa lisenssi.

Linux on avoimen lähdekoodin käyttöjärjestelmä, mikä tarkoittaa, että sitä kehitetään vapaaehtoisvoimin. Linuxista on olemassa yli 700 eri jakelupakettia (Distrowatch, 2014), joista suosituimmat palvelinkäytössä ovat Ubuntu, Debian, CentOS ja Red Hat Enterprise Linux.

OS X Server on Apple-yhtiön tarjoama palvelinkäyttöjärjestelmä. Se pohjautuu Unix-käyttöjärjestelmään. Se ei ole saavuttanut suosiota, koska se on maksullinen ja se toimii vain Apple-yhtiön omalla laitteistolla.

Käyttöjärjestelmäksi valittiin Ubuntu Server Linux 12.04 LTS, joka on suunniteltu erityisesti palvelinohjelmistojen ajamista varten. LTS nimen perässä tarkoittaa pitkää tukea, Long Term Support. LTS-julkaisuihin on saatavilla tietoturvapäivityksiä ja sovelluspäivityksiä vähintään viisi vuotta. Ubuntu Linux on käyttöjärjestelmänä kahdeksassa prosentissa 10 miljoonaa suosituinta web-palvelua. Valintakriteereinä olivat palvelinohjelmiston ilmaisuus, vaatimattomat laitteistovaatimukset, runsas sovellusten saatavuus

ja helppo asennettavuus. Ubuntu Server Linux 12.04 LTS laitteistovaatimukset (Ubuntu dokumentation, 2012) ovat

- 300 MHz x86-arkkitehtuurin prosessori
- 128 Mt keskusmuistia
- vähintään 500 Mt kiintolevytilaa.

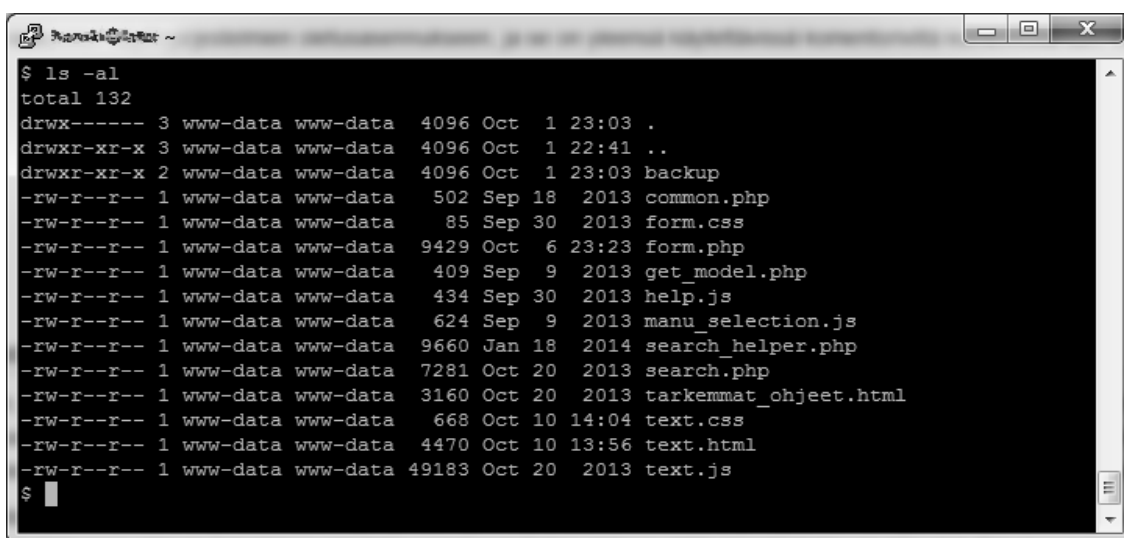
Ubuntu Linux on ilmainen käyttää. Laajennettua tukea on mahdollista saada maksua vastaan.

Käyttöjärjestelmää pyörittämään tarvittiin tietokoneen keskusyksikkö. Keskusyksikön yksityiskohtaiset tiedot ovat:

- Intel Atom 330 –prosessori, x86-yhteensopiva (2 ydintä, 4 säiettä),
- 4 Gt DDR3-RAM keskusmuisti,
- 4 Gt USB muistitikku.

Koska toteutettava verkkopalvelu ei edellytä vaatimuksia suorituskyvyn suhteen, käytetään sitä laitteistoa, joka on helposti saatavilla. Ubuntu Server Linux käyttöjärjestelmä asennettiin 4 Gt USB-muistitikulle käyttäen toista muistitikkuja, jossa olivat asennustiedostot. Tavallisesti käyttöjärjestelmä asennetaan kiintolevylle, mutta tässä tapauksessa käyttöjärjestelmä ja sovellukset mahtuivat hyvin muistitikulle ja sen suorituskyky oli aivan riittävä. Vapaata tilaa jäi 900 Mt. Ubuntu omalla apt-get –sovelluksella on helppo asentaa ja poistaa sovelluksia. Web-palvelua varten asennettiin Nginx, PostgreSQL-tietokanta, Ruby Version Manager, Ruby-ohjelmointikieli, Rails-kehys ja PHP5-skriptikieli.

Palvelinta hallinnoidaan tekstikäyttöliittymällä SSH-yhteyden kautta. SSH (Secure Shell) salaa tietoliikenteen palvelimen ja yhteyden ottajan välillä. Se on oleellista, koska salaamattomasta yhteydestä voisi ulkopuolinen saada selville käyttäjätunnuksen ja salasanan palvelimen liikennettä kuuntelemalla. SSH-salauksen toteuttavaan sovellukseen syötetty teksti toistetaan palvelimella. Linux-palvelimen hallinnointiin ei siis tarvita välttämättä graafista käyttöliittymää. Kuvassa 23 nähdään PuTTY-ohjelmalla muodostettu SSH-yhteys palvelimelle.



```

$ ls -al
total 132
drwx----- 3 www-data www-data 4096 Oct  1 23:03 .
drwxr-xr-x  3 www-data www-data 4096 Oct  1 22:41 ..
drwxr-xr-x  2 www-data www-data 4096 Oct  1 23:03 backup
-rw-r--r--  1 www-data www-data  502 Sep 18  2013 common.php
-rw-r--r--  1 www-data www-data   85 Sep 30  2013 form.css
-rw-r--r--  1 www-data www-data 9429 Oct  6 23:23 form.php
-rw-r--r--  1 www-data www-data  409 Sep  9  2013 get_model.php
-rw-r--r--  1 www-data www-data  434 Sep 30  2013 help.js
-rw-r--r--  1 www-data www-data  624 Sep  9  2013 manu_selection.js
-rw-r--r--  1 www-data www-data 9660 Jan 18  2014 search_helper.php
-rw-r--r--  1 www-data www-data 7281 Oct 20  2013 search.php
-rw-r--r--  1 www-data www-data 3160 Oct 20  2013 tarkemmat_ohjeet.html
-rw-r--r--  1 www-data www-data  668 Oct 10 14:04 text.css
-rw-r--r--  1 www-data www-data 4470 Oct 10 13:56 text.html
-rw-r--r--  1 www-data www-data 49183 Oct 20  2013 text.js
$

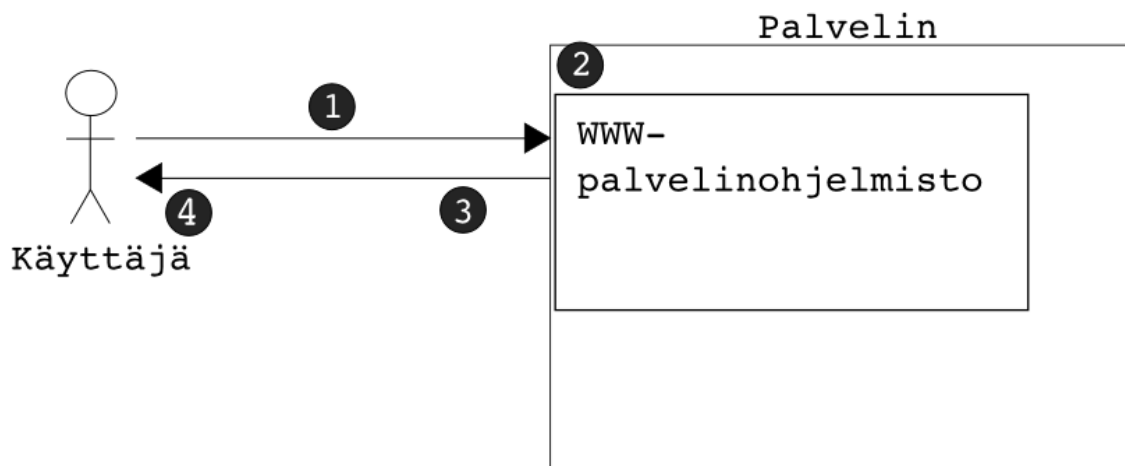
```

Kuva 23: SSH-yhteys palvelimelle PuTTY-ohjelmalla.

## 3.2 WWW-palvelinohjelma

HTML-dokumentit voi katsoa paikallisesti omalla tietokoneella, mutta paljon käteväm-  
pää on selata HTML-dokumentteja Internet-verkossa. Selaamista varten tarvitsee palve-  
limen suorittaa WWW-palvelinohjelmistoa, mikä kuuntelee jatkuvasti HTTP-pyyntöjä  
tietyissä TCP-portissa ja palauttaa pyydetyn dokumentin. Kuvassa 24 on esitetty palve-  
limen toimintaa kun käyttäjä lataa palvelun sivun, jossa on vain staattista sisältöä (taval-  
lisesti kuva tai HTML-dokumentti)

1. Käyttäjä syöttää selaimen osoitteen ja selain käynnistää hakupyynnön osoitteen mukaiselta palvelimelta
2. WWW-palvelinohjelmisto vastaanottaa ja käsittelee selaimen pyynnön
3. WWW-palvelinohjelmisto palauttaa selaimen pyytämän HTML-dokumentin
4. Selain näyttää HTML-dokumentin



Kuva 24: WWW-palvelinohjelmiston toiminta

### 3.2.1 Nginx

Nginx on yksi käytetyimmistä WWW-palvelinohjelmistoista (Netcraft, 2013). Se julkaistiin vuonna 2002 ja se on saatavilla Linux, Windows, Solaris, Mac OS X ja UNIX-käyttöjärjestelmille. Alun perin Nginx kehitettiin hyvin suurten liikennemäärien verkkopalvelujen www-palvelinohjelmäksi. (About nginx, 2013) Nginx valittiin tutkielmas-  
sa käytettäväksi WWW-palvelinohjelmäksi sen hyvän suorituskyvyn ja pienen muistinkulutuksen vuoksi. Yleisin WWW-palvelinohjelmisto, Apache HTTP Server, olisi ollut myös hyvä valinta sen toimintavarmuuden ja Internetissä olevien lukuisten asennus- ja

konfiguraatio-ohjeiden vuoksi. Tässä tapauksessa haluttiin kuitenkin tutustua uuteen ohjelmistoon.

### 3.2.2 Apache HTTP Server

Apache HTTP Server, lyhyemmin Apache, on maailman käytetyin WWW-palvelinohjelmisto. Sen on kehittänyt Apache Software Foundation ja se perustuu avoimeen lähdekoodiin. Alun perin NCSA:n httpd-palvelinohjelman päälle kehitetty ohjelma kirjoitettiin kokonaan uudelleen ja siitä tuli versio Apache versio 2.0, joka on pohjana nykyisille versioille 2.2 ja 2.4. Apachen ohjelmisto on toimintavarma, koska se on niin laajalti käytössä ja ohjelmaa on kehitetty vuodesta 1995 lähtien (Apache HTTP Server 2014). Apachen asennukseen ja käyttöön löytyy paljon oppaita Internetistä. Ohjelmiston perustoiminto on staattisten dokumenttien jakelu. Toimintaa voidaan täydentää asentamalla moduuleja. Kuten tässäkin tutkielmassa on laajennettu Nginx-ohjelmiston toimintaa asentamalla PHP-käsittelijä, voidaan myös Apacheen asentaa moduuli, joka lisää PHP-skriptien käsittelyn.

### 3.3 PHP ja muita skriptikieliä

PHP-skriptikieli julkaistiin vuonna 1995. Se on suunniteltu eritoten WWW-palvelujen toimintojen toteuttamiseen palvelimella, mutta sitä voi käyttää myös yleisenä ohjelmointikielenä. PHP:n voi asentaa joko WWW-palvelinohjelmiston moduulina tai erillisenä PHP-käsittelijäohjelmalla, jolloin se on tausta-ajossa ja odottaa käsittelypyyntöjä määritellyssä TCP-portissa. (Achour, M. et al. 2014.).

PHP-skriptikieli valittiin toteuttamaan tämän tutkielman palvelun hakutoiminnon ja tulosten näyttämiseen. Sen etuja ovat sovelluskehityksen helppous, kielen selkeys ja ohjeiden saatavuus. Haittana on se, että kun ohjelmiston koko kasvaa, ylläpito vaikeutuu tietokanta-kyselyiden määrän ja niiden tiedon riippuvaisuuksien takia. Tätä ei kuitenkaan nähty ongelmana, koska ohjelmisto tulisi olemaan suppea ja ohjelmiston ylläpitoa ei tultaisi tekemään.

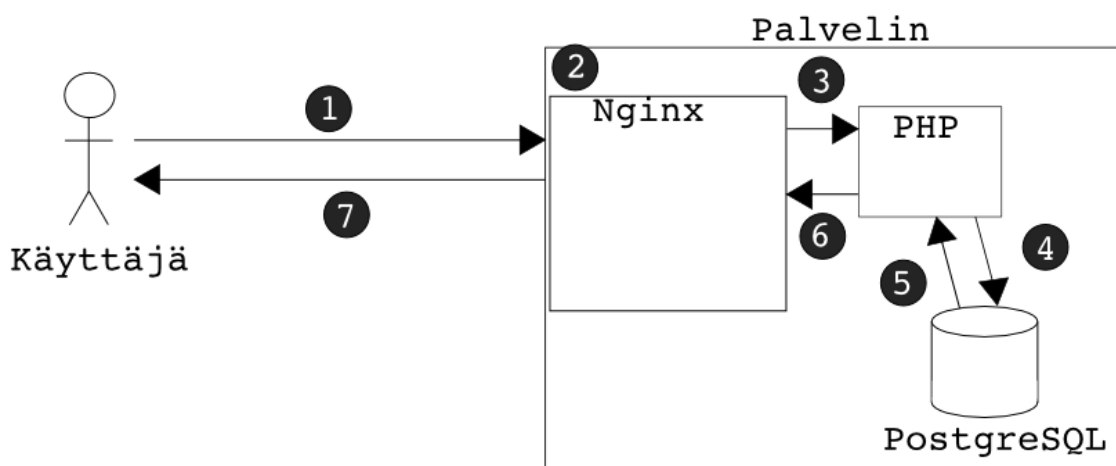
Muita yleisiä WWW-palvelinpään skripti- ja ohjelmointikieliä WWW-palvelun toteutukseen ovat ASP, JSP/Java Servlet, Perl, Ruby on Rails ja Python. ASP, Active Server Pages, on Microsoftin kehittämä Visual Basicia muistuttava skriptikieli. Tutkielman verkkopalvelun toteuttaminen olisi varmasti onnistunut myös tällä kielellä, mutta tässä haluttiin pysytellä vapaan lähdekoodin kehyksien parissa. Java-ohjelmointikielen pohjautuvat JSP ja Servlet –tekniikat ovat helppoja käyttää, jos Java-kielen syntaksi on tuttu. Niiden käyttöönotto vaatii kuitenkin enemmän valmisteluja kuin PHP:n. Perl on ohjelmointikieli, jota on käytetty paljon web-palvelujen toteuttamisessa, mutta on hieman hankala ja vanhanaikainen. Ruby on Rails on MVC-kehys web-palvelun toteutukseen. Se on kuitenkin liian massiivinen tämän työn tekemiseen. Python ja sen web-kehys, Django, olisivat voineet olla hyvä vaihtoehto myös, mutta PHP valit-



tiin sen keveyden ja yksinkertaisuuden vuoksi. Kuvassa 25 on esitetty käyttäjän aktiivisuuden palvelupyynnön kulkeminen järjestelmän läpi.

1. Käyttäjä aktivoi vaihtoautojen haun. Se luo palvelupyynnön selaimesta palvelimelle annetuilla hakuehdoilla.
2. Nginx ottaa pyynnön vastaan ja toteaa, että se käyttää PHP:tä.
3. Nginx lähettää koodin PHP-moduulin käsiteltäväksi
4. PHP-moduuli suorittaa kyselyn PostgreSQL-tietokantaan, jos ohjelma sisältää SQL-kyselyn (muuten suoritus siirtyy kohtaan 6)
5. Tietokanta palauttaa kyselyn tuloksen PHP-moduulille
6. PHP-moduuli palauttaa käsitellyn vastauksen Nginx:lle

Nginx muodostaa HTML-dokumentin PHP:n käsittelemästä tiedosta ja palauttaa sen käyttäjän nähtäväksi



Kuva 25: PHP-skriptin toiminta palvelussa

Kuvan 25 esittävässä palvelussa käytetty HTML-dokumentti sisältää sen toiminnallisuuden toteuttavaa PHP-koodia. Seuraavassa on esimerkki dokumenttiin sisältyvän PHP-skriptin käytöstä.

```

<!DOCTYPE html>
<html><head><title>PHP esimerkki</title></head><body>
<?php
  echo 'Hei maailma';
?>
</body></html>
  
```

PHP-skripti lisätään HTML-dokumentin sisään `<?php` ja `?>` -merkkijonojen väliin. Aloituserkkijonona voi käyttää myös `<?php` sijasta pelkkää `<?-`merkkijonoa. Kuvassa 26 on PHP-skripti, jolla suoritetaan tietokantakysely.

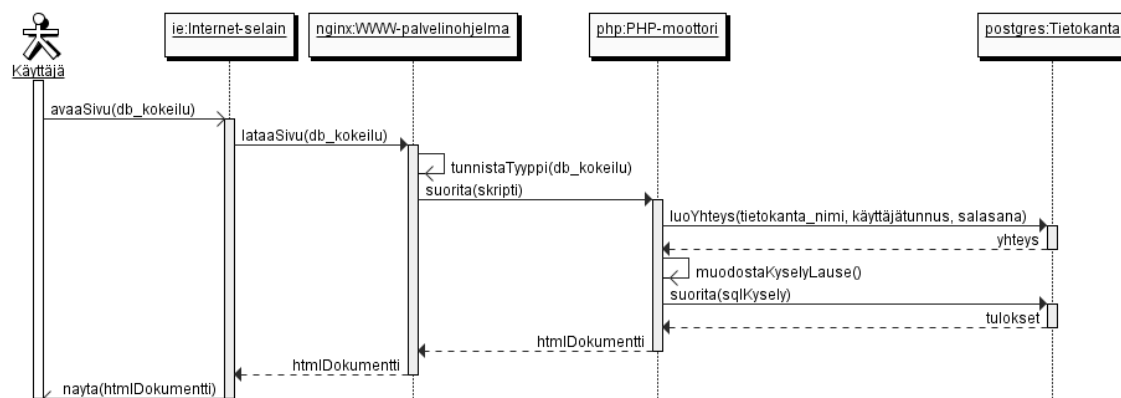
```

<html><body>
<h2>Tietokantakokeilu</h2>
<table>
<?
$dbconn = pg_connect('host=localhost dbname=auto_db user=auto pass-word=pwd connect_timeout=5') or die
('Ei yhteyttä tietokantaan: ' . pg_last_error());
$query = "SELECT id, manufacturer, model, gears_type, fuel from Cars where manufacturer='Volvo' and
model='V70' ";
$result = pg_query($query) or die('Query failed: ' . pg_last_error());
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC)) {
    echo "\t<tx><td>\n";
    echo $line['id'];
    echo "\t</td><td>\n";
    echo $line['manufacturer'];
    echo "\t</td><td>\n";
    echo $line['model'];
    echo "\t</td><td>\n";
    echo $line['gears_type'];
    echo "\t</td><td>\n";
    echo $line['fuel'];
    echo "\t</td></tx>\n";
}
pg_free_result($result);
pg_close($dbconn);
?></table>
</body>
</html>

```

Kuva 26: Tietokantakyselyn suorittaminen PHP-skriptillä

Kuvan 26 PHP-skripti on esimerkki tietokantakyselystä. Skripti muodostaa yhteyden tietokantaan antamalla tietokannan nimen, käyttäjätunnuksen ja salasanan. Kun yhteys on avattu, annetaan SQL-lauseke tietokannan suoritettavaksi. Tietokannan palauttavat tiedot sisällytetään HTML-taulukkoon. Lopuksi yhteys suljetaan. PHP-skripti on osana HTML-dokumenttia. Kuvassa 27 nähdään sekvenssikaavio tietokantakyselystä.



Kuva 27: Tietokantakysely sekvenssikaaviolla mallinnettuna

Kun skripti suoritetaan palvelimella, selain saa seuraavanlaisen HTML-dokumentin palvelimelta:

```

<html><body>
    <h2>Tietokantakokeilu</h2>
    <table>

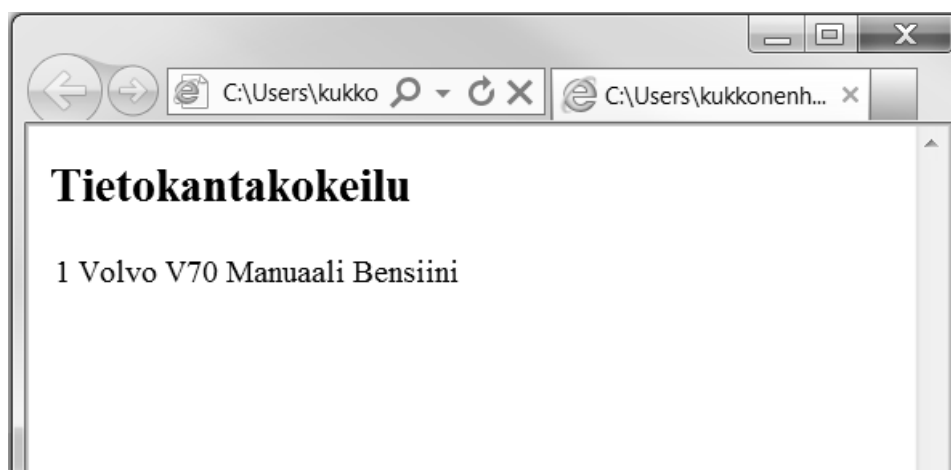
```

```

        <tr><td>
            1
        </td><td>
            Volvo
        </td><td>
            V70
        </td><td>
            Manuaali
        </td><td>
            Bensiini
        </td></tr>
    </table>
</body>
</html>

```

Kuvassa 27 nähdään, miltä äskeinen HTML-dokumentti näyttää selaimessa.



Kuva 28: PHP-skriptin tulos IE 9 -selaimessa

### 3.4 Tietokanta

Web-palveluissa käytetään yleensä tietokantaa tietojen varastointipaikkana, koska tietoja pitää pystyä lisäämään, lukemaan, päivittämään ja poistamaan monen käyttäjän toimesta samanaikaisesti. Tiedot ovat yhdessä paikassa ja ne ovat saatavilla monelle käyttäjälle web-palvelun kautta. Vaihtautotietoja lisääviä tahoja voi olla useampia ja vaihtautotietojen tulee olla ajan tasalla ja paikkansa pitäviä. Tietokantaohjelma huolehtii tiedon eheydestä ja ajantasaisuudesta. Tietokanta on kokoelma yhteneviä tietoja, joko fyysisessä (paperilla) tai sähköisessä muodossa (tietokoneohjelma). Keskitymme tässä tutkielmassa sähköisen muodon tietokantoihin.

Yleisimmät tietokannat, joita käytetään web-palvelun yhteydessä ovat PostgreSQL, MySQL, Oracle, MSSQL Server, SQLite, MongoDB, CouchDB ja Redis. Tutkielman

työosuuden tietokantaa valittaessa tietokantavaihtoehtoista karsittiin pois kaikki maksulliset, eli Oracle ja MSSQL Server, koska ilmaisiakin tietokantoja on saatavilla ja ne ovat varmasti riittäviä toteuttamaan tietovarasto. MongoDB, CouchDB ja Redis ovat NoSQL-tietokantoja, mikä tarkoittaa, että ne käyttävät SQL:stä poikkeavaa kyselykieltä tietojen tallentamiseen, kyselyyn ja poistamiseen. SQL-kyselykieltä pidettiin helpompana käyttää PHP-skriptien kanssa, joten mahdolliseksi tutkielman tietokantaohjelmiksi muodostuivat MySQL, SQLite ja PostgreSQL.

MySQL on julkaistu GPL-lisenssillä vuonna 1995. Se on maailman käytetyimpiä tietokantoja. MySQL on suosittu tietokanta web-sovelluksissa ja se on osana tunnettua web-kehityspakettia, LAMP, joka käsittää Linux-käyttöjärjestelmän, Apache HTTP - WWW-palvelinohjelmiston, MySQL-tietokannan ja PHP-sovelluskehityksen valmiiksi asennettuna.

SQLite on relaatiotietokanta, jota voidaan ajaa tietokoneen keskusmuistissa tai yhdessä tiedostossa (About SQLite, 2014). Se on hyvin pieni (225 kt) ja sitä käytetään erityisesti testitietokantana Ruby on Rails -sovelluksissa testien suorittamiseen, koska tietokantaoperaatiot ovat nopeita kun tietokantaa ajetaan keskusmuistissa. Lisenssinä on Public domain -ohjelmisto, joka mahdollistaa muokkauksen, vapaan levityksen ja ohjelmistoihin linkityksen.

PostgreSQL on avoimen lähdekoodiin perustuva tietokanta. Sen kehitys on aloitettu vuonna 1986. Sen etuihin kuuluu vakaus, luotettava toiminta, käyttötuki, saatavuus moneen eri käyttöjärjestelmälle ja graafiset työkalut tietokannan suunnitteluun ja ylläpitoon. (About PostgreSQL, 2013)

Ylläolevissa kolmessa tietokannassa käytetään SQL-kyselykieltä tietojen luomiseen, hakuun, päivittämiseen ja poistamiseen. Koska tutkielmassa tehdyssä työssä tiedon määrä on vähäinen eikä pieniä päivityksiä tarvitse tehdä paljon, osuu valinta relaatiotietokantaan. Tietokantavaihtoehtoista valittiin PostgreSQL, koska se ilmainen käyttää, helppo asentaa, helppo konfiguroida ja sille on hyvä ajurituki sekä PHP:ssä että Ruby on Rails-sovelluskehityksessä. Hyviä valintoja tietokantaohjelmaksi olisi ollut myös MySQL ja SQLite, koska ne ovat tukevat SQL-kyselykieltä ja niille on olemassa ajurituki Ruby on Railsissä ja PHP-skriptikielessä.

PostgreSQL:n asennus onnistui käyttämällä Ubuntu Linuxin asennusohjelmaa, apt-get install:ia. Se asentaa ohjelmiston tarvittavat paketit sekä riippuvaisuuspaketit, konfiguroi tietokantaohjelman käyttöjärjestelmän palveluksi (daemon) ja asettaa tietokannan käynnistymään palvelimen käynnistymisen yhteydessä. Tietokantapalvelu kuuntelee oletuksena TCP-porttia 5432. PostgreSQL-tietokannan hallinnointiin voi käyttää mukana tulevaa psql-komentorivisovellusta, erikseen asennettavaa pgadminIII GUI-työkalua tai Ruby on Railsin mukana tulevaa console -sovellusta.

### **3.5 Ruby on Rails**

Ruby on Rails, lyhyemmin RoR, on Ruby-ohjelmointikielen päälle tehty ohjelmistokehys WWW-pohjaisten sovellusten luomiseen. Ruby on Rails -web-kehys sisältää työka-

lut ja laajennusmoduulit dynaamisten web-sovellusten tekemiseen (Kasurinen, 2011). Ruby on Railsin etu on nopea sovelluskehitys. Koska RoR-sovelluksen kansiodien nimet ovat ennalta sovittuja, sovelluskehittäjä tietää tarkalleen, mistä kansioista tiedostot löytyvät.

RoR pohjautuu MVC-arkkitehtuuriin. Sen ideana on jakaa sovellus kolmeen osaan: Modeliin eli malliin, Viewiin eli näkymään ja Controlleriin eli käsittelijään. Tällä tavalla eriytetään sovelluslogiikka, tieto ja tiedon näyttö, jolloin sovellus on helpompi ylläpitää.

Tässä tutkielmassa Ruby on Rails -kehystä käytettiin vaihtoautotietojen luomiseen tietokantaan. Osana sovelluksen toteutusta, Ruby on Rails -sovelluksessa tietokannan luonti ja pohjustaminen ovat helppoja tehtäviä. Tietokanta alustettiin n. 5000:llä vaihtoautotiedolla. Ruby on Rails -sovellus arpoi autojen valmistajan, merkin, ja muut tiedot ja tallensi ne tietokantaan.

Rails-sovelluksen hakemistorakenne on sovittu tietyntylaiseksi, jotta ohjelmistokehittäjät osaisivat sijoittaa tiedostot oikeaan paikkaan ja ylläpito olisi helppoa. config-hakemistossa ovat tiedostot, joilla alustetaan sovelluksen yhteydet, mm. tietokantamääritykset eri ympäristöjä varten: kehitys, testi ja tuotanto. Ympäristöt voi määrittää itse, mutta kehitys on yleensä ohjelmistokehittäjän oma tietokone, testi on testien ajamista varten ja tuotanto tarkoittaa ympäristöä, joka on varsinainen sovelluksen suorittamispaikka. App-hakemistossa ovat sovellustiedostot: mallit, kontrollerit ja näkymät.

Tietokannan alustus tapahtuu luomalla projekti käyttäen *-d postgresql* -vipua, jolloin projektitiedostot alustetaan tiedolla, joka kertoo PostgreSQL-tietokannan olevan käytössä. PostgreSQL-kantaan luotiin käyttäjätunnus ja salasana, jotta RoR-projektin tiedoistoilla on yhteys tietokantaan. PostgreSQL-tiedot määritellään *config/database.yml*-tiedostoon. Tietokannan alustuksen suorittava koodi kirjoitetaan *db/seeds.rb*-tiedostoon.

### 3.6 JavaScript

JavaScript on skriptikieli, jota käytetään Internet-selaimissa suoritettavien ohjelmien toteuttamiseen (Crockford 2008). Tässä työssä JavaScriptiä käytettiin tekstihaun toteutuksessa, koska JavaScript on hallitseva kieli dynaamisten Web-sovellusten toteuttamisessa.

JavaScript luotiin vuonna 1995 vain parin viikon kehittämisen jälkeen Netscape-selaimelle. Jotta muutkin selaimet voisivat hyödyntää skriptikieltä, vietiin JavaScript standardoitavaksi ECMA-standardiorganisaatioon. Standardista tuli ECMAScript 262 Ed. 1. JavaScript on yksi standardin toteutuksista.

HTML DOM, Document Object Model, on W3C:n standardoima rajapinta HTML-dokumentin muokkaukseen. JavaScriptillä voi muokata DOM-rajapinnan kautta HTML-dokumentin sisältöä. Esimerkiksi voidaan muuttaa tekstiä JavaScriptillä. Tiedetään, että elementin id on ”testi-elementti”, joten kirjoitetaan

```
document.getElementById("testi-elementti").innerHTML = "Hello JavaScript";
```

Edellinen skripti hakee elementtiä id-tunnisteella ”testi-elementti” ja sen löydettyään kirjoittaa sen sisällöksi ”Hello JavaScript”.

JavaScript sisältää luokan XMLHttpRequest, jolla voidaan suorittaa pyyntöjä palvelimelle taustalla. Tätä kutsutaan AJAX-tekniikaksi.

JavaScript-kielen syntaksi muistuttaa C-kieltä ja Javaa. JavaScriptin opettelu on helppoa, jos osaa edellä mainittuja kieliä jo ennestään. JavaScriptissä on tyypittömät muuttujat, mikä tarkoittaa, että kaikki muuttujat määritellään käyttämällä *var*-sanaa. Skriptin voi sisällyttää HTML-dokumenttiin `<script>`-elementillä joko sen sisään kirjoittamalla tai määrittelemällä skriptitiedoston nimen `src`-attribuutilla.

### 3.7 Monimutkainen web-hakulomake

Toteutus monimutkaiseen hakulomakkeeseen tehtiin HTML-kuvauskielellä ja JavaScript-skriptikielellä. Sivua avataan web-selaimella. Lomakkeella on monta graafista lomake-elementtiä, joihin syötetään hakuarvot. Lomake alkaa auton merkin valinnalla. Kun käyttäjä on valinnut sen, selain lähettää pyynnön valitun auton malleista taustalla. Kun vastaus saadaan palvelimelta, vastauksesta erotellaan automallien nimet ja ne syötetään valittaviksi arvoiksi ”Auton malli”-valintalaatikkoon. Kuvassa 28 on havainnollistettu, miten edellä mainittu tapahtuu.

#### Vaihtoautohaku

Valmistaja

- Seat
- Skoda
- Subaru
- Toyota
- Volkswagen
- Volvo

Malli



#### Vaihtoautohaku

Valmistaja

- Seat
- Skoda
- Subaru
- Toyota
- Volkswagen
- Volvo

Malli

- Beetle
- Bora
- Golf
- Jetta
- Kupla
- Passat

Kuva 29: Auton valmistajan valinta täyttää Malli-elementin

#### 3.7.1 HTML

Kuvauskielellä toteutettiin kaikki staattinen sisältö hakupalveluun. Pelkästään staattista sisältöä ei tullut kuin kaksi sivua: testikäyttäjän tarkempi ohjesivu ja tekstihaun lomake. Muut sivut sisälsivät PHP-skriptikieltä.

Lomakkeen elementtien sijoittelu tehtiin HTML:n taulukkoelementillä `<table>`. Elementtien kuvaus sijoitettiin elementin vasemmalle puolelle ja elementit sijoitettiin allekkain. Niiden arvojen kohdalla, joissa pystyi syöttämään kaksi arvoa, minimin ja maksimin, sijoitettiin niiden kentät vierekkäin, kuten kuvassa 29 näkyy.

Vuosimalli  -

**Kuva 30: Elementtien sijoittelu kahden syöttökentän kohdalla**

Lisävaruste-elementin valinnat sijoitettiin kahteen jonoon ja allekkain, jotta se olisi selkeän näköinen.

### 3.7.2 AJAX

Asynchronous JavaScript and XML, lyhyemmin AJAX, tarkoittaa asynkronisten pyyntöjen suorittamista JavaScriptillä tausta-ajossa. Yleisimmin sitä käytetään hakemaan tietoa tietokannasta ja päivittämään tieto web-sivulle ilman, että koko web-sivua tarvitsee hakea palvelimelta. Tässä tutkielmassa AJAX-tekniikkaa käytetään auton merkkien hakemiseen valitun auton valmistajan perusteella. Palvelin palauttaa auton valmistajan kaikki automerkit JavaScriptin tietojen siirtomuodossa eli JSON-muodossa. JavaScript-sovellus käsittelee vastauksen ja päivittää automerkit hakulomakkeelle ilman, että koko sivu päivittyisi.

### 3.7.3 CSS

Cascading Style Sheets, CSS, on tekniikka, jolla toteutettiin HTML-dokumenttien tyyli. HTML-dokumentti kuvaa vain dokumentin rakenteen, ei sen ulkoasua. CSS:llä voi määrittää tekstin ja elementtien ulkoasun, sekä niiden sijoittelun sivulla.

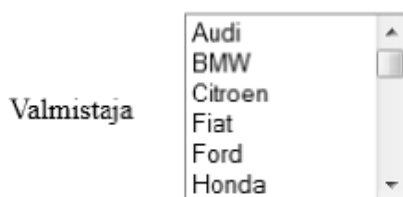
CSS on kehitetty W3C:n toimesta ja sen ensimmäinen versio on julkaistu 1996. Viimeisin virallinen versio on 2.1, joka on julkaistu suosituksena 2011. Versio 3.0 on kehitteillä ja sen tekeminen on aloitettu jo 1999.

HTML-dokumentin tyylit voidaan sisällyttää joko erillisessä tiedostossa tai kirjoittaa suoraan HTML-dokumenttiin. Ylläpidollisesti on paljon helpompaa ja järkevämpää käyttää erillistä tiedostoa.

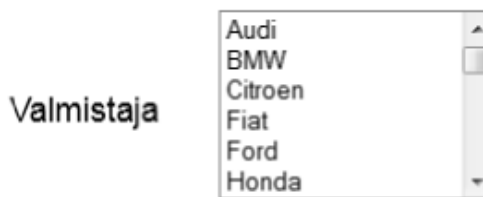
```
<link rel="stylesheet" href="tyyli.css">
```

Ylläolevalla HTML-koodilla saadaan liitettyä tyyli.css-tiedosto HTML-dokumenttiin, jolloin CSS-tiedoston tyylimäärittelyt vaikuttavat HTML-dokumentin elementteihin. Sekä tekstihakulomakkeelle että graafiselle lomakkeelle tehtiin omat CSS-tiedostonsa. Kuvassa 30 on vasemmalla puolella osa HTML-dokumenttia ilman CSS-tyylimäärittelyä.

## Vaihtoautohaku



## Vaihtoautohaku



Kuva 31: CSS-tyylimäärityillä voi muotoilla tekstiä

Tällöin selain päättää, mikä kirjasin on käytössä ja minkä kokoinen se on. Oikealla puolella on HTML-dokumentti, jossa on lisätty CSS-tyylitiedosto. Selaimen käyttämä kirjasin on vaihdettu ja kokoa suurennettu. CSS-tekniikkaa käytettiin, jotta saataisiin määriteltyä halutunlainen ulkoasu tekstille.

### 3.8 Tekstihakulomake

Tässä kohdassa käydään läpi tekstihakulomakkeen toteutus. Yksinkertaisin toteutus tekstihauulle olisi toteuttaa tekstin syöttölaatikko, joka välittäisi kirjoitetun tekstin suoraan tietokannalle. Tässä tapauksessa käyttäjä siis kirjoittaisi hakunsa SQL-kyselykielellä. Koska sen oppiminen vie paljon enemmän aikaa kuin graafisen hakulomakkeen käyttö, sitä ei tehdä. Sen sijaan toteutetaan oma syntaksi hakuehtojen syöttämiseen. Kuvassa 30 on toteutetun tekstihaun käyttöliittymä.



## Vaihtoautohaku

[Avaa ohje](#)



Kuva 32: Vaihtoautojen tekstihaku

Käyttöliittymä on hyvin yksinkertainen, jos sitä verrataan graafiseen hakulomakkeeseen. Käyttöliittymä sisältää otsikon, tekstin syöttölaatikon, ohje-linkin ja toimintopainikkeet. Tekstin syöttölaatikkoa on suurennettu ja pidennetty oletuspituudesta, jotta ainakin suurin osa hakuehdoista olisi selkeästi näkyvissä, jos hakuehtoja on paljon.



### 3.8.1 Toiminta

Hakua rajoittavasta syntaksista pyrittiin tekemään helposti omaksuttava ja nopeasti kirjoitettava. Haku palauttaa samat tulokset riippumatta hakuehtojen järjestyksestä. Kun hakuehdot on kirjoitettu, käyttäjä käynnistää haun painamalla Enter-näppäintä tai klikkaamalla ”Hae vaihtoautoja”-nappia. Se käynnistää JavaScript-sovelluksen, joka lukee käyttäjän kirjoittamat hakuehdot ja yrittää tunnistaa ne. Tunnistetut hakuehdot välitetään PHP-skriptille, joka suorittaa haun. Tunnistamattomat hakusanat näytetään hakutulossivun alussa punaisella tekstillä, jotta haun suorittaja huomaisi ne helposti.

### 3.8.2 JSON

Vaihtoautojen hakusanat ja niiden tietokantatunnisteet tallennettiin JSON-formaatissa JavaScript-tiedostoon, joka ladattiin kun tekstihakulomake avattiin. JavaScript Object Notation, eli JSON, on tiedon säilyttämiseen ja siirtoon käytettävä formaatti. Kun käyttäjä on kirjoittanut hakusanat ja painanut Hae-nappia, niin hakusanoja verrataan JavaScript-tiedoston sanoihin. Tunnistetuista sanoista saatiin tietokantatunnisteet ja niistä muodostettiin hakuehdot, joilla tietokantaan suoritettiin haku autotietojen saamiseksi ja näyttämiseksi. Alla olevassa listauksessa on autotietoja JSON-formaatissa.

```
{
  "Autot": [{
    "nimi": "Volvo",
    "maa": "Ruotsi"
  }, {
    "nimi": "Volkswagen",
    "maa": "Saksa"
  }]
}
```

Autot-avaimen alla on lista autotiedoista. Koska hakusanat on tallennettu muistissa olevaan muuttujaan, voidaan käyttäjän syöttämiä hakuehtoja verrata niihin nopeasti ja tunnistaa oikeat hakusanat. Haittana on se, että jos hakusanat muuttuvat, esimerkiksi tietokantaan lisätään uusi automalli, pitää myös hakusanat sisältävää tiedostoa muokata. Jos hakusanat olisi määritetty tietokantaan, ylläpito olisi helpompaa.

### 3.8.3 Säännöllinen lauseke

Koska vaihtoautojen hakusanoja on paljon, tarvittiin nopea tekniikka sanojen vertailemiseen. Säännöllisten lausekkeiden, Regular Expression l. regexp, avulla voidaan suorittaa tekstimerkkivertailuja hyvin nopeasti. Sitä käytetään etsimään, korvaamaan ja

irrottamaan informaatiota merkkijonoista. Säännöllisillä lausekkeilla on yleensä huomattava etu suoritusnopeudessa (Crockford, 2008).

Vaikka se on nopea tekniikka, sen syntaksin ymmärtäminen on hankalaa, jos vaikka vertaa JavaScriptin merkkijonojen vertailuun. Säännöllinen lauseke annetaan yleensä kenoviivojen sisällä, esim. `/sana/`.

Tässä on muita esimerkkejä säännöllisten lausekkeiden syntaksista:

`/auto/` palauttaa kaikki ”auto”-merkkijonon sisältävät sanat esim. "auto", "automaatti", "mauton"

`/^auto/` palauttaa kaikki "auto"-alkuiset sanat, esim. ”automaatti”, mutta ei ”mauton”

`/^auto$/` palauttaa kaikki sanat jotka alkavat "auto"-merkkijonolla ja päättyvät ”auto”-merkkijonoon (eli käytännössä palauttaa vain ”auto”-sanat)

`/auto$/` palauttaa sanat, jotka päättyvät merkkijonoon "auto", esim. ”asuntoauto”

Käyttäjä voisi syöttää hakuun esimerkiksi: ”V70 da 180tkm- 2006-”, joka tarkoittaisi, että käyttäjä haluaa saada tiedot kaikista Volvo V70-malleista, joiden polttoaine on diesel, vaihteisto on automaatti, mittarilukema vähintään 180 000 kilometriä ja vuosimallista 2006 lähtien uudempaan.

### 3.8.4 Sumea hakutekniikka

Edellisessä alakohdassa kerrotaan, kuinka hakusanoja suoraan vertaamalla saadaan hakutuloksia. Sumean haun tekniikassa on mahdollista saada hakutuloksia ilman, että hakusana täsmäämät täysin. Sumean haun etuna on se, että jos käyttäjälle sattuu kirjoitusvirhe, niin sumea haku osaa tunnistaa myös hakusanoja liki olevat sanat. Haittapuolena voidaan pitää pidempää haun kestoa. Tässä tutkielmassa ei kuitenkaan käytetä sumeaa hakutekniikkaa.

### 3.8.5 Ennakoiva ja passiivinen haku

On olemassa erilaisia hakutaktiikoita. Ennakoiva haku tarkoittaa sitä, että hakusanoja kirjoittaessa sovellus yrittää tunnistaa hakusanan, suorittaa haun taustalla ja päivittää jopa hakutuloksia. Tässä etuna on se, että käyttäjä näkee heti, onko hakusana kirjoitettu oikein ja löytyykö sillä hakutuloksia. Haittapuolena voidaan pitää sitä, että verkkopalvelun toteuttajan täytyy ottaa huomioon, miten palvelu suoriutuu moninkertaisesta hakusanojen kyselymäärästä. Passiivinen haku tarkoittaa sitä, että haku käynnistyy vasta kun käyttäjä itse käynnistää haun. Etuna on se, ettei palvelinta suorituskykyä rasiteta monilla kyselyillä yhden sijaan. Haittapuolena on se, ettei järjestelmä välttämättä tunnista väärin kirjoitettua hakusanaa ja käyttäjä joutuu palaamaan hakusivulle muokkaamaan hakua,

mikä hidastaa tulosten löytämistä. Tässä tutkielmassa käytetään passiivista hakutaktiikkaa.

## 4 Käyttäjätetit

Hakutapojen vertailemiseksi luotiin käyttäjätesti, jolla pyritään saamaan selville, onko vaihtoautojen haku tekstihaulla nopeampaa ja helpompaa kuin graafisella hakulomakkeella ja voiko monimutkaisen web-hakulomakkeen korvata tekstihaulla. Tässä luvussa käydään läpi testin valmistelut, kuvaus ja käyttäjäprofiilit.

### 4.1 Testin valmistelu

Vaihtoautohakupalvelu asennettiin ja konfiguroitiin toimimaan WWW-palvelimella, joka oli julkisessa Internet-verkossa. Suoritusohjeet kirjoitettiin testiin osallistujia varten. Pääsy palveluun oli suojattu käyttäjätunnuksella ja salasanalla, jottei ulkopuolisilla tahoilla olisi pääsyä. Tunnukset palveluun annettiin testin suorittajille suoritusohjeissa. Palvelun konfigurointi julkiseen Internetiin helpotti testin suorittamista siten, että testikäyttäjä pystyi suorittamaan testin omalla ajallaan ja tietokoneellaan. Edellytyksenä oli, että testikäyttäjän tietokoneella oli toimiva Internet-yhteys.

Testin tarkoituksena oli suorittaa ohjeen mukaisia hakuja vaihtoautopalvelussa ensin graafisella hakulomakkeella ja sitten samat haut uudelleen tekstihaulla. Tekstihakulomakkeen yhteydessä oli hyperlinkki, jota klikkaamalla aukesi pikaohje samalle sivulle. Pikaohje sisälsi linkin tarkempaan ohjeeseen, jossa selitettiin tekstihaun syntaksia useiden esimerkkien avulla. Tarkempi ohje avautui omalle sivulleen.

Testin ohje lähetettiin sähköpostilla ilmoittautuneille testihenkilöille. Testihenkilöitä pyydettiin suorittamaan testi omaan tahtiin ja kirjoittamaan vastaukset testin jälkeen kysytyihin kysymyksiin. Vastaukset kysymyksiin tuli lähettää sähköpostilla tutkielman tekijälle.

### 4.2 Testin kuvaus

Testiin ilmoittautuneiden henkilöiden tuli ennen ilmoittautumista varmistua, että heillä oli käytössään tietokone, jossa olisi Internet-selain ja Internet-yhteys. Testin suorittajia varten luotiin ohje [Liite 1], jonka mukaan testin suorittajat suorittaisivat hakuja sekä graafisella hakulomakkeella että tekstihakulomakkeella. Testi sisältää seitsemän hakuja. Hauista tehtiin eri hakuehtoja sisältäviä, jotta hakulomakkeiden toimintaa tulisi testattua kohtalaisen kattavasti. Ensimmäinen haku on minimaalinen, sisältäen vain N vaihtoehtoa, jotta testihenkilö saisi heti onnistumisen aikaan ja näkisi heti, miltä hakutulossivu näyttää. Viimeinen haku on monimutkainen, sisältäen monta hakuehtoa. Tässä vaiheessa käyttäjä on oletettavasti oppinut jo hieman sisäistämään tekstihaun syntaksia.

### **4.3 Käyttäjäprofiilit**

Testin suorittajiksi valittiin henkilöitä allekirjoittaneen lähipiiristä. Heillä ei ollut ennakotietoa testin sisällöstä. Testihenkilöistä on miehiä 75 prosenttia.

- Testihenkilö 1 on 36-vuotias nainen. Henkilöllä on tietokoneen käytön perustaidot: web-sovellusten käyttö on tuttua.
- Testihenkilö 2 on 41-vuotias mies. Henkilöllä on useiden vuosien kokemus tietokoneen sovellusten ja web-sovellusten käytöstä.
- Testihenkilö 3 on 39-vuotias mies. Henkilöllä on monen vuoden kokemus tietokoneiden sovellusten ja web-sovellusten käytöstä.
- Testihenkilö 4 on 38-vuotias mies. Henkilöllä on jonkin verran kokemusta tietokoneiden sovellusten käytöstä ja web-sovellusten käytöstä.

Testihenkilöillä tuli olla kokemusta tietokoneen, Internet-selaimen ja web-sovellusten käytöstä. Testijoukko on pieni, mutta kattava validoimaan tämän tutkimuksen kannalta olennaiset selvitettävät asiat.

## 5 Tulokset ja niiden tarkastelu

Käyttäjätettiin osallistui neljä henkilöä. Naisia oli yksi ja miehiä kolme. Testihenkilöiden iät vaihtelevat välillä 36 ja 41. Henkilöiden koulutus oli DI, Fil. maist., ravintola-alan tutkinto, ja auto- ja sähkötekniikan ammattitutkinto.

Kaksi testihenkilöä piti graafista hakulomaketta helpompana ja nopeampana tapana suorittaa hakuja. Kaksi muuta testihenkilöä piti tekstihakua nopeampana ja helpompana tapana suorittaa haku. Käyttäjätestin tuloksista voidaan päätellä, että paremmuutta näistä hakutavoista ei ollut mahdollista selvittää tällä käyttäjämäärällä. Kuitenkin yksi testihenkilö, joka piti graafista hakulomaketta parempana ja nopeampana, oli kirjoittanut palaute-kohtaan yllättyneensä tekstihaun nopeudesta ja tarkkuudesta joillain valinnoilla. Testikäyttäjä kertoi tekstihaun olevan käyttökelpoinen kun tarkkuus paranee. Myös toinen testikäyttäjä, jonka mielestä graafinen haku oli parempi, oli kirjoittanut palaute-osioon: ”tekstihaku soveltuu ihmisille, jotka ovat nopeita kirjoittamaan koneella ja tietävät tarkasti mitä hakevat”. Twenten yliopiston tutkimuksessa testihenkilöt ilmoittivat, että tunsivat tekstihaun hieman nopeammaksi hakutavaksi. Tekstihaun huonoista kokemuksista tuli esiin se, että jos hakusanan ollessa pitkä ja vieraskielinen, niin kirjoitusvirhe tuli helposti. Silloin hakuehtoa ei tunnisteta ja haku palauttaa virheelliset tiedot.

Verrattuna Twenten yliopistossa tehtyyn tutkimukseen tässä tutkielmassa testihenkilöitä oli 76 % vähemmän ja testihenkilöiden iät olivat keskimäärin vanhempia. Twenten yliopistossa tehdyssä tutkimuksessa mitattiin myös, miten kauan haun suorittaminen kesti. Tekstihaun tekeminen oli keskimäärin 9 % nopeampaa. Ajan mittaaminen ei ollut tämän tutkielman testin tavoite, koska ajan mittaaminen olisi tarkoittanut sitä, että testihenkilöt olisivat joutuneet käymään suorittamassa testin paikan päällä.

## 6 Johtopäätökset

Tässä tutkielmassa on tutkittu voiko monimutkaisen web-hakulomakkeen korvata tekstihaulla. Tutkielman pohjana on Twenten yliopiston tutkimus, jossa monimutkaista web-hakulomaketta ja tekstihakua verrataan toisiinsa. Tutkielma sisältää kirjallisen tutkimuksen hakulomakkeiden toteutukseen, sen hyviin ja huonoihin puoliin. Tutkielmassa toteutettiin verkkopalvelu, missä on mahdollista tehdä hakuja sekä graafisella hakulomakkeella että tekstihaulla. Testihenkilöt suorittivat ennalta suunnitellun testin ja arvioivat hakutapoja vastaamalla heille asetettuihin kysymyksiin.

Testin tulokset osoittivat, että yhtä suuri määrä testihenkilöistä piti graafista hakulomaketta ja tekstihakua helpompana ja nopeampana tapana suorittaa hakuja. Se, että kaksi testihenkilöä piti tekstihakua parempana todistaa, että käyttäisivät tekstihakua ennemmin kuin monimutkaista web-hakulomaketta. Se taas osoittaa, että tekstihaulla pystyy korvaamaan monimutkaisen web-hakulomakkeen. Tulokset saattavat johtua siitä, miten testihenkilö mieltää tietokoneen käytettävyyden. Käyttäjillä voi olla tarve nähdä vaihtoehdot tai he vain kokevat olevansa hitaita käyttämään näppäimistöä ja pitävät parempana käyttää hiirtä. Toisaalta käyttäjät eivät ehkä halua opetella tekstihaun syntaksia, koska oppiminen vie aikaa ja vaatii hieman keskittymistä. Mielestäni testihenkilöt, jotka pitivät monimutkaista web-hakulomaketta parempana, käyttivät hakuja ensi kertaa, eikä testin suorittamisen aikana tekstihaun syntaksi ei jää kunnolla mieleen. Graafinen hakulomake on helpompi omaksua, koska käyttäjän ei tarvitse opetella, miten hakuetoja syötetään. Graafinen käyttöliittymä on hyvin havainnollinen ja helppo omaksua. Tekstihaun käyttö tuntui alussa varmasti vaikealta, koska käyttäjä ei tiennyt, millä hakusanoilla hän olisi voinut rajoittaa hakuja. Pitää ottaa kuitenkin huomioon, että pikaohje oli helposti näkyvillä, joilla pääsee helposti alkuun. Arvelen, että jos testihenkilöt olisivat käyttäneet lomakkeita pidemmän aikaa, olisi tekstihaun käyttö tullut tutummaksi ja sitä myöten myös mielipide nopeammasta ja helpommasta hakutavasta vaihtunut.

Jatkokehitysideoita tutkielmalle on monia. Tekstihakua voisi parantaa siten, että se olisi ennakoiva. Sitä mukaa kun hakuetoja kirjoitettaisiin, niitä yritettäisiin samalla tunnistaa ja hakuja suoritettaisiin taustalla saman tien ja hakutulokset päivitetäisiin. Se olisi käyttäjälle paljon parempi käyttökokemus, koska palautteen toiminnoistaan näkee heti. Passiivista hakuja käyttäessä käyttäjä näkee vasta haun suorituksen jälkeen, ovatko kaikki hakuetoja tunnistettu ja kirjoitettu oikein. Toinen parannusidea on käyttää sumeaa hakuja kirjaintarkan täsmävyuden sijasta. Koska hakusanoja on paljon ja jotkut hakusanoista voivat olla pitkiä ja ehkä vaikeitakin kirjoittaa, kirjoitusvirheitä tulee helposti. Kolmas idea on ottaa kattavampi joukko testihenkilöitä suorittamaan testiä. Yli

kymmenen testihenkilön joukko tuottaisi luotettavamman testituloksen verrattuna neljään testihenkilöön.

Testin suorittamista voisi parantaa siten, että hakujen suorittamisen kesto mitattaisiin. Mitataan ajat silloin kun käyttäjät suorittavat testin ensi kertaa ja sitten kun on käytetty hakuja useamman testin ajan. Sitä varten tarvittaisiin lisää testejä, missä hakuehdot vaihtelevat nykyisistä. Tällä tavalla testihenkilöt oppivat tekstihaun syntaksin ajan kanssa, niin kuin verkkopalvelua toistuvasti käyttäessä tapahtuukin.



## Lähteet

About nginx. 2013. [WWW]. [Viitattu 1.11.2013] . Saatavissa <http://www.nginx.org/en/>

About PostgreSQL. 2013. [WWW]. [Viitattu 12.10.2013]. Saatavissa: <http://www.postgresql.org/about>.

About SQLite. 2014. [WWW]. [Viitattu 12.10.2014]. Saatavissa <http://www.sqlite.org/about.html>

Achour, M., Betz, F, Dovgal, A., Lopes, N., Magnusson, H., Richter, G., Seguy, D., Vrana, Y., and others. 2014. PHP manual. [WWW]. [Viitattu 20.10.2014]. Saatavissa: <http://php.net/manual/en/index.php>.

Apache HTTP Server. 2014. [WWW]. [Viitattu 4.9.2014]. Saatavissa [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)

Apple. 2013. iOS: Siri. [WWW]. [Viitattu 04.05.2014]. Saatavissa: [http://support.apple.com/kb/HT4992?viewlocale=fi\\_FI&locale=fi\\_FI](http://support.apple.com/kb/HT4992?viewlocale=fi_FI&locale=fi_FI)

Crockford, D. 2008. JavaScript: The Good Parts. O'Reilly. 153 s.

Distrowatch.com. 2014. Linux Distributions. [WWW]. [Viitattu 18.9.2014]. Saatavissa <http://distrowatch.com/search.php?status=All>

Ekonoja, A., Lahtonen, T., Mäntylä, J. 2009. Lomake-elementit – WWW-julkaiseminen. [WWW] [Viitattu 21.1.2014]. Saatavissa: <http://appro.mit.jyu.fi/doc/forms/index2.html>

Google Developers, 2014. The Mirror API. [WWW]. [Viitattu 20.9.2014]. Saatavissa <https://developers.google.com/glass/develop/mirror/index>

Kasurinen, J. P. 2011. Ruby on Rails ohjelmointi. Docendo. 260 s.

Microsoft. 2014. Windows Phone 8.1 new features – Cortana. [WWW]. [Viitattu 04.05.2014]. Saatavissa <http://www.windowsphone.com/en-us/features-8-1#Cortana>

Netcraft.com. 2013. December 2013 Web Server Survey. [WWW]. Saatavissa: <http://news.netcraft.com/archives/2013/12/06/december-2013-web-server-survey.html>

NetMarketShare. 2014. Desktop Top Browser Share Trend. [WWW]. [Viitattu 18.05.2014]. Saatavissa <http://www.netmarketshare.com/>

Tjin-Kam-Jet, K., Trieschnigg, D., & Hiemstra, D. 2011. Free-text search versus complex web forms. In: Advances in Information Retrieval, pp. 670-674. Springer Berlin Heidelberg.

Ubuntu documentation. 2012. Ubuntu Server Guide. [WWW]. [Viitattu 28.1.2014]. Saatavissa: <https://help.ubuntu.com/12.04/serverguide/index.html>

World Wide Web Consortium. 1999. HTML 4.01 specifications Candidate Recommendation. [WWW]. [Viitattu 20.1.2014]. Saatavissa: <http://www.w3.org/TR/REC-html40/>

World Wide Web Consortium. 2013. HTML 5 specifications Candidate Recommendation. [WWW]. [Viitattu 20.1.2014]. Saatavissa: <http://www.w3.org/TR/html5/>

## LIITE 1: Ohje testihenkilöille

### TESTIN ENSIMMÄINEN OSA - GRAAFINEN HAKULOMAKE

Avaa Internet-selain ja mene osoitteeseen <http://projekti.sytes.net/php/form.php>.

Siellä on graafinen hakulomake.

Suorita seuraavat haut:

Haku 1: Hae punaiset Volvo V70:t, joiden polttoaine on diesel.

Haku 2: Hae Skoda Octavia, polttoaineena diesel, vaihteisto automaatti, mittarilukema alkaen 150000 km ja päättyen 190000 km

Haku 3: Hae Audi A4 farmariautot, joiden polttoaine on diesel, vaihteisto automaatti, hinta 20 000e - 25 000e.

Haku 4: Hae beiget Volvot korimallilla hatchback vuosimallit 2001-2003.

Haku 5: Hae valkoiset Ford Mondeot, joiden polttoaine on diesel ja joiden vuosimalli on 2006-2010, sekä joiden korimalli on sedan ja joiden hinta alle 20 000 euroa.

Haku 6: Hae farmariautot, joiden polttoaine on bensiini ja joissa on vetokoukku audioin -liitin, vakionopeuden säädin ja ilmastointi.

### TESTIN TOINEN OSA - TEKSTIHAKU

Mene osoitteeseen <http://projekti.sytes.net/php/text.html> ja suorita samat haut.