

Saimaan ammattikorkeakoulu
Tekniikka Lappeenranta
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

Teppo Pötry

IMPATI-järjestelmän toteutus

Opinnäytetyö 2015

Tiivistelmä

Teppo Pötry

IMPATI-järjestelmän toteutus, 70 sivua

Saimaan ammattikorkeakoulu

Tekniikka Lappeenranta

Tietotekniikan koulutusohjelma

Ohjelmistotekniikka

Opinnäytetyö 2015

Ohjaajat: lehtori Martti Ylä-Jussila, Saimaan ammattikorkeakoulu, työvalmentaja, Juha Rasimus, IntoPajat ry

Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa ennalta määritelty IMPATI-järjestelmä IntoPajat ry:lle. IMPATI-järjestelmä on palvelunohjauksen apuväline.

IMPATI-järjestelmän tavoitteena oli nopeuttaa ja tehostaa työllistymispalveluiden hakua ja niiden tarjoamista asiakkaille. Järjestelmä hakee tietokannasta palveluita annettujen hakuehtojen perusteella ja lajittelee hakutulokset asiakkaalle sopivuuden mukaan.

Järjestelmän määrittelyn oli toteuttanut Sami Repo omassa opinnäytetyössään, joskin määrittely on muuttunut toteutusvaiheen aikana kohtalaisen paljon järjestelmään tulleiden ja siitä poistuneiden ominaisuuksien myötä.

Järjestelmä toteutettiin käyttämällä PHP-, HTML- ja JavaScript-ohjelmointikieliä sekä MySQL-tietokantaa.

Opinnäytetyön tuloksena asiakkaalle tuotettiin toiminnallinen määrittely, vaatimusten mukainen järjestelmä sekä sen ylläpitämiseen vaadittava käyttöohje.

Asiasanat: PHP, JavaScript, MySQL, HTML, tietokanta, web-ohjelmointi

Abstract

Teppo Pötry

Implementation of the IMPATI database application, 70 pages

Saimaa University of Applied Sciences

Technology Lappeenranta

Information Technology Degree Programme

Software Engineering

Bachelor's Thesis 2015

Instructors: Senior lecturer Martti Ylä-Jussila, Saimaa University of Applied Sciences, Youth workshop trainer, Juha Rasimus, IntoPajat ry

The objective of this thesis was to define, design, implement and deploy a web based database system called IMPATI for IntoPajat ry. IMPATI is a service counseling assistance tool.

The goal of IMPATI-system was to speed up and streamline the search of employment services and offer those services to the customers. The System performs database search by given criteria and sorts the search results by suitability for given customers.

Functional specification of the system were made earlier by Sami Repo in his own thesis.

The system was developed by using PHP, HTML, JavaScript and MySQL database.

As a result, the system functional specification and admin operation manual was produced to the customer. The system is scheduled to be deployed during the spring 2015.

Keywords: PHP, JavaScript, MySQL, HTML, database, web-development

Sisällys

Termit ja lyhenteet	6
1 Johdanto	8
1.1 IntoPajat ry.....	9
1.2 Imatran työllistämispalvelut	10
2 Ohjelmistotuotannon käytännöt	12
2.1 Vaihejakomallit.....	12
2.1.1 Vesiputousmalli	12
2.1.2 Protoilu	13
2.1.3 Iterointi	13
2.1.4 Ketterät menetelmät	14
2.2 Vaatimusmäärittely	15
2.3 Ohjelmistosuunnittelu.....	15
2.3.1 Arkkitehtuurisuunnittelu	15
2.3.2 Moduulisuunnittelu	16
2.3.3 Tietokannan suunnittelu	16
2.4 Integrointi	17
2.5 Testaus	17
2.6 Versionhallinta	18
2.7 Projektinhallinta	19
2.8 Vaatimustenhallinta	20
2.9 Dokumentointi.....	20
2.10 UML 2.....	21
3 Tietokantaperustaisen verkkopalvelun toteutustekniikat ja välineet	29
3.1 Verkkopalvelu	29
3.2 Internet.....	30
3.3 Selain ja web-palvelin	32
3.4 Relaatiotietokanta	33
3.5 Web-hotelli	34
3.6 PHP	34
3.7 HTML.....	34
3.8 JavaScript.....	35
3.9 CSS	36
4 Käytetyt kehitysvälineet	36
4.1 Adobe Dreamweaver	36
4.2 StarUML.....	37
4.3 Notepad++	38
4.4 MySQL Workbench.....	39
4.5 cPanel.....	40
4.6 phpMyAdmin.....	41
5 IMPATI-projekti	42
5.1 Esitutkimus ja projektisuunnittelu	42
5.1.1 Organisaatio	43
5.1.2 Toimintatapa	46
5.2 Toiminnallinen määrittely	47
5.3 Suunnittelu	47
5.3.1 Arkkitehtuurisuunnittelu	47
5.3.2 Moduulisuunnittelu	48

5.4	Toteutus.....	48
5.5	Testaus.....	49
5.6	Käyttöönotto.....	49
6	Lopputuotteen esittely.....	50
6.1	Yleiskuvaus.....	50
6.2	Järjestelmän rakennekaavio	51
6.3	Tietokanta.....	51
6.4	Käyttötapaukset	52
6.5	Sisäänkirjautuminen	54
6.6	Etusivu	55
6.7	Palveluiden haku	56
6.8	Palveluiden listaus	57
6.9	Palvelun näyttö	57
6.10	Omien tietojen hallinta	59
6.11	Uloskirjautuminen	59
6.12	Uutistenhallinta	60
6.13	Käyttäjähallinta	61
6.13.1	Käyttäjän lisääminen	62
6.13.2	Käyttäjän tietojen muokkaaminen	63
6.13.3	Käyttäjän poistaminen.....	63
6.14	Palveluhallinta	64
6.14.1	Palvelun lisääminen	64
6.14.2	Palvelun muokkaaminen	65
7	Yhteenveto.....	67
	Kuvat.....	68
	Lähteet.....	69

Termit ja lyhenteet

Termi	Kuvaus
Ajax	Asynchronous JavaScript and XML, tekniikka jonka avulla voidaan suorittaa pieniä kyselyitä asiakkaan (Client) ja palvelimen (server) välillä, siten että koko sivua ei tarvitse ladata uudelleen.
Asiakas	Peruskäyttäjän asiakas (yksittäinen henkilö), jolle palvelua ollaan hakemassa. Esimerkiksi sosiaalitoimen asiakas. Asiakas ei itse käytä järjestelmää.
Asiakkuuspaikka	Asiakkuuspaikalla tarkoitetaan jotakin yritystä, yhdistystä, organisaatiota, tms. johon asiakkaalla on asiakkuussuhde.
Avainsana	Järjestelmään tallennettua palvelua kuvaava yksittäinen sana.
CAPTCHA	Lyhenne sanoista "Completely Automated Public Turing test to tell Computers and Humans Apart". Kuvavarmennus, jonka avulla varmistetaan, että järjestelmän käyttäjä on ihminen.
CKEditor	Avoimen lähdekoodin wysiwyg-tekstieditori. Käytetään web-sivuilla html-muotoisen koodin tuottamiseen ja muokkaamiseen.
cPanel	cPanel on yhdysvaltalaisen cPanel Inc:n kehittämä Unix-pohjainen sovellus, jolla voidaan ylläpitää niin sanottuja hosting-palveluita.
CGI	Common Gateway Interface, web-teknologia, selaimen ja palvelimen väliseen datan välitykseen.
CSS	Cascading Style Sheets, web-sivuilla käytettävä tyyliohjelaji, jonka avulla määritellään sivuston ulkoasu.
DNS	Domain Name System, maailman laajuinen verkotunnusjärjestelmä, joka muuntaa verkkotunnuksia IP-osoitteiksi.
Eksote	Etelä-Karjalan sosiaali- ja terveystieteiden tutkimuskeskus.
Flexigrid	Avoimen lähdekoodin käyttöliittymäkomponentti tiedon esittämiseen. Perustuu jQuery-kirjastoon.
IMPATI	Imatran palvelutietokanta -järjestelmä.
DOM	Document Object Model, puumainen rakenne esimerkiksi HTML-dokumentista.

DNS	Domain Name System, internetin nimipalvelujärjestelmä, joka muuntaa verkkotunnuksia IP-osoitteeksi.
Dreamweaver	Adobe Sytems in kehittämä web-sivujen luontiin tarkoitettu ohjelma.
Dropbox	Tiedostojen synkronointiin, siirtoon ja jakoon tarkoitettu pilvipalvelu.
ER-kaavio	Entity-relationship model, käsitekaavio, jolla kuvataan yksilötyypit ja niiden suhteet.
HTML	HyperText Markup Language, web-sivujen kuvauskieli.
HTTP	Hypertext Transfer Protocol, selaimen ja palvelimen välinen tiedonsiirtoprotokolla.
JavaScript	Selaimessa suoritettava komentosarjakieli.
JSON	JavaScript Object Notation, yksinkertainen avoimen standardin tiedostomuoto tiedonvälittämiseen. Käytetään yleisesti ajax-kyselyiden yhteydessä.
jQuery	Avoimen lähdekoodin JavaScript-kirjasto.
MySQL	My Structured Query Language, relaatiotietokanta-ohjelmisto.
Notepad++	Avoimen lähdekoodin tekstieditori.
PHP	Hypertext Preprocessor, web-sivuilla käytettävä palvelinpuolen komentosarjakieli.
phpMyAdmin	Selainpohjainen MySQL-hallintatyökalu.
StarUML	UML-kaavioiden luontiin tarkoitettu ohjelma.
TCP/IP	Transmission Control Protocol/Internet Protocol.
UML	Unified Modeling Language, standardoitu graafinen mallinnuskieli.
Ylläpitäjä	Henkilö, joka ylläpitää tietokantaa ja järjestelmää.
WYSIWYG	What You See Is What You Get, käytetään yleensä sisällön muokkauksen yhteydessä, jossa muokattava sisältö näyttää samalta kuin lopputulos.

1 Johdanto

Opinnäytetyön tarkoituksena on suunnitella ja toteuttaa asiakkaana toimivalle IntoPajat ry:lle sekä muille työllistämispalveluja Etelä-Karjalan alueella tarjoaville tahoille IMPATI-järjestelmä.

Perimmäinen ongelma, josta projekti on lähtenyt liikkeelle, on nykyinen toimintatapa, joka on täysin manuaalinen, joskin osa organisaatioista käytti ainakin osittain sähköisiä järjestelmiä. Asiakasorganisaatiot tarjoavat palveluja asiakkaalle itsenäisesti. Projektin tarkoituksena on kehittää eri organisaatioiden palveluille keskitetty ja yhdistetty tietovarasto, joka toimii samalla uutisointialustana eri organisaatioiden välillä ja tällä tavoin tehostaa työskentelyä.

Järjestelmän avulla palveluohjauksessa toimivat työntekijät voisivat tarjota asiakkailleen heidän tarpeidensa mukaisia ja heille parhaiten soveltuvia palveluja helposti ja nopeasti. Palveluohjauksessa toimiva henkilö voisi hakea järjestelmästä palveluja hakuehtojen mukaisesti, tarkastella tietokannassa olevia palveluita ja tulostaa palvelun esitteen asiakkaalle. Käytössä ei ole aiempaa järjestelmää, joten palveluiden etsintä ja tarjoaminen asiakkaalle on toteutettu manuaalisesti.

Järjestelmän suurimpana hyötynä on palveluiden keskitetty hallinta, jolloin palvelu on heti käytettävissä kaikilla organisaatioilla lisäämisen jälkeen. Ylläpitäjällä olisi myös mahdollisuus lisätä uutinen järjestelmään ja näin tiedottaa eri organisaatioita ja niiden työntekijöitä esimerkiksi uuden palvelun lisäämisestä.

Järjestelmä tulee ensimmäisessä vaiheessa ainoastaan IntoPajat ry:n käyttöön, mutta laajenee todennäköisesti myös muihin organisaatioihin eri hankkeiden kautta. Tällaisia organisaatioita ovat muun muassa paikalliset toimijat, kuten TyönVuoksi ry, Kansaneläkelaitos (Kela), Työ- ja elinkeinotoimisto, sekä Imatran kaupungin mielenterveys-, sosiaali- sekä vammaispalvelut. Järjestelmä on myös käytettävissä kaikilla toimijoilla, jotka tekevät palvelunohjausta työllistämispalveluita tarvitseville henkilöille. Järjestelmän opastuksesta, ohjauksesta ja koulutuksesta loppuasiakkaille vastaa IntoPajat ry.

1.1 IntoPajat ry

IntoPajat ry (entinen INTOa Työstä ry) on Imatralla vuonna 1998 perustettu yleishyödyllinen yhdistys. Yhdistyksen toiminta sai alkunsa sairaanhoitopiirin alaisuudessa toimineessa psykiatrisen kuntoutuksen hoidollisen työn yksikössä, kun yksikön toimintaa tahdottiin alkaa kehittämään yhdistyspohjalta. Toiminta perustuu erilaisiin asiantuntijapalveluihin sekä työpajoilla tehtävään työ- ja yksilövalmennukseen. Yhdistyksen toiminta-ajatuksena on tuottaa ja kehittää laadukkaita työpaja- ja valmennuspalveluita asiakkaiden tarpeet huomioiden yhteistyössä viranomaisten kanssa. Yhdistyksessä toimii useita erilaisia työpajoja: Ruokapalvelu, Tekstiilipaja, Keikkapaja, Kotityöpaja ja ATK-paja. (IntoPajat ry 2014. Tietoa yhdistyksestä.)

Ruokapalveluyksikön työvalmennuksesta vastaa keittiöalan ammattilainen yhdessä apuvalmentajan kanssa. Yksikön yhteydessä toimii lounasruokala sekä kahvila. Yksikkö tarjoaa perehdytystä ja harjoittelua seuraaviin työtehtäviin: laitoskeittiön eri työmenetelmät ja koneiden käyttö, tilausleivonta, sekä kahvila-työ/asiakaspalvelu. (IntoPajat ry 2014. Työpajat.)

Tekstiilipajassa tehdään tilaustöitä asiakkaille sekä omaa tuotantoa, joita myydään yksikön alaisuudessa toimivassa IntoPuoti-myymälässä. Yksikössä pääsee harjoittelemaan muun muassa ompelua, kudontaa, kodintekstiilien valmistusta, huovutusta, somistusta ja neulontaa. (IntoPajat ry 2014. Työpajat.)

Keikkapajalla tehdään pääasiallisesti tilaustöitä yrityksille, yksityisille sekä yhteisöille. Keikkapaja vastaa myös IntoPajat ry:n toimitiloihin kohdistuvista kunnostuksista. Yleisimpiä työtehtäviä ovat muutot, pienimuotoiset kiinteistön huolto- ja korjaustyöt, nurmikoiden leikkuu, haravointi, lumityöt, rakennusaputyöt sekä kuljetus- ja kuriiripalvelut. (IntoPajat ry 2014. Työpajat.)

Kotityöpalveluyksikössä suoritetaan koti- ja siivouspalveluja yksityisille, yrityksille sekä yhteisöille. Yleisimpiä työtehtäviä ovat yksityiskotien ja pienyritysten siivous, ikkunoiden pesu, muutto- ja remonttisiivous sekä lattioiden koneellinen pesu ja vahaus. (IntoPajat ry 2014. Työpajat.)

ATK-ohjaus sisältyy lähes jokaisen Intopajalla työskentelevän valmennusohjelmaan. Vaatimuksena ATK-pajalle harjoitteluun edellytetään joko koulutusta tai vahvaa osaamista alalta. ATK-pajan tehtäviin kuuluvat muun muassa IntoPajat ry:n oma tiedotus, markkinointimateriaalin tekninen toteutus, henkilökunnan tekninen tuki sekä laitehankintojen konsultointi. (IntoPajat ry 2014. Työpajat.)

1.2 Imatran työllistämispalvelut

Imatran työllistämispalvelujen tehtävänä on ennalta ehkäistä sekä vähentää pitkäaikais- ja nuorisotyöttömyyttä. Työllistämispalveluiden tavoitteena on tarjota nuorille työttömille sekä pitkäaikaistyöttömille monipuolisia palveluja sekä reittejä avoimille työmarkkinoille ja näin vähentää pitkäaikaistyöttömyydestä aiheutuvia kustannuksia yhteiskunnalle. Työllisyyttä edistäviä palveluja ovat esimerkiksi kuntouttava työtoiminta, työkokeilu sekä palkkatuettu työ. Työllistämispalvelut koostuvat Etelä-Karjalan työvoiman palvelukeskuksesta (TYP), Imatran kaupungin omista työllistämispalveluista, työllisyyden kuntakokeilusta sekä ostopalveluista. Ostopalveluita Imatran kaupungille tuottavat TyönVuoksi ry sekä IntoPajat ry. (Työllisyyden toimijakartta.)

Työvoiman palvelukeskus

Työvoiman palvelukeskus (lyhenne TYP) on työ- ja elinkeinotoimiston, kuntien, Kelan ja Etelä-Karjalan sosiaali- ja terveystoimiston yhdessä muodostama asiantuntijapalveluita tarjoava toimipiste. Toimipaikat sijaitsevat Lappeenrannassa, Imatralla sekä Parikkalassa. Toiminta-alueena on koko Etelä-Karjala (Etelä-Karjalan työvoimanpalvelukeskus). Asiakkaat ohjautuvat Työvoiman palvelukeskukseen työ- ja elinkeinotoimiston, sosiaalitoimiston sekä Kelan kautta (Työllisyyden toimijakartta).

TyönVuoksi ry

TyönVuoksi ry tarjoaa pääasiassa kuntouttavaa toimintaa sekä palkkatuettua työtä sekä erilaisia valmennuspalveluja, kuten: työ- ja yksilövalmennus, palveluohjaus, starttivalmennus ja työhönvalmennus. Toiminta-alueena on Imatran seutukunta. (TyönVuoksi ry. Valmennuspalvelut.)

TyönVuoksi ry:ssä toimii myös osuuskunta TyönVuoksi, joka tarjoaa muun muassa siivouspalveluita, ikkunoiden pesua, ruoanlaitto- ja leivontapalveluja, kauppa- ja asiointipalveluja. (TyönVuoksi. Osuuskunta TyönVuoksi.)

Etelä-Karjalan työllisyyspoliittinen kuntakokeilu

Hankkeen lähtökohtana on alentaa rakennetyöttömyyttä sekä tukea pitkään työttömänä olleiden työllistymistä. Pyrkimyksenä on ottaa tämän hetkistä paremmin huomioon työttömien työnhakijoiden ja paikallisten työmarkkinoiden tarpeet sekä suunnitella erilaisia palveluita yhteistyössä eri organisaatioiden kanssa. Käytännössä tämä tarkoittaa yritys yhteistyötä sekä asiakaspalveluprosessien ja uusien toimintamallien kehittämistä. Hanke päättyy 31.12.2015. (Kuntakokeilu. Hanke.)

Keskeisimpiä yhteistyöorganisaatioita kuntakokeilussa ovat Kaakkois-Suomen työ- ja elinkeinotoimisto, Etelä-Karjalan työvoiman palvelukeskus, Kela, Eksote ja Imatran kaupungin sosiaali- ja terveystoimi. Näiden lisäksi yhteistyötä pyritään tekemään paikallisten yrittäjä yhdistysten, kolmannen sektorin sekä oppilaitosten kanssa. (Kuntakokeilu. Taustaorganisaatiot.)

Imatran kaupungin TOIMARI-hanke

TOIMARI-hanke on työllisyyden edistämiseen tähtäävä hanke, jossa kehitetään työllistämisprosessia sekä vaikeasti työllistyvien palveluita. Hankkeen pääkohde-ryhmänä ovat nuoret ja heikossa työmarkkina-asemassa olevat henkilöt. Hankkeessa työllistymisen tukeminen tapahtuu vaiheittaisena prosessina, jossa keskitytään työttömän terveyteen, motivaatioon sekä toimintakykyyn ja pyritään rakentamaan väyliä avoimille työmarkkinoille tai opiskelulle. (TOIMARI-hanke 2014. Esite.)

2 Ohjelmistotuotannon käytännöt

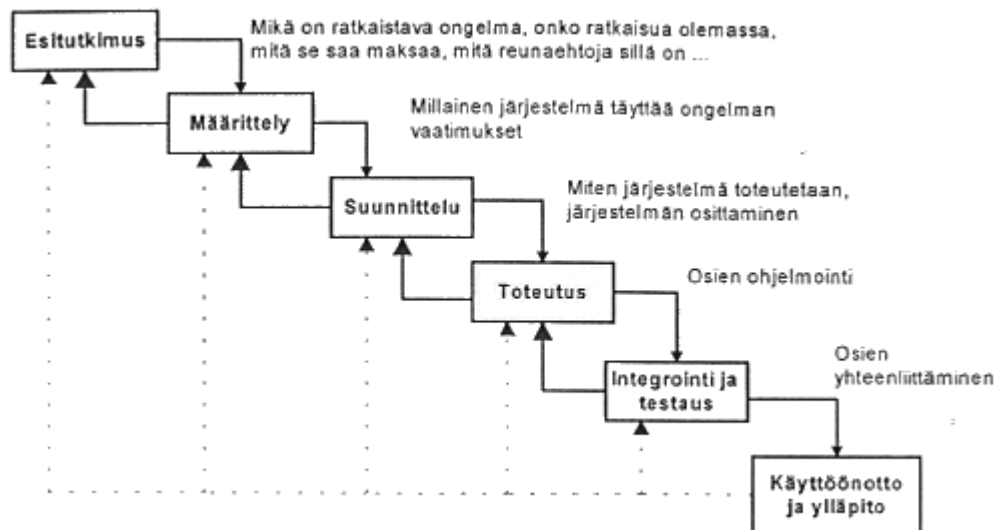
Ohjelmistoprojektiin katsotaan yleisesti liittyvän ainakin määrittelyä, suunnittelua, ohjelmointia ja testausta sekä käyttöönottoon ja ylläpitoon liittyviä asioita. Tässä kappaleessa käydään läpi tärkeimpiä käytäntöjä ohjelmistoprojekteissa.

2.1 Vaihejakomallit

Ohjelmistotuotannon avuksi on vuosien varrella kehitetty useita projektimalleja, joista osa on yleiskäyttöisiä ja osa IT-projekteihin tarkoitettuja. Teollisuudessa yleisesti käytössä olevia projektimalleja ovat vesiputousmalli, protoilu sekä erilaiset ketterät menetelmät. (Haikala & Mikkonen 2011, s.34.)

2.1.1 Vesiputousmalli

Vuonna 1970 Royce esitteli ohjelmistoalan eniten viitatuista sekä eniten väärinymmärretyistä artikkeleista, jonka keskeisin sisältö oli vesiputousmalli (Kuva 1). Royce pitää iterointia taaksepäin mallin tärkeimpänä ominaisuutena. Royce toteaa myös artikkelissaan, että jos järjestelmän ongelmat ja viat paljastuvat vasta testausvaiheessa, kustannusarvio voi jopa kaksinkertaistua. Sallimalla iterointi ja vaiheiden käynnistäminen ennen edellisen vaiheen loppua saadaan useisiin tilanteisiin sopiva toimintamalli. Tarkasteltaessa uudempia projektimalleja, on niiden sisältä löydettävissä vesiputousmallin alkuperäinen idea. (Haikala & Mikkonen 2011, s.37.)



Kuva 1. Vesiputousmalli (Haikala & Märijärvi 2004)

2.1.2 Protoilu

Protoilulla tai prototyypillä tarkoitetaan ohjelmistotuotannossa, jollain lailla vaillinaisen ohjelmiston tai sen osan rakentamista jonkin toiminnan tai ominaisuuksien tarkempaa tutkimista varten. Esimerkkeinä kohteista ovat käyttöliittymälogiikka tai suorituskyky. (Haikala & Mikkonen 2011, s.38.)

Protoilussa on kaksi päävaihtoehtoa: evoluutioprototyyppi tai poisheitettävä. Nimensä mukaisesti evoluutioprototyyppiä kehitetään vaiheittain valmiiksi tuotteeksi ja poisheitettävää käytetään järjestelmän mallintamiseen, jonka jälkeen kehitys aloitetaan alusta. Evoluutioprototyypin huonoina puolina pidetään, että huonosti toimiva prototyyppi jää osaksi lopullista järjestelmää. Poisheitettäviä prototyyppisiä käytetään useasti käyttöliittymien suunnittelussa. (Haikala & Mikkonen 2011, s.38–39.)

2.1.3 Iterointi

Tuotekehityshankkeissa on tyypillistä, että ne etenevät tuotekehityssykleissä: järjestelmästä julkaistaan uusilla ominaisuuksilla oleva versio esimerkiksi kerran vuodessa. Uuden version kehityssykli käynnistyy seulomalla edellisestä versiosta saadusta palautteesta tärkeimmät työkohteet. Kehityssyklit voidaan nähdä toisiaan seuraavina vesiputouksina. Tähän lähestymistapaan liittyy kuitenkin

useita ongelmia, kuten riskienhallinnan monimutkaisuus ja riskien ennalta-arvaamattomuus, vaatimusten tarkentuminen ja myöhäinen asiakaspalaute, itsepetos sekä kärsimättömät asiakkaat. Edelliset ongelmat tulevat esille jo tavallisessa vesiputousmallissa, mutta korostuvat entisestään toistuvissa vesiputouksissa. (Haikala & Mikkonen 2011, s.40–41.)

Ratkaisuna ongelmaan on iterointi, jossa tuotekehityssyklin pituutta lyhennetään yhdestä vuodesta esimerkiksi yhteen kuukauteen. Tällöin toteutettavat vaatimukset voidaan jakaa iteraatioihin, ja jokaisen iteraation lopussa voidaan demostroida siihen asti toteutettuja vaatimuksia. (Haikala & Mikkonen 2011, s.42.)

2.1.4 Ketterät menetelmät

Vesiputousmallien heikkoudet ovat johtaneet viime vuosina yleistyneiden ketterien kehitysmenetelmien kehittelyyn ja käyttöönottoon. Ketterissä menetelmissä lähtökohtana on oletus, että vaatimuksia ei voi tyhjentävästi määrittellä ohjelmistokehitysprosessin alussa, joten sitä ei edes yritetä vaan pyritään saamaan toimiva lopputulos jatkuvasti muuttuvista vaatimuksista huolimatta. (Luukkainen & Laine 2010.)

Ketterä ohjelmistokehitys etenee yleensä siten, että ensin kartoitetaan pääpiirteissään vaatimuksia ja arkkitehtuuria, jonka jälkeen suoritetaan useita iteraatioita, joiden kunkin aikana järjestelmään valitaan suunniteltavaksi ja toteutettavaksi osa järjestelmän vaatimuksista. Yksittäinen iteraatio, joka voi olla kestoltaan muutamista viikoista pariin kuukauteen lisää järjestelmään pienen osan toiminnallisuudesta. Yleensä tärkeimmät ja riskialtteimmat toiminnallisuudet toteutetaan ensin. Yksi iteraatio sisältää vaatimusten tarkennuksen, suunnittelun, toteutuksen sekä testauksen. Asiakas pääsee testaamaan ja koekäyttämään järjestelmää, jolloin voidaan aikaisessa vaiheessa todeta onko kehitystyö menossa oikeaan suuntaan. (Luukkainen & Laine 2010.)

Ketterissä menetelmissä dokumentointi ei ole niin keskeisessä osassa kuin perinteisissä menetelmissä. Vähäisen dokumentaation sijaan testauksella ja jatkuvalla integroinnilla on suuri merkitys. Uudet komponentit pyritään testaamaan välittömästi ja ne pyritään integroimaan kokonaisuuteen. (Luukkainen & Laine 2010.)

Tunnetuimpia ketteriä menetelmiä ovat XP eli Extreme Programming ja Scrum. (Luukkainen & Laine 2010.)

2.2 Vaatimusmäärittely

Määrittelyvaiheessa asiakasvaatimuksia analysoidaan ja niistä johdetaan ohjelmistovaatimukset, jotka määrittelevät toteutettavan järjestelmän. Käytännössä määrittelyvaiheeseen liittyvät tehtävät ovat projektin tarpeellisuuden ja toteuttamiskelpoisuuden selvittäminen, tavoitteiden ja vaatimusten asettaminen sekä ratkaisumallin laatiminen. Määrittelyvaiheen tuloksena syntyneitä dokumentteja kutsutaan toiminnalliseksi määrittelyksi, johon sisällytetään vaatimukset että vaatimukset täyttävän järjestelmän kuvaus. Määrittelyvaihe jaetaan usein kahteen osaan: asiakasvaatimusten kartoittamiseen ja ohjelmistovaatimusten määrittelyyn. Asiakasvaatimusten kartoittamista kutsutaan usein esitutkimukseksi tai tarvekartoitukseksi. (Haikala & Märijärvi 2004, s.38–39, s.78–81.)

2.3 Ohjelmistosuunnittelu

Ohjelmistotekniikan yleinen perusongelma on järjestelmän vaatimusten ja käytävissä olevan toteutusteknologian välissä olevan kuilun ylittäminen ja niiden yhteensovittaminen. (Haikala & Mikkonen 2011, s.177.)

Keskeinen keino tämän kuilun ylittämiseen on arkkitehtuurisuunnittelu, jonka tarkoitus on muuntaa järjestelmän kuvaus ongelmasta (toiminnallinen määrittely) ratkaisuun (tekninen määrittely). Arkkitehtuurisuunnittelun jälkeen järjestelmä voidaan osittaa pienempiin komponentteihin. Jokaisella komponentilla voi olla oma sisäinen arkkitehtuurinsa. Lopulta päädytään tilanteeseen, jossa ratkaisut voidaan toteuttaa ohjelmointikielen rakenteiden avulla. (Haikala & Mikkonen 2011, s.177.)

2.3.1 Arkkitehtuurisuunnittelu

Ohjelmistojen yhteydessä termi ”arkkitehtuuri” nousee keskeiseksi termiksi. Yksinkertaisimmillaan se kattaa ohjelmiston eri komponentit ja niiden väliset suhteet, mutta usein mukaan ajatellaan kuuluvan myös lyhyt kuvaus suunnitteluperiaatteista sekä suuntaviivat jatkokehitykselle. Toinen arkkitehtuuria kuvaava

määritelmä on, että se on osa ohjelmistoa, joka ei muutu normaalin ylläpidon aikana. Eli arkkitehtuuri tarjoaa puitteet kehitykselle, testaukselle ja ylläpidolle, muuta niiden aikana siihen ei tule muutoksia. (Haikala & Mikkonen 2011, s.178.)

Arkkitehtuurisuunnittelun tehtävänä on määritellä järjestelmän keskeinen terminologia sekä työnjako komponenttien välillä. Pohjimmiltaan kyse on monimutkaisen kokonaisuuden hallinnasta arkkitehtuurin määrittelemän käsitteistön avulla. Tavoitteena on rakentaa järjestelmä siten, että sen komponentit riippuvat mahdollisimman vähän toisistaan, jolloin muutokset eivät vaikuta muihin komponentteihin. Tämä helpottaa muutoksien tekemistä sekä mahdollistaa komponenttien toteuttamisen toisistaan riippumatta. (Haikala & Mikkonen 2011, s.178–179.)

Riippuvuuksien hallinnassa käytettäviä tekniikoita ovat paikallisuus, hierarkkisuus, modulaarisuus, rajapintojen tiiviys, tiedon kätöntä ja abstraktit tietotyypit. (Haikala & Mikkonen 2011, s.178.)

2.3.2 Moduulisuunnittelu

Ohjelman abstraktiot toteutetaan ohjelmamoduuleina. Yksi moduuli kuvaa yleensä joko yhden abstraktion tai se koostuu joukosta yhteenkuuluvia aliohjelmiä. Ohjelmassa käytettäviä globaaleja vakioita kootaan myös usein yhteen moduuliin. Useimmiten moduulijako vastaa samalla ohjelman jakoa tiedostoiksi. Moduulin kuvauksessa rajapinnan kuvaus tulee erottaa sisäisestä toteutuksesta. (Haikala & Märijärvi 2004, s.319.)

2.3.3 Tietokannan suunnittelu

Tietokannan suunnittelun ensimmäinen vaihe on informaation analysointi käsitteellisellä tasolla. Tässä analyysissä määritellään tietokannan sisältämät tiedot yksittäisiin kokonaisuuksiin, entiteetteihin, ja näiden sisältämiin ominaisuuksiin eli attribuutteihin. Entiteettien ja attribuuttien määrittelemisen jälkeen näiden välille määritellään yhteydet, josta syntyy relaatiomalli. Relaatiomallia kutsutaan myös tietokannan loogiseksi malliksi. Seuraava vaihe on fyysisen mallin luominen. Fyysisistä mallia luotaessa on otettava huomioon käytettävän tietokantajärjestelmän rajoitukset sekä muut tarvittavat komponentit. (Varjo 2010, s.13–14.)

2.4 Integrointi

Integroinnilla tarkoitetaan ohjelmistotuotannossa suunnittelun ja toteutuksen jälkeistä moduulien yhteensovittamista toimivaksi kokonaisuudeksi.

2.5 Testaus

Puhekielessä testaus termillä tarkoitetaan yleensä mitä tahansa kokeilemistä. Ohjelmiston testauksen yhteydessä testaus määritellään suunnitelmalliseksi virheiden etsimiseksi ohjelmaa tai se osaa suorittamalla. (Haikala & Mikkonen 2011, s.205.)

Testauksen avulla voidaan osoittaa, että ohjelmassa on virheitä. Virheettömyyttä sen sijaan testauksella ei ole mahdollista selvittää tai osoittaa. Käytännössä testauksen avulla pystytään kattamaan vain häviävän pieni osa kaikista toiminnoista. Tämä ei tarkoita sitä, että testaukseen ei kannata panostaa vaan enemmänkin sitä, että ohjelman tai järjestelmän toimivuuteen ei kannata sokeasti luottaa hyvistä testituloksista huolimatta. (Haikala & Mikkonen 2011, s.205.)

Testauksen yhteydessä virheellä tarkoitetaan poikkeamaa määrittelystä. Johdonmukainen testaus ilman toiminnallista ja teknistä määrittelyä on mahdotonta, sillä toiminnon oikeellisuutta ei voida todentaa. Virheen vakavuus voi vaihdella järjestelmän käytön estävästä virheestä käyttäjää ärsyttävään kosmeettiseen virheeseen. Virheellisen toiminnon suoritus järjestelmässä aiheuttaa vian, joka voi joko korjautua itsekseen tai sen vaikutus voi kumoutua toisen virheen seurauksena. Pahimmassa tapauksessa vika voi aiheuttaa häiriön, joka näkyy järjestelmän ulkoisissa toiminnoissa. (Haikala & Mikkonen 2011, s.205–206.)

Testaukseen liittyvät työvaiheet ovat testauksen suunnittelu (testisuunnitelma, testitapaukset), testiympäristön luonti, testin suorittaminen ja tulosten raportointi sekä tulosten tarkastelu. Näihin työvaiheisiin sekä niihin liittyviin virheiden etsimiseen ja korjaukseen kuluu yli puolet ohjelmistoprojektin resursseista. (Haikala & Mikkonen 2011, s.205.)

Yksikkötestauksessa testaan yksittäinen luokka tai moduuli ja testin suorittaa yleensä moduulin toteuttaja. Integrointitestauksessa yhdistellään moduuleita, jotka muodostavat kokonaisia osajärjestelmiä. Integraatiotestauksen kohteena

on rajapintojen toimivuuden tutkiminen. Integrointitestausta suoritetaan yleensä rinnakkain yksikkötestauksen kanssa. (Haikala & Mikkonen 2011, s.207–208.)

Järjestelmätestauksen kohteena on koko järjestelmä ja sen tuloksia verrataan vaatimusmäärittelyyn, toiminnalliseen määrittelyyn ja käyttöohjeeseen. Järjestelmätestauksen suorittamisesta tulisi vastata kehitystyöstä mahdollisimman riippumaton henkilö. Järjestelmätestauksen yhteyteen voidaan sisällyttää myös kenttätestaus ja hyväksymistestaus. Lisäksi järjestelmätestauksessa testataan järjestelmän ei-toiminnalliset ominaisuudet. Ei-toiminnalliset ominaisuudet testataan kuormitustesteillä, luotettavuustesteillä, asennustesteillä ja käytettävyytstesteillä. (Haikala & Mikkonen 2011, s.208.)

Yleensä suositellaan, että toteuttaja ei itse osallistuisi testaustyöhön. Syynä tähän on se, että toteuttajalla on taipumus osoittaa ohjelman virheetön toiminta virheiden etsimisen sijaan. (Haikala & Mikkonen 2011, s.205–209.)

2.6 Versionhallinta

Tuotekehityksen aikaisen tuotteenhallinnan tehtävänä on taata kehittäjille vakaa työympäristö, jossa projektin työntekijät ja kehittäjät eivät vahingossa häiritse toistensa työskentelyä esimerkiksi korjaamalla tai muuttamalla samanaikaisesti samaa moduulia. Kun vaihetuote on hyväksytty, se tuodaan tuotteenhallinnan piiriin (versio 1.0). Tämän jälkeen muutokset vaativat hyväksynnän ja muutoksista tulee tiedottaa esimerkiksi projektipalaverissa. (Haikala & Märijärvi 2004, s.260–261.)

Versiopuu kasvaa yleensä lineaarisesti: versiota 1.0 seuraa esimerkiksi versio 1.1 ja versio 1.2 ja niin edelleen. Tällaista peräkkäistä versiota kutsutaan revisioksi. Joskus versiopuuhun joudutaan tekemään sivuhaaroja esimerkiksi versio 1.2.1.1, jota kutsutaan variaatioksi. Sivuhaarojen tekemistä tulisi kuitenkin välttää, koska haaroittaminen aiheuttaa ylläpitäjän painajaisen. (Haikala & Märijärvi 2004, s.261.)

Yksinkertaisimmillaan versiohallinta voidaan toteuttaa tiedostohakemiston rakenteesta tehdyllä sopimuksella. Voidaan esimerkiksi sopia, että uusimmat versiot ovat aina tiettyssä hakemistossa. (Haikala & Märijärvi 2004, s.263.)

Suuremmissa ohjelmistoprojekteissa versionhallinta tulisi toteuttaa käyttäen jostain versionhallintaohjelmistoa. Tunnetuimpia avoimen lähdekoodin versionhallintaohjelmistoja ovat Git ja Subversion (SVN).

2.7 Projektinhallinta

Projekti määritellään yleisesti kertaluonteiseksi tehtäväksi, jolla on määrätyt resurssit sekä organisaatio ja jonka toteutus tapahtuu ennalta laaditun aikataulun mukaisesti. Ohjelmistotuotantoon liittyviä projekteja ovat muun muassa esitutkimusprojekti, määrittelyprojekti, toteutusprojekti sekä käyttöönottoprojekti. Suuret projektit ositetaan yleensä pienemmiksi osaprojekteiksi. (Haikala & Märijärvi 2004, s.225.)

Projektinhallintaan liittyviä tehtäviä ovat projektin suunnittelu, käynnistäminen, seuranta, ohjaus sekä projektin päättäminen. Suunnittelun lähtökohtina ovat projektille asetetut tavoitteet ja reunaehdot. Projektin tavoitteet tulee viimeistään projektisuunnitelmassa tarkentaa sellaisiksi, että niiden toteutumista on mahdollista arvioida projektin päätyttyä. (Haikala & Märijärvi 2004, s.226–227.)

Projektin suunnitteluun tehtäviä ovat organisointi, tavoitteiden tarkentaminen, riskien tunnistaminen ja analysointi, käytettävien teknologioiden ja työmenetelmien valinta, tukitoimintojen kuten dokumentoinnin, tuotteenhallinnan ja laadunvarmistuksen suunnittelu sekä projektin osittaminen ja osien aikatauluttaminen sekä työmäärien arviointi. Työmäärien arviointi onkin yksi haastavimpia ja ongelmallisimpia osuuksia projektin suunnittelussa. (Haikala & Märijärvi 2004, s.227.)

Seurantaan ja ohjaukseen liittyvät muun muassa projektipalaverit, projektikatselmuksset, toteutumien kirjaaminen ja projektisuunnitelman tarkentaminen projektin edetessä.. Seurannan tavoitteena on havaita esille tulleet riskit ja ongelmat mahdollisimman aikaisessa vaiheessa, jolloin niihin on helpompi puuttua. (Haikala & Märijärvi 2004, s.228.)

Projektin päättymisen yhteydessä tarpeellinen dokumentaatio ja tiedostot arkistoidaan ylläpitoa varten. Projektista kannattaa tehdä päättymisen yhteydessä loppuraportti, josta selviää mitä projekti tuotti, mikä onnistui ja mikä epäonnistui sekä

mitä kannattaa jatkossa tehdä toisin. Projektin päättymisestä on myös informoiva selkeästi osallistujia. (Haikala & Märijärvi 2004, s.228.)

2.8 Vaatimustenhallinta

Tuotantoprosessin perimmäinen tavoite on päätyä asiakasvaatimuksista asiakasvaatimukset ja sopimukset täyttävään ohjelmistoon. Kaiken edellä mainitun varmistamiseen tähtääviä toimenpiteitä kutsutaan vaatimustenhallinnaksi. Vaatimustenhallinnan tehtävänä on varmistaa, että lopputuote vastaa asiakkaiden vaatimuksia. Lopputuotteessa on oltava kaikki halutut ja vaaditut ominaisuudet eikä mitään muuta. Vaatimustenhallinnan osa-alueita ovat asiakasvaatimusten kartoitus ja analysointi, vaatimusten jäljitettävyyden sekä vaatimusmuutosten hallinta. (Haikala & Märijärvi 2004, s.91–94.)

2.9 Dokumentointi

Ohjelmistotyölle on tyypillistä, että siinä kirjataan projektin aikana syntyneitä tietoja dokumenttien muotoon. Minimidokumentaatio koostuu projektisuunnitelmasta, toiminnallisesta määrittelystä, teknisestä määrittelystä sekä testausuunnitelmasta. Suuremmissa projekteissa ja hyvätasoisissa laatu järjestelmissä dokumentaatio on huomattavasti laajempi. (Haikala & Märijärvi 2004, s.51.)

Vaikka ohjelmistotyö onkin dokumenttien tuottamista, on laadukkaiden dokumenttien tuottaminen yksi ohjelmistotyön heikoimmista hoidetuista osa-alueista. Tilanteeseen joudutaan yleensä aikataulupaineiden kasvaessa, jolloin dokumentointi jää vähemmälle huomiolle. (Haikala & Märijärvi 2004, s.71.)

Ohjelmistonkehitysprosessin dokumentit voidaan jakaa karkeasti seuraaviin ryhmiin: laatukäsikirjaan liittyvät dokumentit, projektinhallintaan liittyvät dokumentit ja tuotedokumentit. (Haikala & Märijärvi 2004, s.71.)

Laatujärjestelmän dokumentteihin kuuluvat muun muassa laatukäsikirja, ohjeistukset ja tietyt palaveripöytäkirjat. Projektidokumentteihin kuuluvat muun muassa sopimukset, projektisuunnitelma, projektin seurantaraportit ja loppuraportit. Tuotedokumentilla tarkoitetaan tiettyyn tuotteeseen tai tuoteperheeseen liittyviä do-

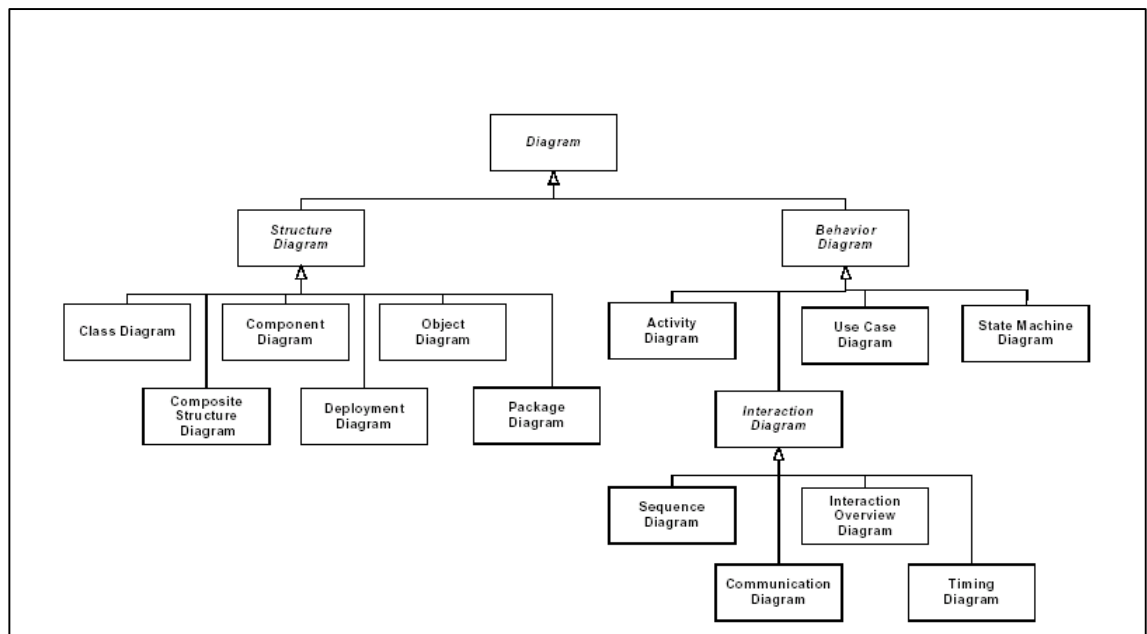
kumentteja. Tuotedokumentit voidaan jakaa vielä projektikohtaisiin tuotedokumentteihin ja tuotekohtaisiin tuotedokumentteihin. (Haikala & Märijärvi 2004, s.71.)

2.10 UML 2

UML (Unified Modeling Language) on mallinnuskieli, jonka avulla kuvataan järjestelmän toiminnallisuuksia eri näkökulmista. UML kehitettiin alun perin 1990-luvulla yhdistämään kolmen tunnetuimman oliomenetelmän käyttämät kuvaus-tekniikat, joita olivat Boochin oliotekniikka, OMT (object-modeling technique) ja OOSE (object-oriented software engineering). (Luukkainen & Laine 2010.)

UML määrittelee joukon kaaviotyyppejä, joita voi käyttää useassa eri tilanteessa. UML ei sinänsä määrittele missä tilanteessa tiettyä kaaviotyyppiä tulisi käyttää. Tämä on kehittämismenetelmien tehtävä. UML on laajennettava kuvauskieli eli käyttäjällä on mahdollisuus lisätä siihen omia piirteitä. (Luukkainen & Laine 2010.)

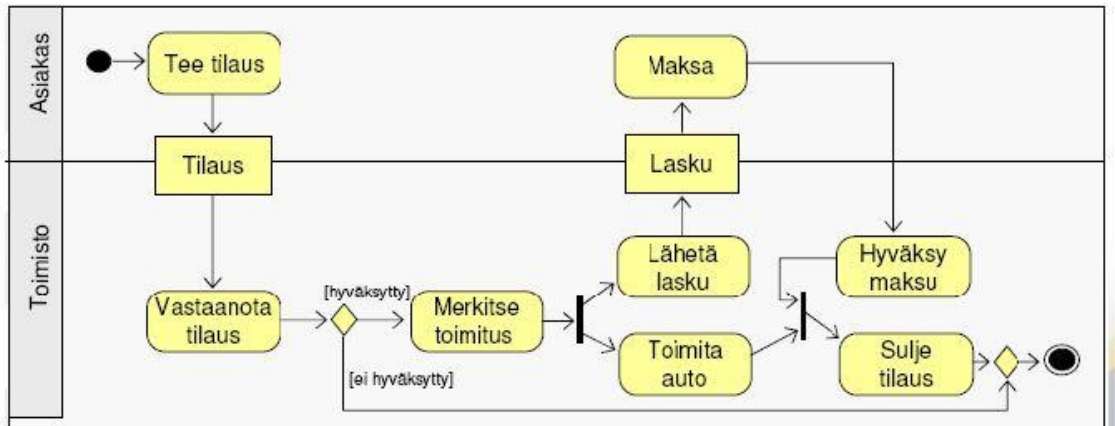
Vuonna 2004 julkaistiin versio 2.0, joka lisäsi kaaviotyyppejä edelliseen versioon. Tällä hetkellä UML:n uusin versio on 2.4.1. Uusimmassa versiossa kaaviotyyppejä on kolmetoista (Kuva 2. UML kaaviohierarkia), jotka esitellään seuraavaksi.



Kuva 2. UML kaaviohierarkia (Wikipedia. UML-mallinnus)

Aktiviteettikaavio (Activity diagram)

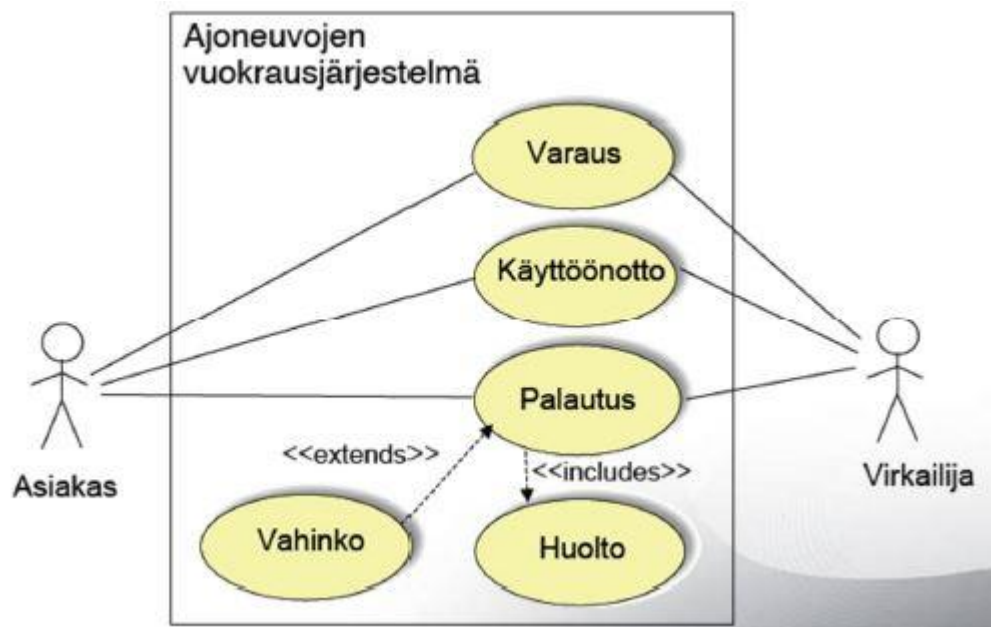
Aktiviteettikaavio (Kuva 3) kuvaa käyttäytymisen ja toiminnan kulun suorituksen edetessä. Käytetään yleensä täydentämään käyttötapauksia. (Häkkinen 2013.)



Kuva 3. Aktiviteettikaavio (Häkkinen 2013)

Käyttötapauskaavio (Use case diagram)

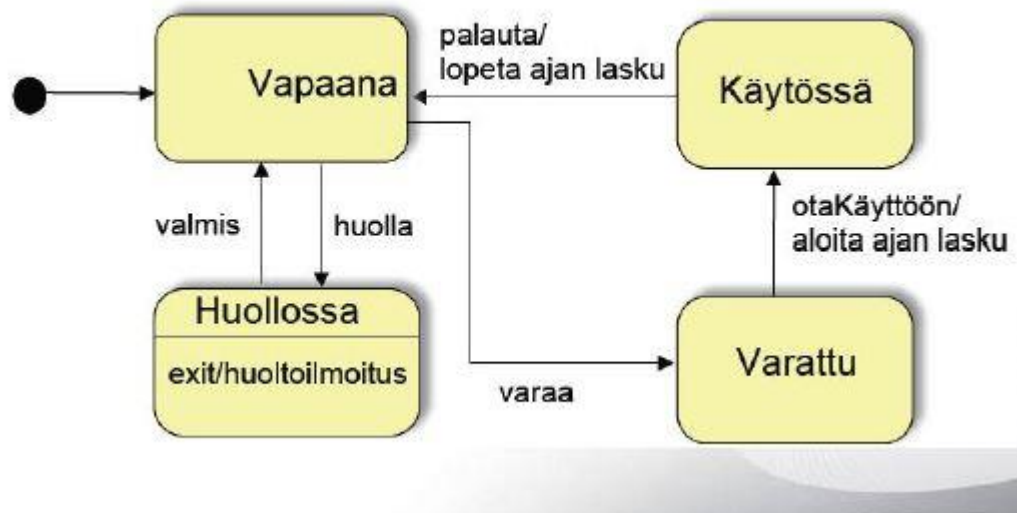
Käyttötapauskaavio (Kuva 4) kuvaa järjestelmää toimijoiden näkökulmasta sekä kiteyttää järjestelmän toiminnalliset vaatimukset. (Häkkinen 2013.)



Kuva 4. Ajoneuvojen vuokrausjärjestelmän käyttötapauskaavio (Häkkinen 2013.)

Tilakaavio (State machine diagram)

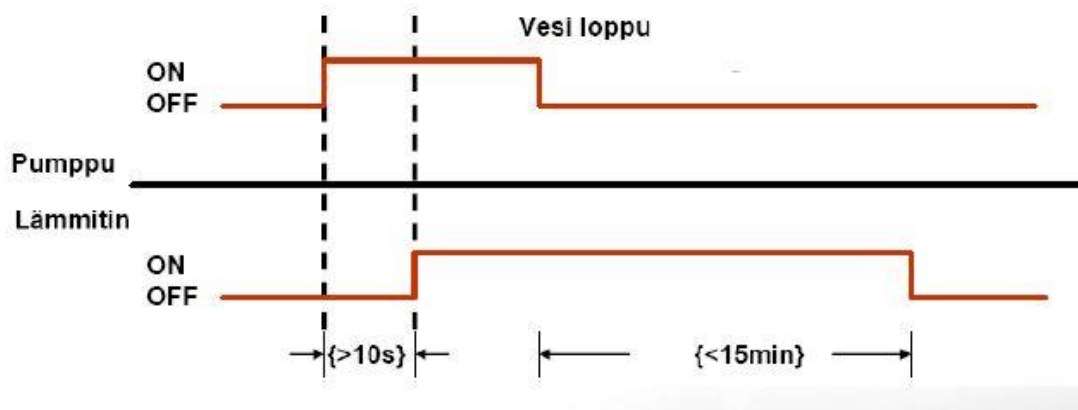
Tilakaavio (Kuva 5) kuvaa olioiden tai järjestelmän tilat sen elinkaaren aikana. (Häkkinen 2013.)



Kuva 5. Tilakaavio (Häkkinen 2013)

Ajoituskaavio (Timing diagram)

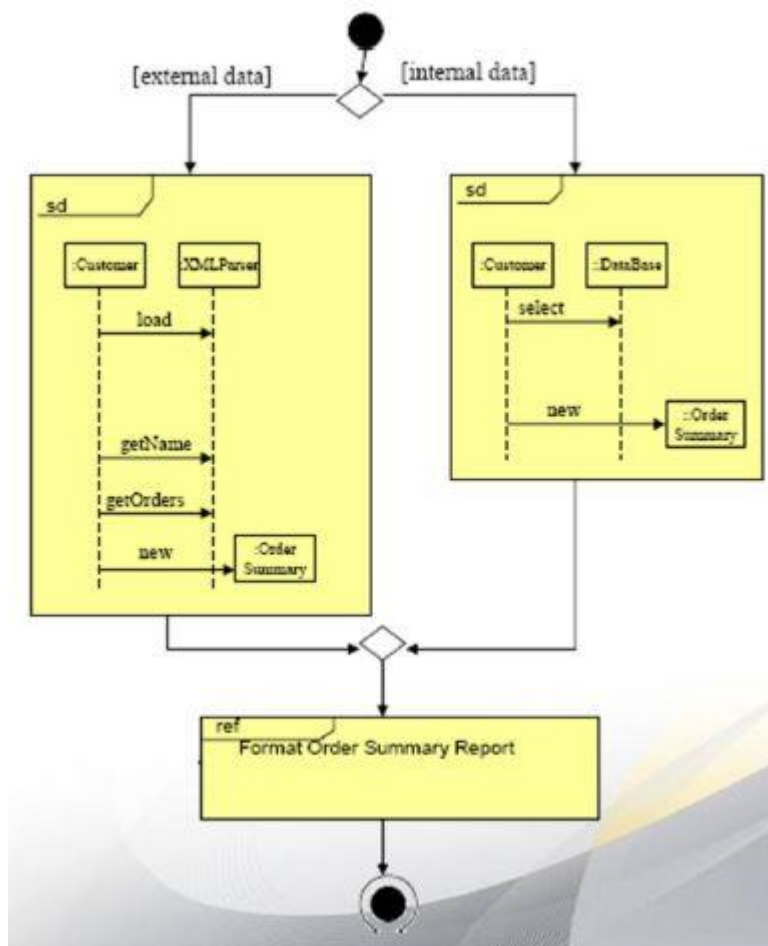
Ajoituskaavio (Kuva 6) kuvaa viestien välityksen ajoituksen järjestelmässä. Selventää vuorovaikutukseen kuluva aikaa. (Häkkinen 2013.)



Kuva 6. Ajoituskaavio (Häkkinen 2013)

Kokoava vuorovaikutuskaavio (Interaction overview diagram)

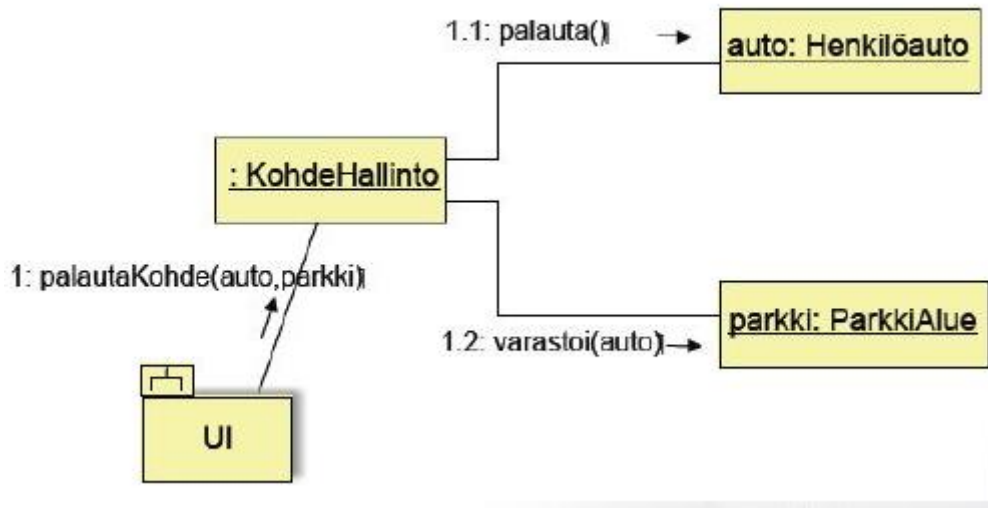
Kokoava vuorovaikutuskaavio (Kuva 7) yhdistää useita sekvenssikaavioita samaan kuvaan. Kuvaa tiettyyn toimintoon liittyvien olioiden suhteita ja vastuita viestinnän aikana. (Häkkinen 2013.)



Kuva 7. Kokoava vuorovaikutuskaavio (Häkkinen 2013)

Kommunikointikaavio (Communication diagram)

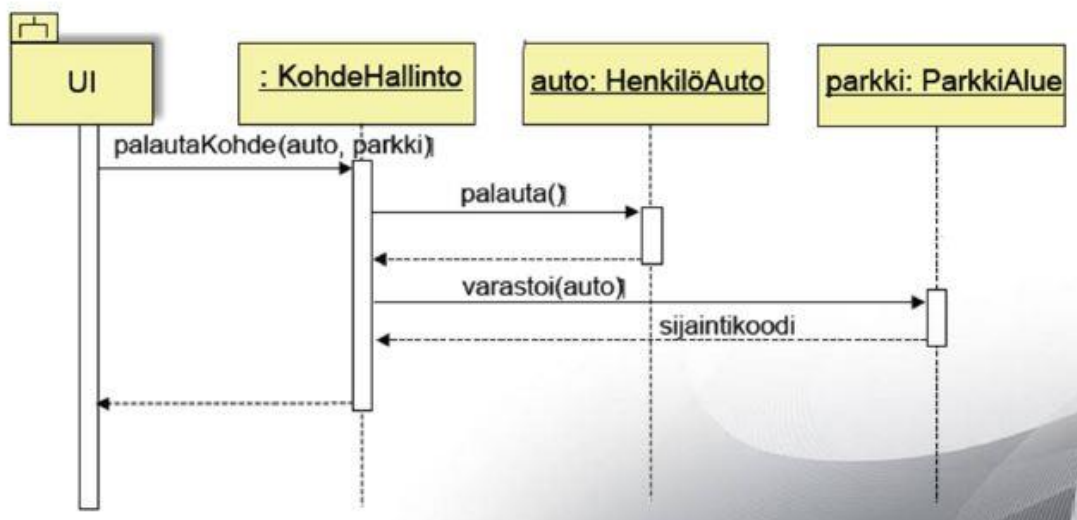
Kommunikointikaaviolla (Kuva 8) kuvataan olioiden suorituksenaikaista yhteistyötä. (Luukkainen & Laine 2010.)



Kuva 8. Kommunikointikaavio (Häkkinen 2013)

Sekvenssikaavio (Sequence diagram)

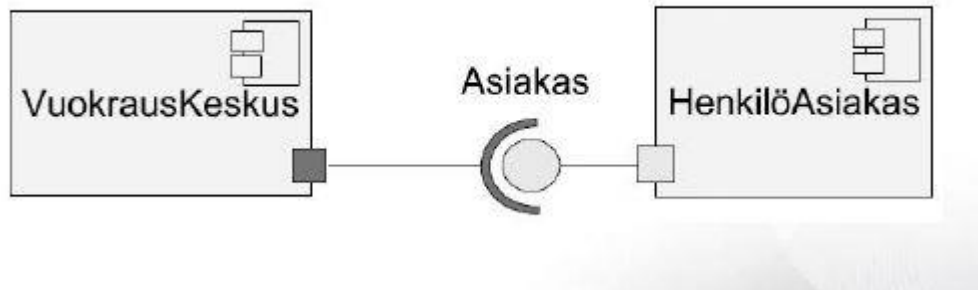
Sekvenssikaavio (Kuva 9) kuvaa kommunikaatiokaavion tapaan olioiden suorituksenaikaista yhteistyötä. (Luukkainen & Laine 2010.)



Kuva 9. Sekvenssikaavio (Häkkinen 2013)

Komponenttikaavio (Component diagram)

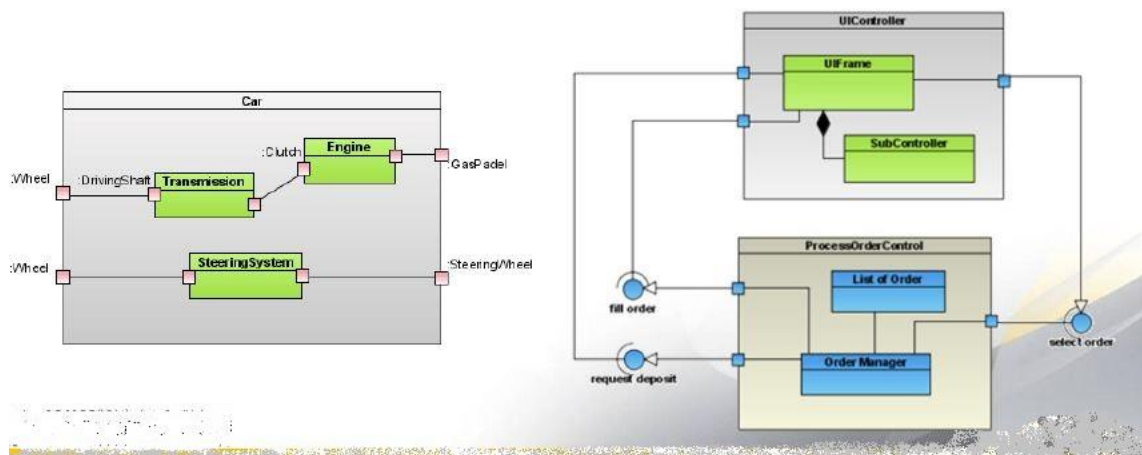
Komponenttikaaviolla (Kuva 10) kuvataan komponenttien väliset toteutus- ja käyttösuhteet sekä niiden tarjoamat ja tarvitsevat rajapinnat. (Häkkinen 2013.)



Kuva 10. Komponenttikaavio (Häkkinen 2013)

Koostekaavio (Composite structure diagram)

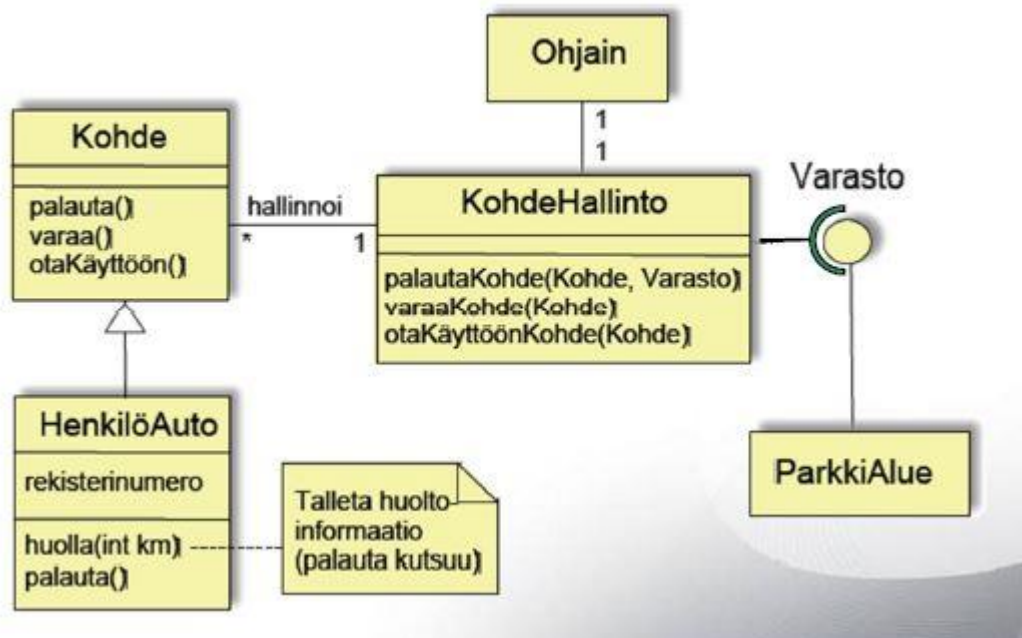
Koostekaavio (Kuva 11) yhdistää luokka- ja komponenttikaaviot. Kuvaa luokkien muodostamat rakenteet. (Häkkinen 2013.)



Kuva 11. Koostekaavio (Häkkinen 2013)

Luokkakaavio (Class diagram)

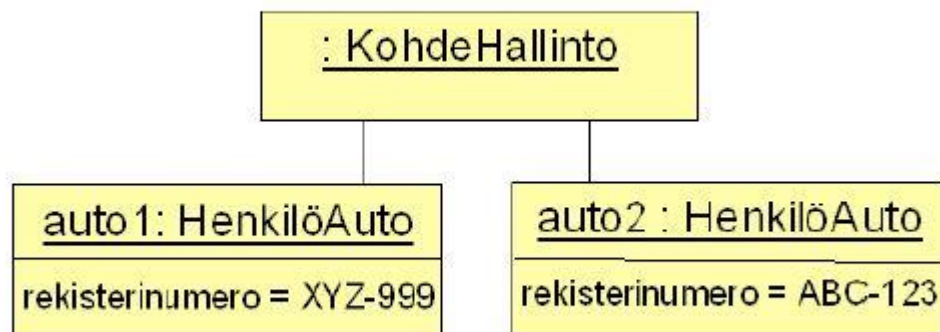
Luokkakaavio (Kuva 12) kuvaa järjestelmän luokkarakenteen ja luokkien väliset suhteet. (Luukkainen & Laine 2010.)



Kuva 12. Luokkakaavio (Häkkinen 2013)

Oliokaavio (Object diagram)

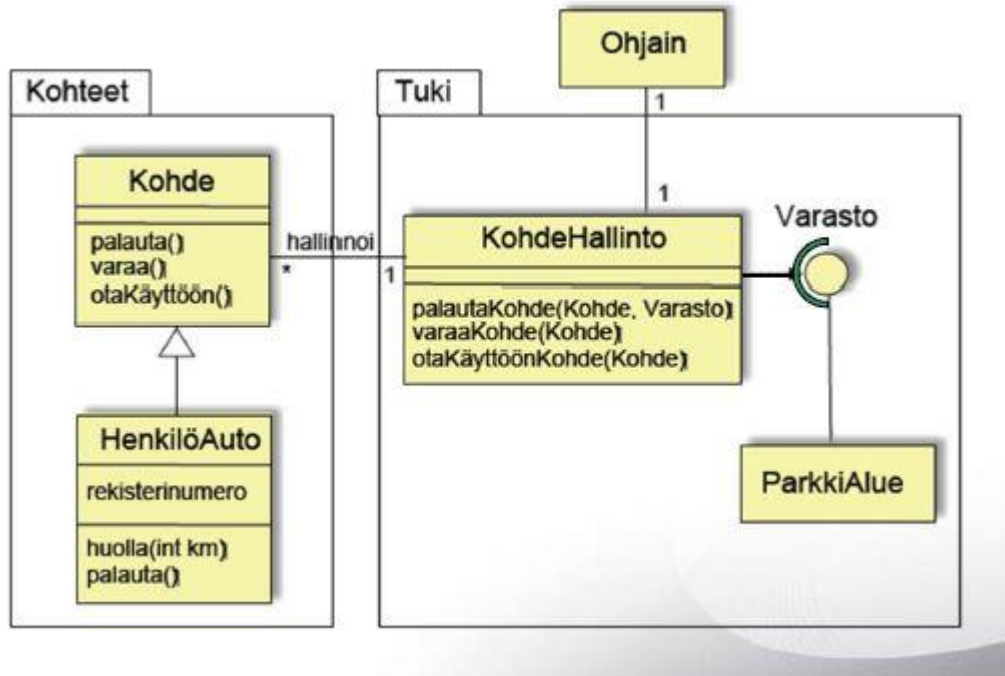
Oliokaavio (Kuva 13) kuvaa ohjelman luokkien instanssien konfiguraatiota tietyllä suoritushetkellä. (Luukkainen & Laine 2010.)



Kuva 13. Oliokaavio (Häkkinen 2013)

Pakkauskaavio (Package diagram)

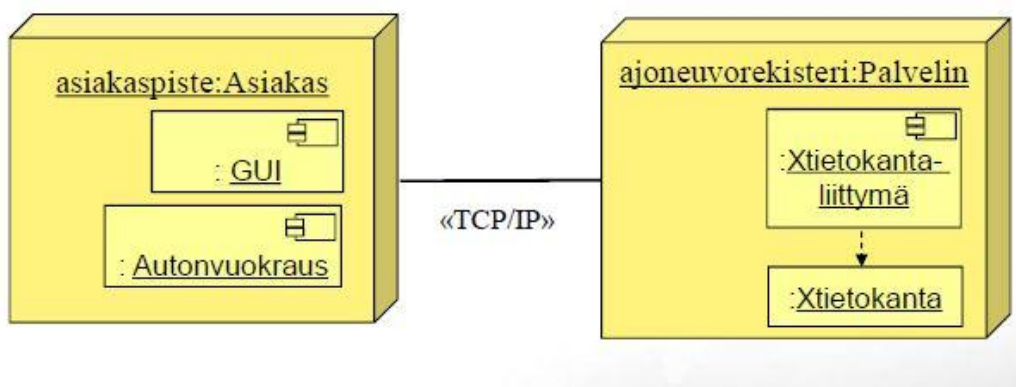
Pakkauskaavio (Kuva 14) ryhmittelee ohjelman rakenteen suurempiin kokonaisuuksiin. (Luukkainen & Laine 2010.)



Kuva 14. Pakkauskaavio (Häkkinen 2013)

Sijoittelukaavio (Deployment diagram)

Sijoittelukaavio (Kuva 15) kuvaa järjestelmän osien sijoittumisen fyysiseen laiteympäristöön ja komponenttien ajamisen kullakin laitteella. Kuvaa myös yhteyden tai kommunikointitavan laitteiden välillä. (Häkkinen 2013.)



Kuva 15. Sijoittelukaavio (Häkkinen 2013)

3 Tietokantaperustaisen verkkopalvelun toteutustekniikat ja välineet

Tässä kappaleessa käydään läpi verkkopalvelun toteutustekniikoita, toteutusvälineitä sekä aiheeseen liittyviä termejä.

3.1 Verkkopalvelu

Verkkopalvelulla tarkoitetaan Internet-verkkoon kytkettyä tietojärjestelmää, jonka kautta palvelua tarjotaan. Verkkopalvelulla voidaan tarkoittaa myös internetissä olevaa multimedia- tai sisältökokonaisuutta. Verkkopalvelu voidaan rinnastaa myös www- tai web-sivustoksi, jolloin voidaan sanoa, että verkkopalvelu on tietyn organisaation tuottama tai tiettyä aihetta käsittelevä, selkeän kokonaisuuden muodostama www-sivujen joukko. Esimerkiksi yritysten ja organisaatioiden www-sivuista voidaan käyttää nimitystä verkkopalvelu. Toisaalta esimerkiksi yrityksen sisäiset (intranet) ja ulkoiset (extranet) palvelut ovat myös verkkopalveluja. Verkkopalvelulle ominaisia piirteitä ovat verkko-osoite, sisällöntuotannosta vastaava taho, maailmanlaajuinen tai kohdennettu yleisö, reaaliaikainen saavutettavuus sekä mahdollisuus vuorovaikutteisuuteen. Verkkopalvelut voidaan jakaa karkeasti kahteen eri ryhmään viestinnällisiin- ja operatiivisiin verkkopalveluihin. (Wikipedia: Verkkopalvelu.)

Viestinnällisillä verkkopalveluilla tarkoitetaan tekstiin, kuviin, ääneen tai videokuvaan perustuvaa palvelua. Käyttäjän kannalta tällaisen palvelun hyöty pohjautuu informaatioon tai viihteellisyyteen. (Wikipedia: Verkkopalvelu.)

Operatiivisilla verkkopalveluilla tarkoitetaan, että järjestelmän tai palvelun kautta voidaan tehdä jokin muutos Internet-järjestelmän ulkopuolella. Esimerkkejä tällaisista järjestelmistä ovat verkkokaupat, verkkopankit ja sähköiset varausjärjestelmät. (Wikipedia: Verkkopalvelu.)

Perinteisten verkkopalvelujen rinnalle on viime vuosina tullut pilvipalvelut. SaaS (Software as a Service) eli ohjelmistoja palveluna, jossa asiakas maksaa kyseisen ohjelmiston käytöstä esimerkiksi kuukausimaksua. PaaS (Platform as a Service) eli sovellusalusta palveluna, jossa asiakas vuokraa sovellusalustan palve-

luntarjoajalta, johon käyttäjä asettaa oman ohjelmistonsa ajoin. IaaS (Infrastructure as A Service) eli infrastruktuuri palveluna, jossa asiakas hankkii palveluntarjoajalta virtuaalikoneita oman konesalin sijaan. IaaS-palveluun sisältyy yleensä verkkoyhteydet, tallennustila, palvelimet sekä ylläpito. (Finlander.)

3.2 Internet

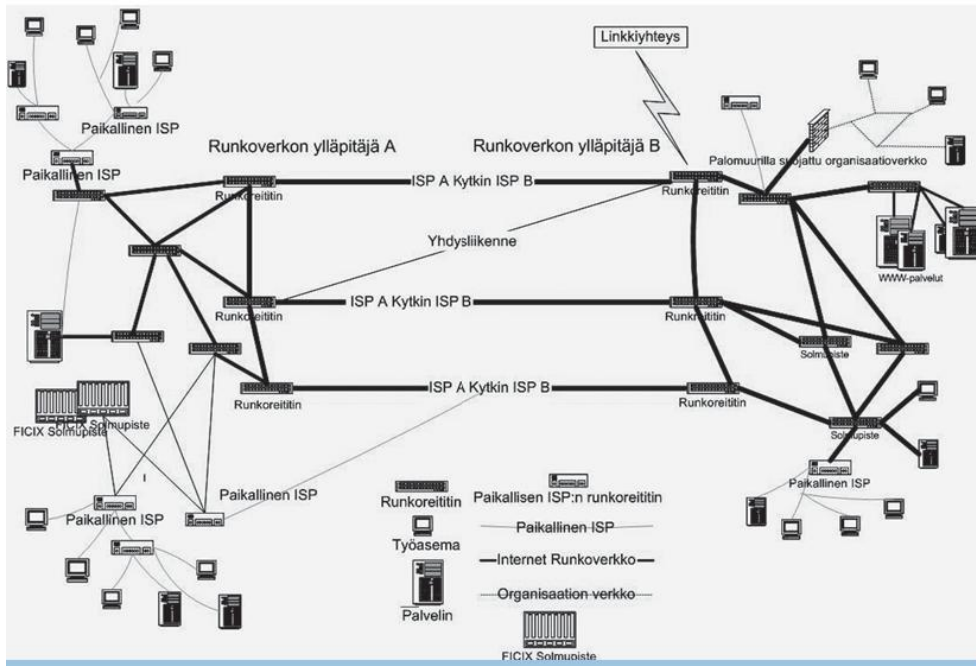
Internet on maailmanlaajuinen tietoverkko, jonka tehtävänä on yhdistää paikallisia tietoverkkoja toisiinsa. Internet on nimitys kuvaamaan yhteen liitettyjen tietoverkkojen kokonaisuutta. (Vahtiohje 2009. Internetin rakenne.)

Internet koostuu lukemattomista aliverkoista ja niiden aliverkoista, joita hallinnoivat tietoliikenneoperaattorit ja muut tietoliikennepalveluja tarjoavat tahot. Tietoliikenneyhteyksien tarjoajien verkkolaitteet muodostavat yhdessä internetin runkoverkon. (Vahtiohje 2009. Internetin rakenne.)

Standardeilla ja protokollilla pidetään huoli siitä, että kaikki verkkoon kytketyt koneet voivat olla yhteydessä keskenään. Koneiden välinen yhteys onnistuu internetin reitityslaitteiden kautta, jotka ohjaavat datan oikeaan osoitteeseen IP-osoitteen perusteella. (Kuntaliitto 2015. Internet Governance.) Reitittimet vastaanottavat tietoliikennepaketteja ja välittävät niitä eteenpäin omien reititystaulujensa perusteella edelleen seuraavalle reitittimelle kohti lopullista vastaanottajaa. (Vahtiohje 2009. Internetin rakenne.)

Internet on vikasietoinen verkko, joka sai alkunsa vuonna 1969 Yhdysvaltojen puolustushallinnon rahoittamana tutkimushankkeena nimeltä ARPAnet (Advanced Research Projects Agency). Hankkeessa pidettiin tärkeänä tietoliikenteen ja tiedonvälityksen toimivuutta, vaikka kaikki tietoliikennekomponentit eivät toimiskaan. (Vahtiohje 2009. Internetin rakenne.)

Kuva 16 on esitetty internetin fyysinen rakenne.



Kuva 16. Internetin fyysinen rakenne (Vahtiohje 2009. Internetin rakenne.)

Teknisesti internet-verkon toteutus jakautuu dataa siirtäviin protokollisiin ja niiden päällä oleviin sovellusprotokollisiin. Näiden kokonaisuutta kutsutaan TCP/IP-protokollaperheeksi. Alemman tason protokollat tarjoavat sovelluksia yhdistäviä TCP-yhteyksiä sekä sovellusten välisiä yhteydettömiä UDP-sähkeitä (User Datagram Protocol). Näiden päälle rakennetaan varsinaisia sovelluksia (Kuva 17).

OSI -MALLI	TCP/IP -PROTOKOLLAPERHE
7. Sovelluskerros (Application layer)	Sovelluskerros (Application layer) Telnet FTP SMTP SNMP HTTP jne.
6. Esityskerros (Presentation layer)	
5. Istuntokerros (Session layer)	
4. Kuljetuskerros (Transport layer)	Kuljetuskerros (Transport layer)
3. Verkkokerros (Network layer)	Internet -kerros (Internet layer)
2. Linkkikerros (Data link layer)	Verkkorajapintakerros (Network interface layer) ARP, RARP
1. Fyysinen kerros (Physical layer)	Fyysinen kerros

FYYSINEN SIIRTOALUSTA

Kuva 17. TCP/IP-protokollaperhe (Vahtiohje 2009. Internetin rakenne.)

Internet itsessään siirtää verkossa olevien koneiden välillä IP-tietoliikennepaketteja. IP-paketin alussa on otsikkokenttä, jossa on vastaanottajan ja lähettäjän numeeriset IP-osoitteet sekä muuta tietoa. Otsikkokentän jälkeen tulee dataa, käytännössä ylemmän tason protokollan otsikkokenttä. IP-paketin sisällä kulkevissa paketeissa kuljetetaan TCP- tai UDP-protokollan paketteja. TCP-protokolla toimii ylemmän tason protokollien kuljetustienä ja hoitaa muun muassa uudelleenlähettykset sekä eheysongelmien tarkistuksen. UDP-protokolla välittää yhteydettömiä tietosähkeitä sovellusten välillä. Se soveltuu yksinkertaisimpiin tarpeisiin, kuten DNS-verkkotunnusjärjestelmän tietokantakyselyihin. UDP-protokollaa käytetään myös esimerkiksi audio- ja videomateriaalin lähettämiseen verkossa. Paketit ovat sisäkkäisiä ja toimivat eri loogisilla tasoilla. Jokaisella tasolla pakettiin lisätään loogisen tason otsikkotietoja, jotka puretaan kerroksittain tarvittavalle tasolle asti reititettäessä pakettia kohdeosoitteeseen. (Vahtiohje 2009. Internetin rakenne.)

IP-protokolla välittää tietoliikennepaketteja koneiden välillä. Palvelimessa ja asiakas-koneessa voi kuitenkin olla useita samanaikaisia yhteyksiä avoinna. Tämän takia sovellukset eivät käytä IP-protokollaa suoraan, vaan TCP- ja UDP-protokollan kautta, jotka tuovat mukaan koneen sisäisen osoiteabstraktion, jota kutsutaan portiksi. Portti on abstrakti käsite eikä mikään fyysinen osa tietokoneessa. Tiettyjä porttinumeroita on sovittu käytettäväksi eri sovellusten välillä. Esimerkiksi web-palvelut varaavat portin numero 80. (Vahtiohje 2009. Internetin rakenne.)

DHCP (Dynamic Host Configuration Protocol) on verkkoprotokolla, jonka tehtävänä on jakaa IP-osoitteita uusille lähiverkkoon kytkeytyville laitteille. (Vahtiohje 2009. Internetin rakenne.)

3.3 Selain ja web-palvelin

Palvelimella tarkoitetaan tietoliikenteestä puhuttaessa tietokoneessa suoritettavaa ohjelmaa sekä tietokonetta, jossa ohjelmaa suoritetaan. Palvelimen tehtävänä on tarjota palveluja verkon välityksellä. (Wikipedia. Palvelin.)

Selaimesta on muodostunut nykyaikaisen tietojenkäsittelyn peruskäyttöliittymäohjelmisto, jonka ominaisuuksien ymmärtäminen ja hallinta on tärkeää, koska selain on usein myös tiedoston hallinnan tai tietojärjestelmien käyttöliittymä. Selaimen perustehtäviin kuuluu HTML-dokumenttien suorittaminen, jotta ne ovat kuvaruudulta luettavissa HTML-standardin määrittelemässä muodossa. Selaimen tehtäviin kuuluu myös CSS-tyylien tulkinta sekä JavaScript-kielen suoritus. Etenkin CSS-tyylisääntöjen tulkinta vaihtelee selainten välillä kohtalaisen paljon. (Vahtiohje 2009. Internetin rakenne.)

Suosituimpia selaimia työpöytäkäytössä ovat Internet Explorer (56,54%), Google Chrome (24,99%), Mozilla Firefox (11,89%) ja Apple Safari (5,00%). (Netmarketshare 2015.)

HTTP (Hypertext Transfer Protocol) on protokolla, jota selaimet ja palvelimet käyttävät tiedonsiirtoon. Protokolla perustuu siihen, että asiakas (esimerkiksi selain) avaa TCP-yhteyden palvelimelle ja lähettää pyynnön, johon palvelin vastaa lähettämällä selaimelle vastauksen, joka voi olla esimerkiksi web-sivu. (Wikipedia. HTTP). Selaimen ja palvelimen välinen kommunikointi voidaan toki suorittaa myös muulla protokollalla, kuten HTTPS-, FTP-, RTMP-protokollalla (Vahtiohje 2009. Internetin rakenne.)

3.4 Relaatietietokanta

Vuonna 1970 englantilainen tietojenkäsittelytieteen tutkija Edgar F. Codd esitteli relaatiomallin, joka oli sen aikaisista tietokantamalleista yksinkertaisin, joustavin sekä toteutti parhaiten tietokannalle asetettavat vaatimukset. (Henkilökohtaisen tiedonhallinnan perusteet 2004.)

Relaatietietokannoissa tiedot esitetään tauluina. Taulujen välille luodaan yhteyksiä, joita kutsutaan relaatioiksi. Taulun yhtä riviä kutsutaan tietueeksi. Taulun jokaisella rivillä eli tietueella on yhtä monta kenttää. Jokaisella tietueella tulee olla yksikäsitteinen perusavain, joka vastaa jotain reaali maailman kohdetta. Taulun kentistä voidaan viitata jonkin toisen taulun perusavaimeen tai toissijaiseen avaimen. Tällaista yhteyttä kutsutaan viite-eheydeksi ja viitattua kenttää kutsutaan viite-avaimeksi. Viite-eheys määrää, että jokaista viittaavassa taulussa

esiintyvää viiteavaimen arvoa kohti on oltava sama perusavaimen arvo viittauksen kohteena olevassa taulussa. Relaatiotietokannassa tietoa haetaan ainoastaan tiedon nimien ja arvojen perusteella eikä koskaan sijainnin tai järjestyksen mukaan. (Henkilökohtaisen tiedonhallinnan perusteet 2004.)

Relaatiotietokannassa tieto tallennetaan vain yhteen paikkaan tämä helpottaa ja nopeuttaa esimerkiksi tietokannan päivitystoimintoja, koska tietoa tarvitsee muuttaa vain yhdessä taulussa. (Henkilökohtaisen tiedonhallinnan perusteet 2004.)

3.5 Web-hotelli

Web-hotelli on palvelu, jossa asiakas vuokraa palveluntarjoajalta kiintolevytilaa omia web-sivuja sekä muita verkkopalveluja varten. Palveluihin kuuluu yleensä myös tuki tietokannoille ja eri ohjelmointikielille sekä sähköpostiosoitteille. Web-hotelleihin saa yleensä liitettyä oman verkkotunnuksen. Käyttöjärjestelmänä web-hotelleissa on yleensä Linux, mutta myös Windows-pohjaisia palvelimia on saatavilla. Web-hotellin valintaan vaikuttaa kehitystyössä käytetty ohjelmointikieli sekä mahdollinen tietokanta. (Wikipedia. Webhotelli.)

3.6 PHP

PHP (Hypertext Preprocessor) oli alun perin kokoelma web-pohjaisten sovellusten tekemistä helpottavia rutiineja, eräänlainen palvelimella ajettavien CGI-ohjelmien tekemistä helpottava komentokokoelma. PHP on tulkittava kieli, joka tarkoittaa, että PHP-koodi ajetaan palvelimella ennen sivun lähettämistä selaimelle. Näin PHP-koodi ei näy selaimelle vaan ainoastaan koodin lopullinen tulos. Ratkaisu on tietoturvan kannalta hyvä. Toinen etu on, että selaimelta ei vaadita mitään erityisominaisuuksia, koska palvelin palauttaa aina HTML-muotoisen dokumentin. PHP-koodi kirjoitetaan aloitus- ja lopetustagien sisään ja se voidaan kirjoittaa suoraan HTML-koodin sekaan. (Heinisuo & Rauta 2007, s. 12–13.)

3.7 HTML

HTML (Hypertext Markup Language) on avoimesti standardoitu web-sivujen rakenteen kuvaava kuvauskieli. HTML-kielen ensimmäinen kuvaus julkaistiin vuoden 1991 lopulla. Uusimman version eli HTML5:sen kehitys aloitettiin vuonna

2004 ja ensimmäinen vedos julkaistiin vuoden 2008 tammikuussa. HTML5-standardi valmistui lokakuussa 2014. (Wikipedia. HTML.)

HTML5-sovelluksista puhuttaessa tarkoitetaan yleisesti sovellusta, joka on tehty toimimaan selainmoottorin perustalla ja joka käyttää moderneja ja avoimia web-tekniikoita kuten HTML5, CSS3 ja JavaScript. (Lehdonvirta & Korpela 2012.)

3.8 JavaScript

JavaScript on ohjelmointikieli, jota käytetään web-sovellusten kehityksessä palvelimen tai selaimen ohjelmointiin. Tässä dokumentissa tarkoitetaan selaimessa suoritettavaa koodia. (Sand & Hyrskykari 2015.)

JavaScript-kielen kehitti alun perin Netscape omiin selaimiinsa. Vuonna 1995 kieli oli käytössä Netscapen selaimissa nimellä LiveScript, mutta Netscape vaihtoi nimen JavaScriptiksi. Microsoft otti kielen mukaan omaan Internet Explorer 3.0 selaimensa, mutta lisäsi siihen omia piirteitään ja kutsui kieltä nimellä Script. Nykyisen menestyksen taustalla on ECMA, joka otti vastuun kielen standardoinnista. (Sand & Hyrskykari 2015.)

JavaScriptiä pidetään yleisesti kevyenä ohjelmointikielenä, jonka syntaksi on helppo omaksua ja oppia. JavaScript mahdollistaa muun muassa DOMin (Document Object Model) manipuloimisen, lomakkeiden käsittelyn ja tietojen tarkistamisen ilman palvelimen apua. Esimerkiksi tietokantakyselyt voidaan suorittaa AJAX-tekniikalla ilman sivun uudelleen latausta. Tässä tekniikassa palvelimelle lähetään pyyntö AJAX-tekniikalla ja palvelin palauttaa esimerkiksi JSON-dataa takaisin, joka näytetään selaimessa JavaScriptin avulla. (Sand & Hyrskykari 2015.)

JavaScriptille on saatavilla useita kirjastoja, jotka helpottavat ja yksinkertaistavat koodin kirjoittamista. Suosituin näistä kirjastoista on jQuery.

Tässä työssä on käytetty jQuery ja jQuery UI -kirjastoja, jotka ovat avoimen lähdekoodin JavaScript-kirjastoja.

3.9 CSS

CSS (Cascading Style Sheets) on yksinkertainen tyylikieli, jonka avulla määritellään HTML-dokumentin ulkoasu ja esitystapa. CSS-kielen tarkoituksena on esitystavan erottaminen varsinaisen dokumentin sisällöstä ja rakenteesta. (2KMediat. CSS-opas)

CSS:n kehitys alkoi 1990-luvun puolivälissä kun huomattiin, ettei HTML-kieli enää vastannut sille asetettuja vaatimuksia. Syynä tähän oli selainsota Netscape Navigatorin ja Microsoft Internet Explorerin välillä. Molemmat selainvalmistajat lisäsivät HTML-kieleen omia selainkohtaisia piirteitä ja ominaisuuksia, joiden avulla muutettiin dokumentin ulkoasua. Tämä johti tilanteeseen, jossa HTML-kieli oli jakautunut kahteen eri selaimelle tarkoitettuun versioon. (2KMediat. CSS-opas.)

Yhtenä CSS-kielen tavoitteista oli yhtenäisyys – selaimesta, käyttöjärjestelmästä ja mediasta riippumaton tapa määritellä web-dokumentin ulkoasu. Tätä tavoitetta ei ole saavutettu kokonaisuudessaan, koska edelleen eri selaimet käsittelevät tyyliä eri tavalla. (2KMediat. CSS-opas.)

4 Käytetyt kehitysvälineet

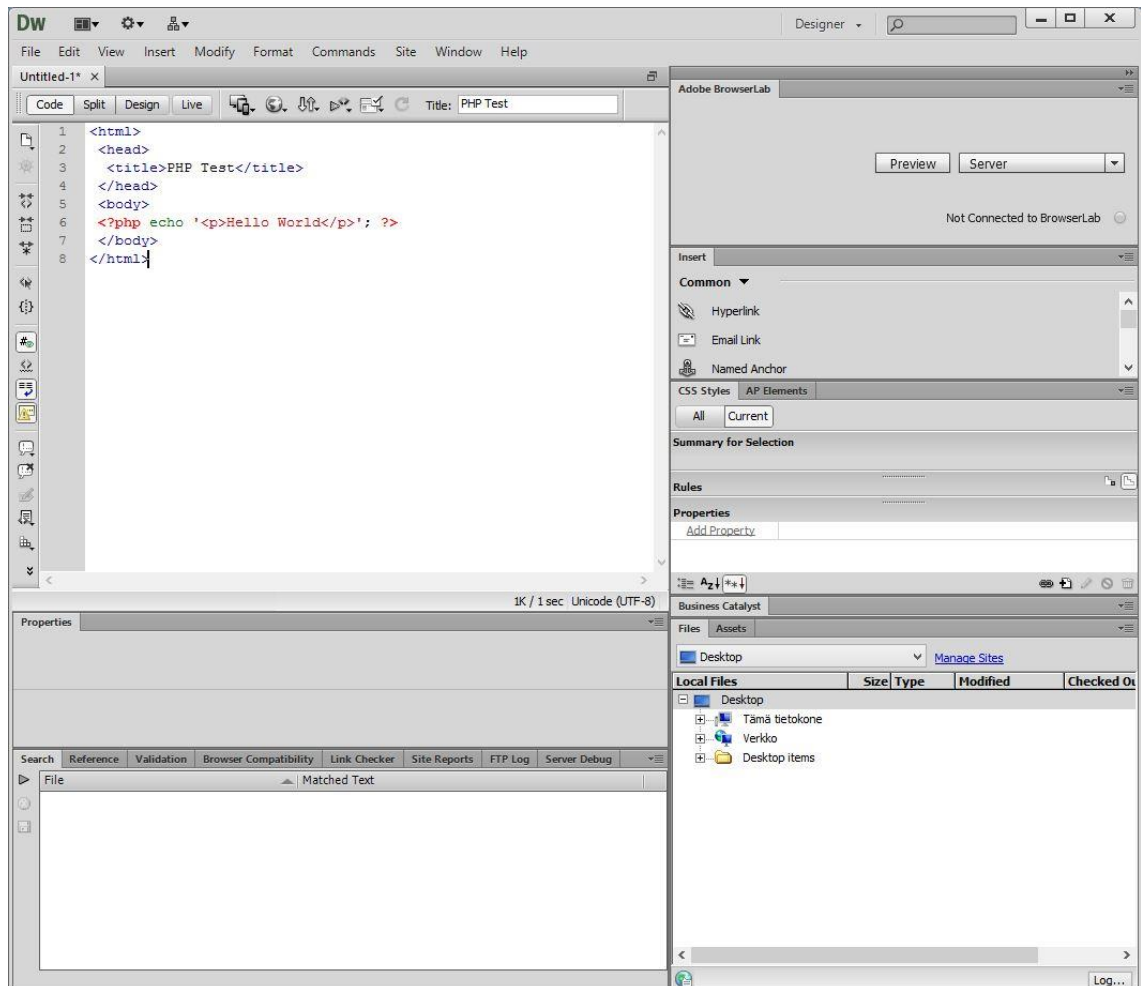
Tässä luvussa käydään läpi järjestelmän kehityksessä käytettyjä välineitä ja ohjelmia. Käyttöjärjestelmänä kehitysympäristössä toimi Microsoft Windows 8.1.

4.1 Adobe Dreamweaver

Adobe Dreamweaver on Adobe Systemsin kehittämä web-sivujen tekemiseen tarkoitettu WYSIWYG-editori (Kuva 18). Dreamweaveriä kehitti vuoteen 2005 asti Macromedia, jolloin Macromedia muuttui osaksi Adobe Systemsiä. (Wikipedia: Adobe Dreamweaver.)

Dreamweaver käyttää tiedostojen siirtoon kehitysympäristön ja palvelimen välillä FTP-, SFTP- ja webDAV –protokollia. Dreamweaver tukee muun muassa CSS3- ja HTML5–tekniikoita, syntaksin korotusta, koodilohkon piilottamista sekä eri nä-

kymiä, joita ovat koodinäkö, suunnittelunäkö, jaettu-näkö, joka on koodinäkö ja suunnittelunäkö välimuoto sekä Live-näkymä. Tuettuja ohjelmointikieliä ovat ActionScript, ColdFusion, C#, HTML(5) Java, JavaScript, PHP, VBScript, Visual Basic ja XML. (Wikipedia: Adobe Dreamweaver.)



Kuva 18. Adobe Dreamweaver CS 6 käyttöliittymä

Dreamweaver sisältää myös kätevän Synchronize-toiminnon, jonka avulla tiedostot on mahdollista synkronoida kehitysympäristön ja palvelimen välillä.

Tässä projektissa oli käytössä Adobe Dreamweaver CS6 -versio.

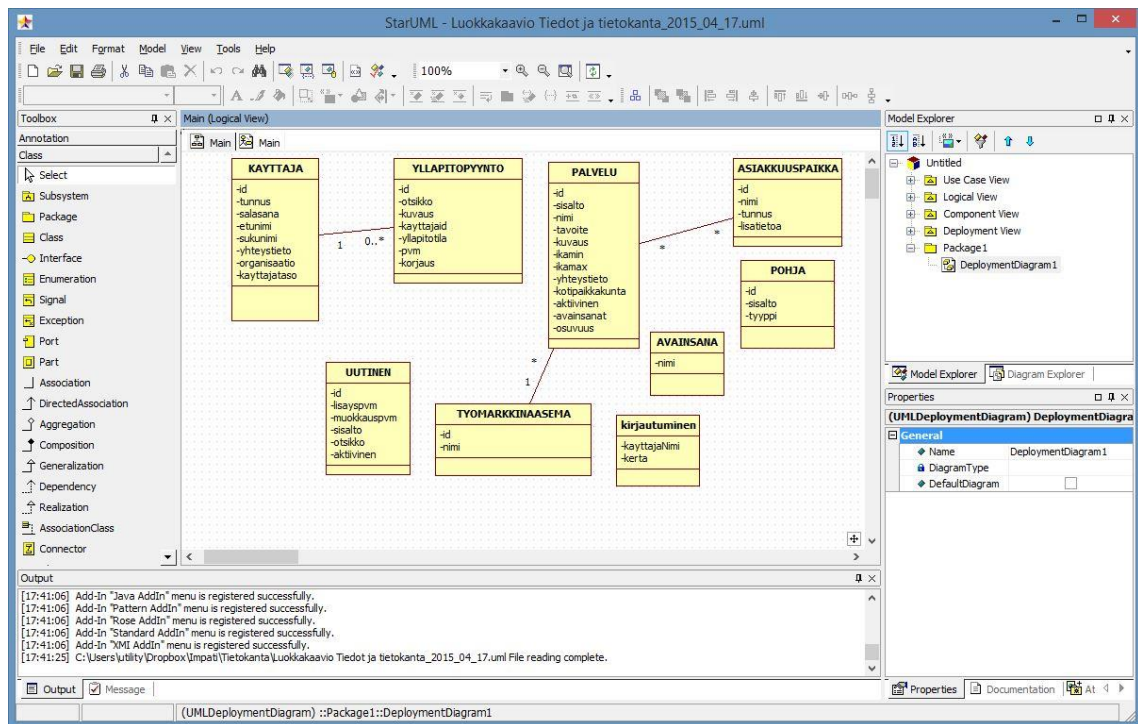
4.2 StarUML

StarUML on avoimen lähdekoodin UML-mallinnustyökalu, StarUML on käyttöliittymältään selkeä ja helppokäyttöinen (Kuva 19Kuva 18). StarUML versio 1 tukee

seuraavia UML 2:sen kaaviotyyppejä: luokkakaavio, käyttötapauskaavio, sekvenssikaavio, yhteistoimintakaavio, aktiveettikaavio, tilakaavio, komponenttikaavio, käyttöönottokaavio ja koostekaavio. (StarUML. User guide).

Ohjelmasta on olemassa myös versio 2, joka tukee kaikki UML 2:ssa olevia kaaviotyyppejä. Versio 2 ei ole avointa lähdekoodia (StarUML).

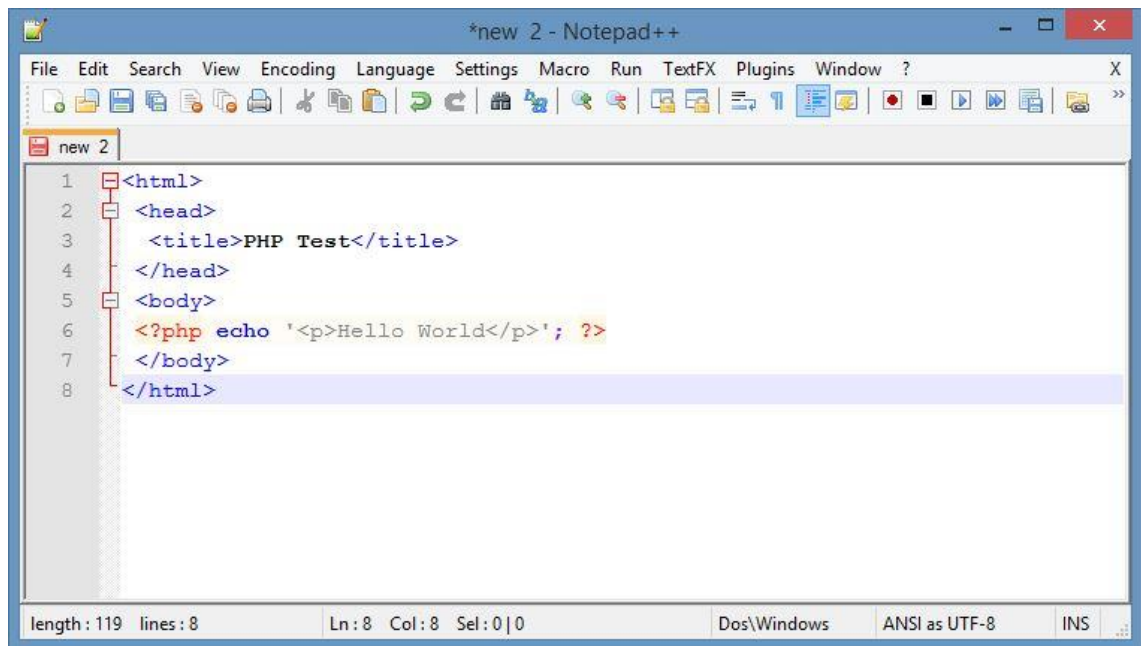
Tässä projektissa käytettiin ohjelman versiota 1.



Kuva 19. StarUML käyttöliittymä (versio 1)

4.3 Notepad++

Notepad++ on avoimeen lähdekoodiin perustuva monipuolinen ja kevyt tekstieditori Windows-alustalle. Notepad++ perustuu tehokkaaseen Scintilla-editorikomponenttiin. Notepad++ on kehitetty käyttäen C++-ohjelmointikieltä. Notepad++ tukee ohjelmoinnissa hyödyllistä syntaksin korotusta useilla eri ohjelmointikielillä, useiden tiedostojen avaamista välilehtiin sekä koodilohkon piilotusta (Kuva 20). Ohjelmaan on saatavilla useita liitännäisiä tekstin muokkaamisen ja ohjelmoinnin helpottamiseksi. Tuettuja ohjelmointikieliä on tällä hetkellä yli 50, joista yleisimpiä ovat C, C++, C#, COBOL, CoffeeScript, CSS, HTML, Java, JavaScript, Objective-C, Pascal, Perl, PHP, Python, Ruby ja Visual Basic. (Wikipedia: Notepad++.)

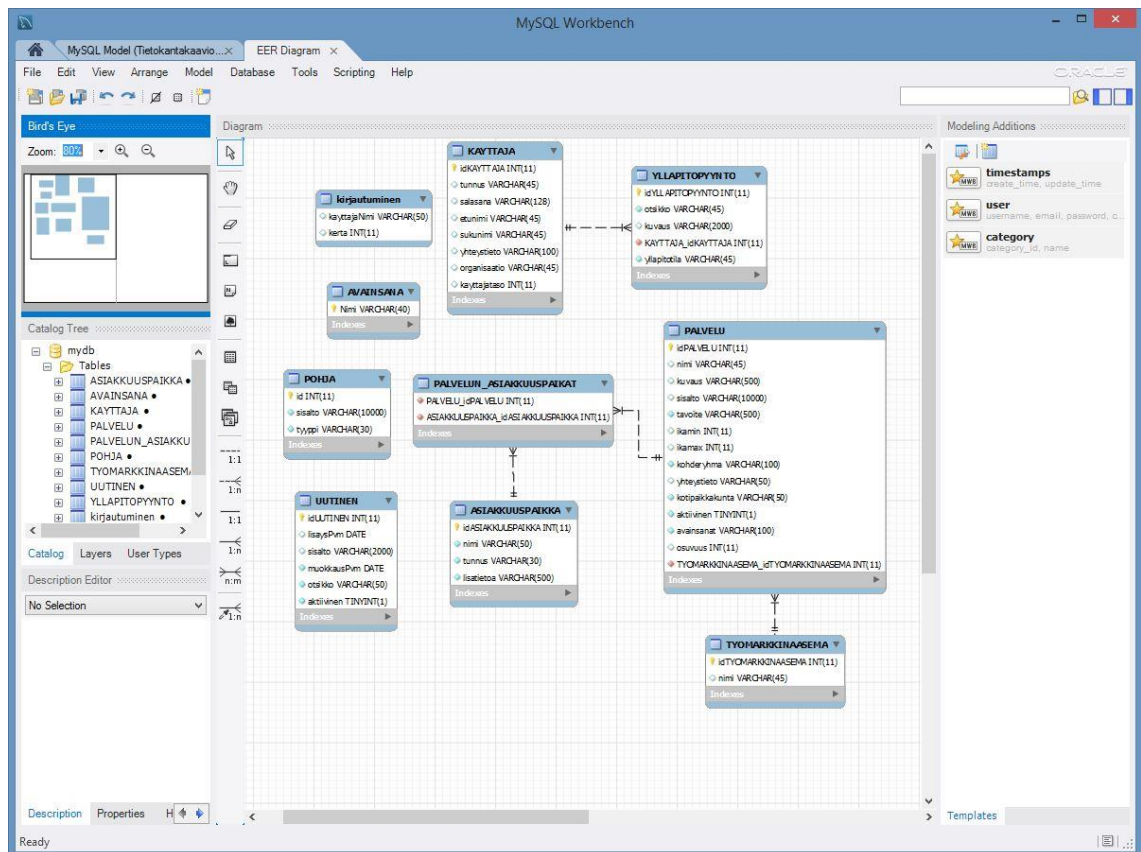


Kuva 20. Notepad++ käyttöliittymä

Tässä projektissa Notepad++-ohjelmaa käytettiin PHP-tiedostojen muokkaamiseen Adobe Dreamweaverin rinnalla.

4.4 MySQL Workbench

MySQL Workbench on visuaalinen tietokantojen suunnittelu-, mallinnus- ja hallintatyökalu MySQL-tietokannoille. Sen avulla on mahdollista esimerkiksi luoda tietokanta ER-kaaviosta kirjoittamatta SQL-kieltä tai luoda ER-kaavio olemassa olevasta tietokannasta (Kuva 21). Ohjelmasta on olemassa ilmaiset ja kaupalliset versiot. Ilmaisversiosta on riisuttu joitain yrityksille suunnattuja edistyneempiä ominaisuuksia. MySQL Workbench on saatavilla Linux-, OS X- ja Windows-alustoille. (MySQL Workbench.)

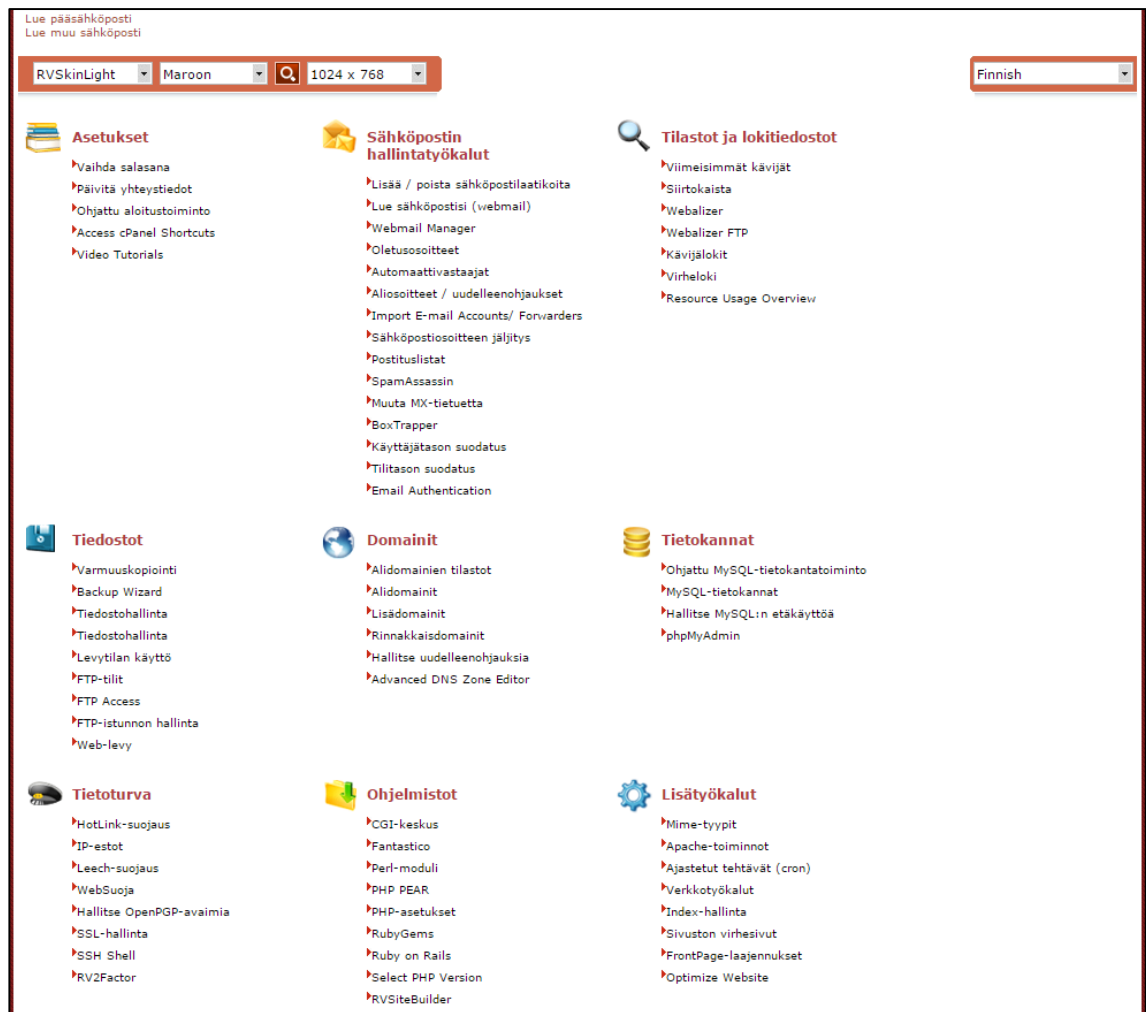


Kuva 21. MySQL Workbench käyttöliittymä

4.5 cPanel

cPanel on Linux-pohjainen web-hotellin ylläpitämiseen ja konfigurointiin käytetty järjestelmä, jota käytetään selaimen kautta.

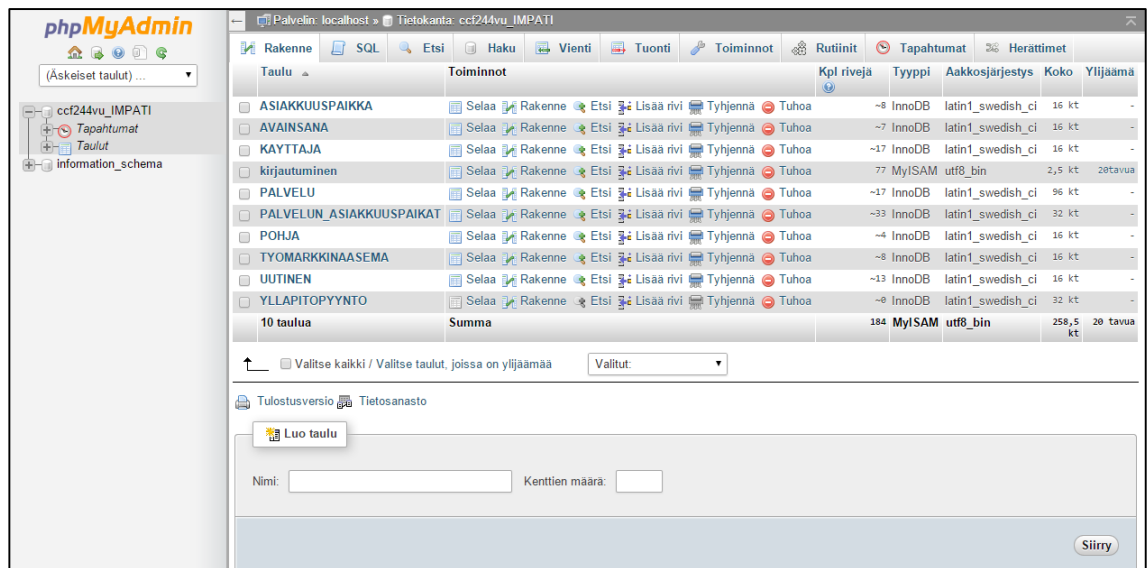
cPanelin avulla voidaan helposti hallita web-sivuihin liittyviä toimintoja, kuten tietokantoja, tiedostoja, sähköpostiosoitteita, domaineja, kävijätilastoja sekä tietoturva-asetuksia (Kuva 22). Ominaisuuksia voi laajentaa cPanel-järjestelmän kautta asennettavilla lisäosilla.



Kuva 22. cPanel käyttöliittymä

4.6 phpMyAdmin

phpMyAdmin on selainpohjainen PHP-ohjelmointikielellä toteutettu MySQL-tietokantojen hallintatyökalu. SQL-kyselyiden suorituksen lisäksi phpMyAdmin sisältää valmiit toiminnot useiden eri toimenpiteiden suorittamiseen. Tällaisia toimintoja ovat esimerkiksi uuden taulun luonti, taulun poistaminen tai tyhjentäminen, uuden tietokannan luonti sekä tietokannan rakenteen ja sisällön tutkiminen (Kuva 23). phpMyAdmin on esiasennettuna useissa web-hotelleissa. (Laaksonen 2009.)



Kuva 23. phpMyAdmin käyttöliittymä

5 IMPATI-projekti

Tässä luvussa käydään läpi IMPATI-projektin vaiheet vesiputousmallin mukaisesti suunnittelu, organisaatio, vaiheet, toiminnallinen määrittely, arkkitehtuurin suunnittelu, toteutus, testaus sekä käyttöönotto.

5.1 Esitutkimus ja projektisuunnittelu

Esitutkimuksen ja projektisuunnitelman projektiin oli alun perin tehnyt Sami Repo omassa opinnäytetyössään. Esitutkimuksessa perusteltiin järjestelmän hyödyllisyyttä asiakkaalle sekä muille mahdollisille asiakasorganisaatioille. Esitutkimuksessa selvisi myös osittain järjestelmän vaatimukset, jotka muodostuivat asiakkaan tarpeiden ja resurssien mukaisesti.

5.1.1 Organisaatio

Projektioorganisaatioon on kuulunut kehityksen aikana useita henkilöitä, joista osa on ollut mukana alusta asti ja osa taas on liittynyt siihen kesken projektin ja poistunut siitä erinäisistä syistä.

Projektiryhmä ja ohjausryhmä pitivät yhteisiä palaverreja. Juha Rasimus on varsinaisen hankkeen projektipäällikkönä ja Teppo Pötry toiminut oman opinnäytetyönsä projektipäällikkönä.

Projektin organisaatio koostuu projektiryhmästä, projektin ohjausryhmästä, asiakkaasta sekä loppuasiakkaista (Kuva 24).



Kuva 24. IMPATI-projektin organisaatiokaavio

Seuraavassa on projektiryhmän jäsenet sekä heidän vastuualueensa:

Juha Rasimus, Kehittämishankkeen projektipäällikkö, ohjausryhmän puheenjohtaja

Juha Rasimus vastaa projektiryhmän johtamisesta sekä IntoPajat ry:n työntekijänä projektin tiedottamisesta IntoPajat ry:n johdolle. Järjestelmän käyttäjän vaatimukset, sekä vaatimusten toteutuminen ovat Juha Rasimuksen vastuulla. Juha Rasimus hyväksyy kaikki järjestelmän lisätyt tai siitä poistuvat ominaisuudet ja toiminnot.

Martti Ylä-Jussila, ohjaaja

Martti Ylä-Jussila toimii projektissa Sami Revon ja Teppo Pötryn opinnäytetyön ohjaajana sekä Saimaan ammattikorkeakoulun edustajana yhteyshenkilönä IntoPajat ry:n ja Saimaan ammattikorkeakoulun opiskelijoiden välillä. Martti Ylä-Jussilan vastuualueena on myös opiskelijoiden ohjaus sekä neuvonta.

Teppo Pötry, ohjelmistosuunnittelija ja projektipäällikkö

Teppo Pötryn vastuualueena projektissa oli toiminnallinen määrittely, projektin tekninen suunnittelu, johon kuuluvat moduulisuunnittelu ja tietokantasuunnittelu. Tepon vastuualueisiin kuului myös ohjeen tuottaminen järjestelmän ylläpitäjälle, sekä töiden jakaminen projektissa eri henkilöille.

Elina Hakulinen, ohjelmistosuunnittelija

Elina Hakulinen tuli mukaan projektiin keväällä 2013 ja hänen vastuualueensa oli Teppo Pötryn avustaminen moduulien toteutuksessa sekä toiminnallinen määrittely. Elina Hakulinen vastaa järjestelmässä olevien ylläpitopyyntöjen toteutuksesta ja suunnittelusta.

Tuomas Porvali, ohjelmistosuunnittelija

Tuomas Porvali tuli mukaan projektiin syksyllä 2013 ja hänen vastuualueensa oli Teppo Pötryn avustaminen moduulien toteutuksessa. Tuomas Porvali vastaa järjestelmässä olevien palveluhallinnan, työmarkkina-asemien hallinnan sekä asiakkuuspaikkojen suunnittelusta ja toteutuksesta.

Tomi Kiiveri, graafikko ja testaaja

Tomi Kiiveri tuli mukaan projektiin kesällä 2014, jolloin hänen vastuualueekseen määritettiin sivuston graafinen ilme ja ulkoasu. Myöhemmin Tomi Kiiveri osallistui myös järjestelmän testaukseen.

Sami Repo, ohjelmistosuunnittelija ja projektipäällikkö

Sami Revon vastuualueena projektissa oli alkuperäisen toiminnallisen määrittelyn tekeminen, projektin tekninen suunnittelu, johon kuuluvat arkkitehtuurin suunnittelu, moduulisuunnittelu ja tietokantasuunnittelu. Sami Repo poistui projektista kesällä 2012.

Janne Raunila, ohjelmoija

Janne Raunilan toimialue projektissa oli Sami Revon ja Teppo Pötryn avustaminen kehitystyössä, joihin sisältyi pienien moduulien tekoa järjestelmään. Janne Raunila poistui projektista keväällä 2013.

Kari Neuvonen, ohjelmoija

Kari Neuvosen toimialue projektissa oli Sami Revon ja Teppo Pötryn avustaminen kehitystyössä, joihin sisältyi pienien moduulien tekoa järjestelmään. Kari Neuvonen poistui projektista vuonna 2012.

Jari Vilksa, testaaja

Jari Vilksa tuli mukaan projektiin keväällä 2014, jolloin hänen vastuualueisiin kuuluivat toiminnallisen määrittelyn katselmointi sekä pöytäkirjojen kirjoittaminen viikkopalaverien aikana. Myöhemmin Jari Vilksa vastasi myös testitapausten suunnittelusta sekä itse testaamisesta ennen poistumistaan projektista talvella 2015.

Jukka Ilvonen, tietoturva

Jukka Ilvosen vastuualueena projektissa oli järjestelmän tietoturva ja sen suunnittelu, sekä alkuvaiheessa Teppo Pötryn avustaminen palvelimen konfiguroinnissa. Jukka Ilvonen poistui projektista talvella 2013.

Sami Pulkkinen, graafikko

Sami Pulkkinen oli mukana projektissa muutaman viikon ajan vuonna 2014. Sami Pulkkinen vastuualueisiin kuului logon suunnitteluja ja toteutus.

Ritva Virolainen, sihteeri

Ritva Virolainen vastasi projektin alkuvaiheessa kokouspöytäkirjojen ylläpidosta, projektisuunnittelusta sekä alustavasta testaussuunnittelusta. Ritva Virolainen poistui projektista vuonna 2012.

Vesa Vaittinen, graafikko

Vesa Vaittinen vastasi projektissa ollessaan sivuston ulkoasusta sekä graafisesta ilmeestä. Vesa Vaittinen poistui projektista vuonna 2013.

Mika Veikkolainen, testaaja

Mika Veikkolaisen vastuualueisiin projektissa kuului järjestelmän testaus sekä pöytäkirjojen ylläpitäminen viikkopalavereissa.

Ilkka Tuomainen, testaaja

Ilkka Tuomaisen vastuualueisiin projektissa kuului järjestelmän testaus.

Mika Paakki, testaaja

Mika Paakin vastuualueisiin projektissa kuului järjestelmän testaus.

5.1.2 Toimintatapa

Ohjausryhmä piti palavereja noin kahden viikon välein, jossa seurattiin projektin edistymistä sekä täsmennettiin tarvittaessa järjestelmän toimintaa sekä päätettiin poistuvista ja lisättävistä ominaisuuksista. Syksyyn 2014 asti palaverit pidettiin

IntoPajat ry:n tiloissa Imatran Honkaharjussa. Tämän jälkeen siirryttiin etäpalaveriin ja aloitettiin käyttämään Adobe Connect Pro –verkkokokousohjelmaa matkustelun vähentämisen ja toiminnan tehostamisen vuoksi. Etäpalaverien etuna oli, että jokainen osallistuja pystyi osallistumaan palaveriin mistä tahansa verkkoyhteyden kautta. Projektille luotiin Dropbox-palveluun oma kansio, jossa säilytettiin kaikki projektiin liittyvät dokumentit sekä muut tiedostot. Kaikilla projektiryhmän jäsenillä oli oikeus käsitellä dokumentteja.

Projektisuunnitelmassa määriteltiin projektin alkuperäinen aikataulu sekä projektin organisaatio. Projektin aikataulu ei pitänyt paikkaansa ollenkaan, joskin se oli alun perinkin tehty melko optimistiseksi resursseihin nähden. Projektisuunnitelmassa määritettiin myös projektin päättymisehdot ja tavoitteet, alkuperäinen projektiorganisaatio, resurssit, sekä menetelmät ja työkalut. Oikeudet lähdekoodiin ja tietokantaan jäivät asiakkaalle. Projektiryhmän tuottamat dokumentaatio on julkista ja sen oikeudet jäivät asiakkaalle.

5.2 Toiminnallinen määrittely

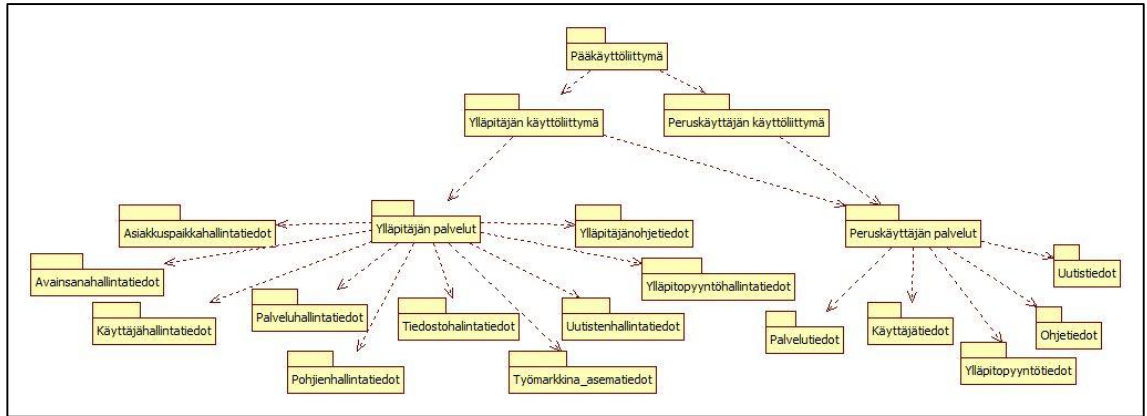
Toiminnallinen määrittely sisältää kaiken projektissa tehdyn järjestelmää koskevan määrittelyn. Määrittelydokumenttia pyrittiin päivittämään aina kun projektiryhmä päätti lisätä, poistaa tai muuttaa järjestelmässä olevia toimintoja ja vaatimuksia. Toiminnallisen määrittelyn tuottamiseen ja katselmointiin osallistui useita henkilöitä projektiryhmässä.

5.3 Suunnittelu

Järjestelmän suunnitteluvaihe jaettiin kahteen osaan: arkkitehtuurisuunnittelu ja moduulisuunnittelu.

5.3.1 Arkkitehtuurisuunnittelu

Kuvassa 25 on esitetty IMPATI-järjestelmän kerrosarkkitehtuuri. Ylimpänä on eri käyttäjätyyppien käyttöliittymä pakkaukset, niiden alapuolella on toiminto- ja palveluryhmien pakkaukset ja alimpana tietoryhmien pakkaukset (Kuva 25).



Kuva 25. Arkkitehtuurikaavio

5.3.2 Moduulisuunnittelu

Moduulisuunnittelusta vastasi jokainen projektin kehitystyössä mukana oleva oman vastualueensa mukaisesti. Moduulien suunnittelusta oli kuitenkin sovittu yhdessä pääpiirteet, joiden mukaan toimittiin. Esimerkiksi käyttöliittymä ja tietokantoihin liittyvät kyselyt ja toiminnot oli pyrittävä eristämään toisistaan mahdollisuuksien mukaan ylläpidon parantamiseksi. Käyttöliittymiä suunniteltiin projektiryhmän kanssa viikkopalavereissa.

Tietokannasta oli olemassa aiempi versio, jota kuitenkin jouduttiin muuttamaan niin paljon, että palavereissa käytyjen suunnittelujen pohjalta se päätettiin luoda uudelleen alusta. Lopullinen tietokanta on esitetty Kuva 27.

5.4 Toteutus

Järjestelmä toteutettiin iteratiivisesti yksi käyttötapaus kerrallaan. Tullessani mukaan projektiin toteutettuna oli osittain sisäänkirjautuminen ja uloskirjautuminen sekä hahmotelma järjestelmän hakutoiminnosta. Katsoin kuitenkin parhaaksi toteuttaa sisäänkirjautumisen kokonaan uudestaan käyttäen hiukan eri tekniikkaa kuin millä se oli alun perin tehty. Alussa tietokannan rakennetta jouduttiin myös muuttamaan. Tämän jälkeen järjestelmään toteutettiin ylläpitäjän toimintoja, koska niiden avulla tuotetaan kaikki sisältö järjestelmään. Kun ylläpitäjän toiminnot oli saatu suurin piirtein valmiiksi, aloitettiin peruskäyttäjien käyttötapausten toteutus.

Toteutuksessa tehtiin ensin toiminnallisen määrittelyn mukaisia karkeita prototyyppisiä toimintoja, jotta asiakkaalle pystyttiin näyttämään järjestelmän toimintaa käytännössä sekä kuuntelemaan heidän mielipiteitään ja ottamaan huomioon muutostarpeet. Tämän jälkeen toimintaa korjattiin vastaamaan asiakkaan vaatimuksia.

Järjestelmää kehitettiin paikallisesti, jonka jälkeen tiedostot ladattiin ftp-yhteyden kautta kehityspalvelimelle testattavaksi. Tällä tavoin asiakkaan oli mahdollista tutustua järjestelmän toimintaan koko kehityksen ajan.

5.5 Testaus

Jokainen järjestelmän ohjelmoija suoritti oman vastualueensa yksikkötestauksen. Tämän lisäksi jokaisesta järjestelmän toiminnosta laadittiin testitapauksia, jotka sisällytettiin testisuunnitelmaan. Jokaista testitapausta kohti kirjattiin testausraporttiin oliko toiminta virheetöntä, virheellistä, osittain virheellistä sekä testaajan omat huomiot testitapauksesta ja testitapauksen toiminnasta. Tällä tavalla kehittäjiä oli helppo seurata testauksen edistymistä sekä korjata mahdolliset virheet ja puutteet. Järjestelmän jokainen testitapaus on testattu Microsoft Internet Explorer-, Google Chrome- sekä Mozilla Firefox-selaimilla. Järjestelmän testauksesta vastasi asiakkaan tiloissa toimiva ATK-paja sekä Saimaan ammattikorkeakoulun opiskelija Tomi Kiiveri.

Testaus aloitettiin lokakuussa 2014 ja jatkuu tätä dokumenttia kirjoittaessa yhä.

5.6 Käyttöönotto

Ennen käyttöönottoa asiakas on järjestänyt järjestelmän toiminnoista ja käytöstä muutamia esittelytilaisuuksia. Ennen varsinaista käyttöönottoa on tarkoitus pitää asiakkaan toimesta perehdytys ja koulutustilaisuus järjestelmän käytöstä. Järjestelmä on suunniteltu käyttöönotettavaksi kevään 2015 aikana.

6 Lopputuotteen esittely

Tässä luvussa käydään läpi valmiin lopputuotteen ulkoasu ja toiminnot. IMPATI-järjestelmä on itsenäisesti toimiva ja yleiseltä käytöltä suojattu järjestelmä. Järjestelmän on tarkoitus toimia palvelunohjauksen apuvälineenä ja se on suunniteltu helpottamaan ja tehostamaan palvelunohjausprosessia. Järjestelmää käytetään internetin kautta selaimella. Tuettuja selaimia ovat Microsoft Internet Explorer, Google Chrome ja Mozilla Firefox. Sivuston toiminta on riippuvainen JavaScriptin suorituksesta, joten sen on oltava sallittuna selaimen asetuksista.

6.1 Yleiskuvaus

Järjestelmässä on kahdenlaisia käyttäjärooleja: ylläpitäjä ja peruskäyttäjä.

Peruskäyttäjällä on sisäänkirjautumisen jälkeen mahdollisuus listata ja lukea uutiset, listata palvelut, hakea palveluita, tarkastella yksittäisen palvelun tietoja, lähettää ylläpitopyyntö ylläpitäjälle sekä vaihtaa omaa salasanaa tai yhteystietoa.

Ylläpitäjällä on sisäänkirjautumisen jälkeen mahdollisuus hallinnoida järjestelmässä olevia palveluja, asiakkuuspaikkoja, avainsanoja, työmarkkina-asemia, ylläpitopyyntöjä, käyttäjiä, uutisia, tiedostoja sekä muokata palvelun ja uutisen lisäämisessä käytettäviä mallipohjia. Hänellä on myös mahdollisuus suorittaa peruskäyttäjän toimintoja.

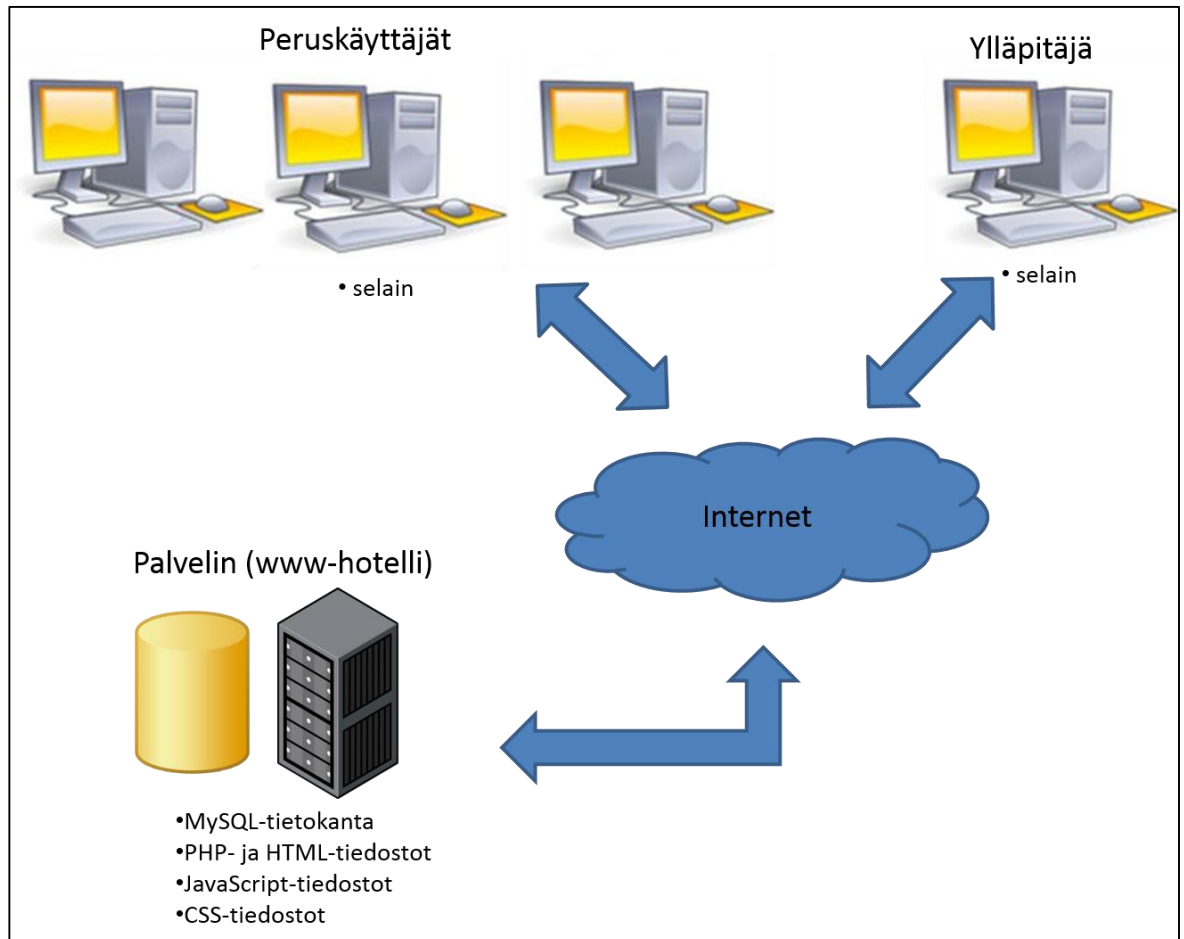
Navigointipalkissa näkyy Ylläpitäjä-valikko, mikäli sisäänkirjautuminen on suoritettu käyttäjätunnuksella, jonka taso on Ylläpitäjä.

Järjestelmässä on sisäinen ohje järjestelmän käytön helpottamiseksi. Ohjeesta on omat versiot ylläpitäjälle ja peruskäyttäjälle.

Järjestelmässä on sisäinen viestintätoiminto peruskäyttäjän ja ylläpitäjän väliseen kommunikointiin. Tästä toiminnosta käytetään nimitystä Ylläpitopyyntö, jonka avulla peruskäyttäjä voi pyytää ylläpitäjää tekemään jonkin asian järjestelmässä, jota hän ei itse voi suorittaa. Esimerkki tällaisesta toiminnosta on palvelun lisääminen.

6.2 Järjestelmän rakennekaavio

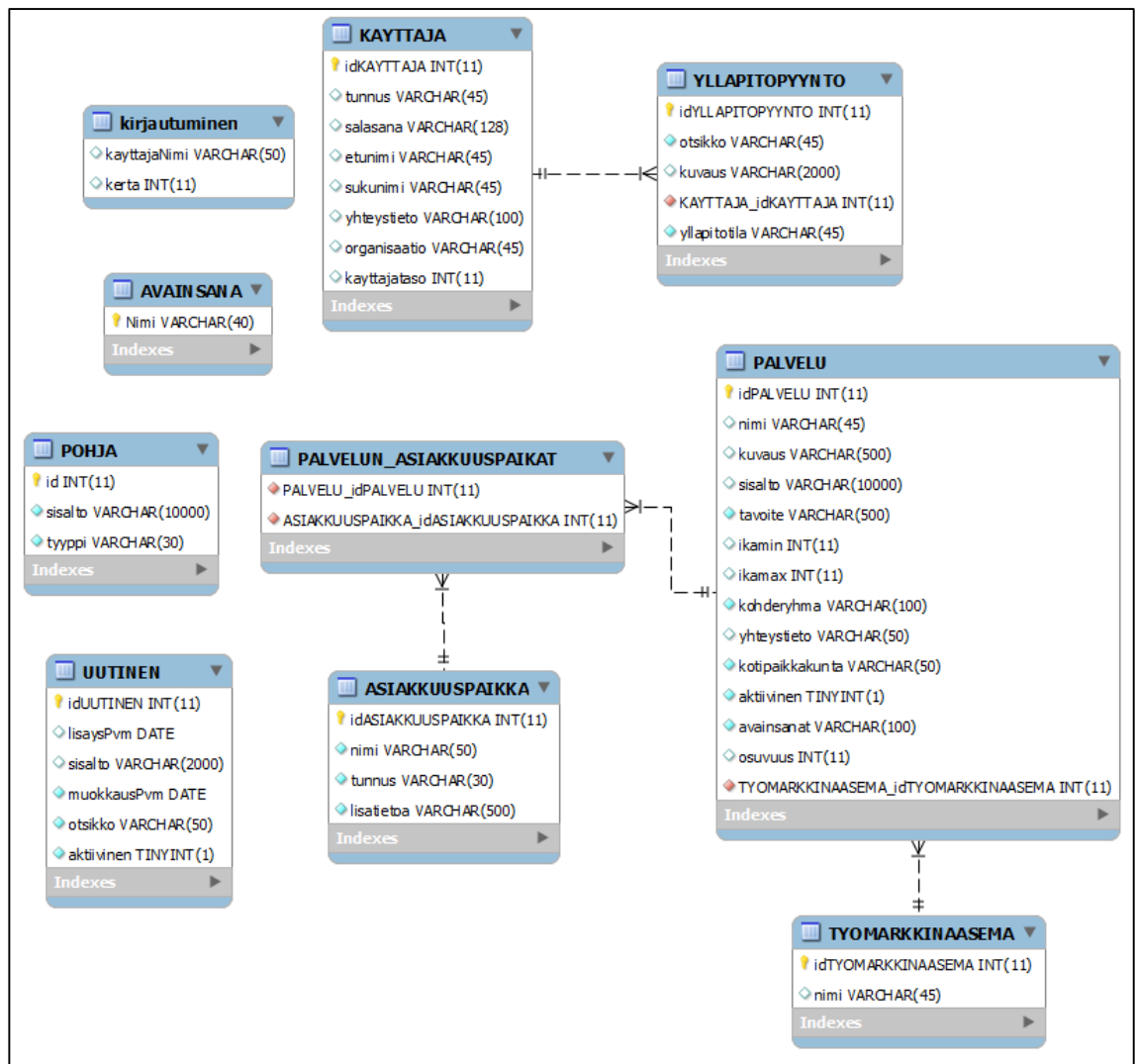
Kuva 26. Laitteistokaavionimetään järjestelmän laitteet sekä käyttäjäroolit ja niiden väliset yhteydet.



Kuva 26. Laitteistokaavio

6.3 Tietokanta

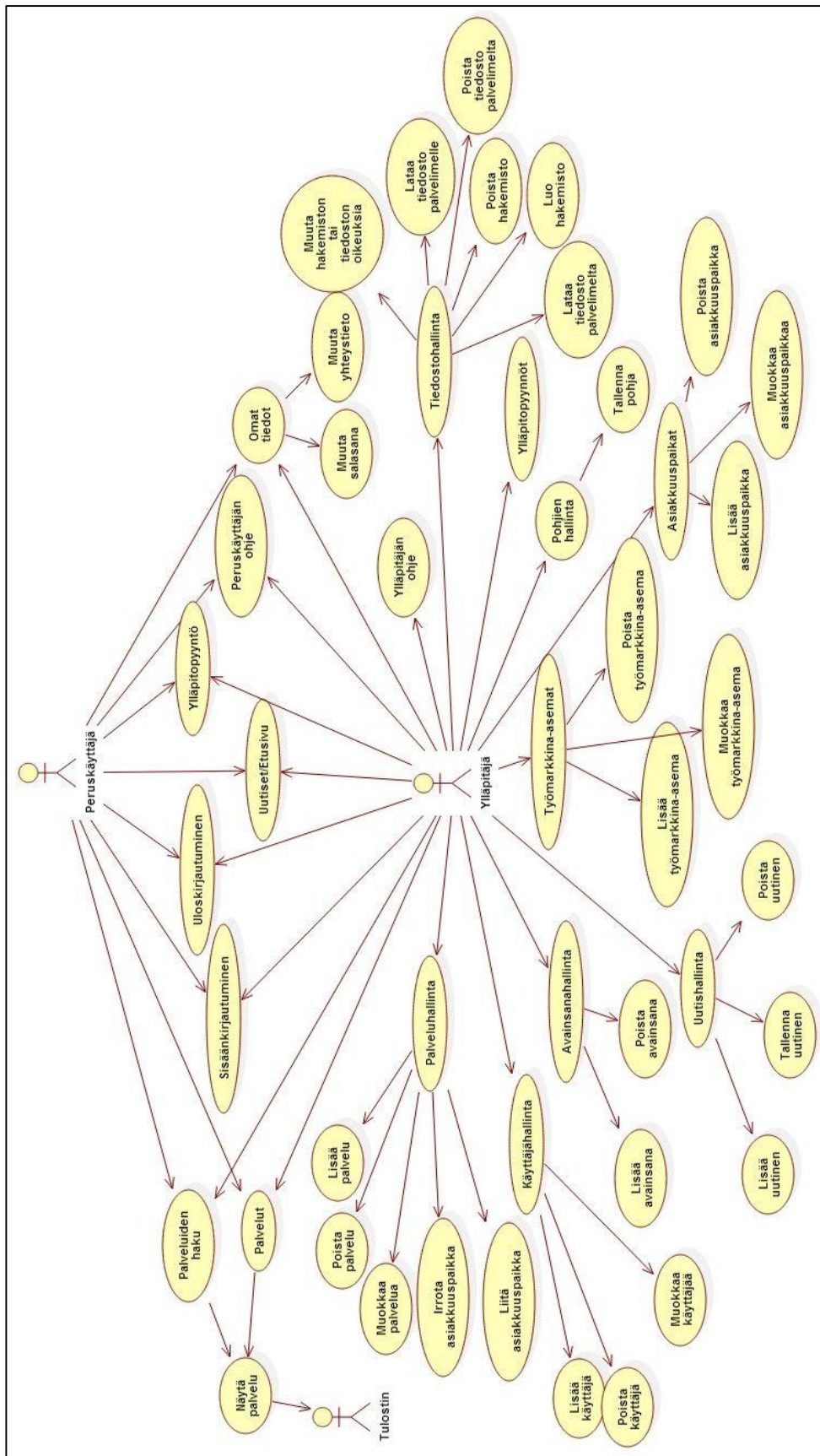
Järjestelmän tietokanta suunniteltiin tietokantakaavion pohjalta. Tietokannan suunnittelussa käytettiin apuna MySQL Workbench -ohjelmistoa sekä selainpohjaista phpMyAdmin-hallintatyökalua. Järjestelmän tietokantakaavio on esitetty Kuva 27.



Kuva 27. Tietokantakaavio

6.4 Käyttötapaukset

Järjestelmässä on kahdentyyppisiä käyttötapauksia: ylläpitäjän käyttötapaukset ja peruskäyttäjän käyttötapaukset. Ylläpitäjällä on molempien roolien oikeudet. Järjestelmän käyttötapaukset on esitetty Kuva 28.

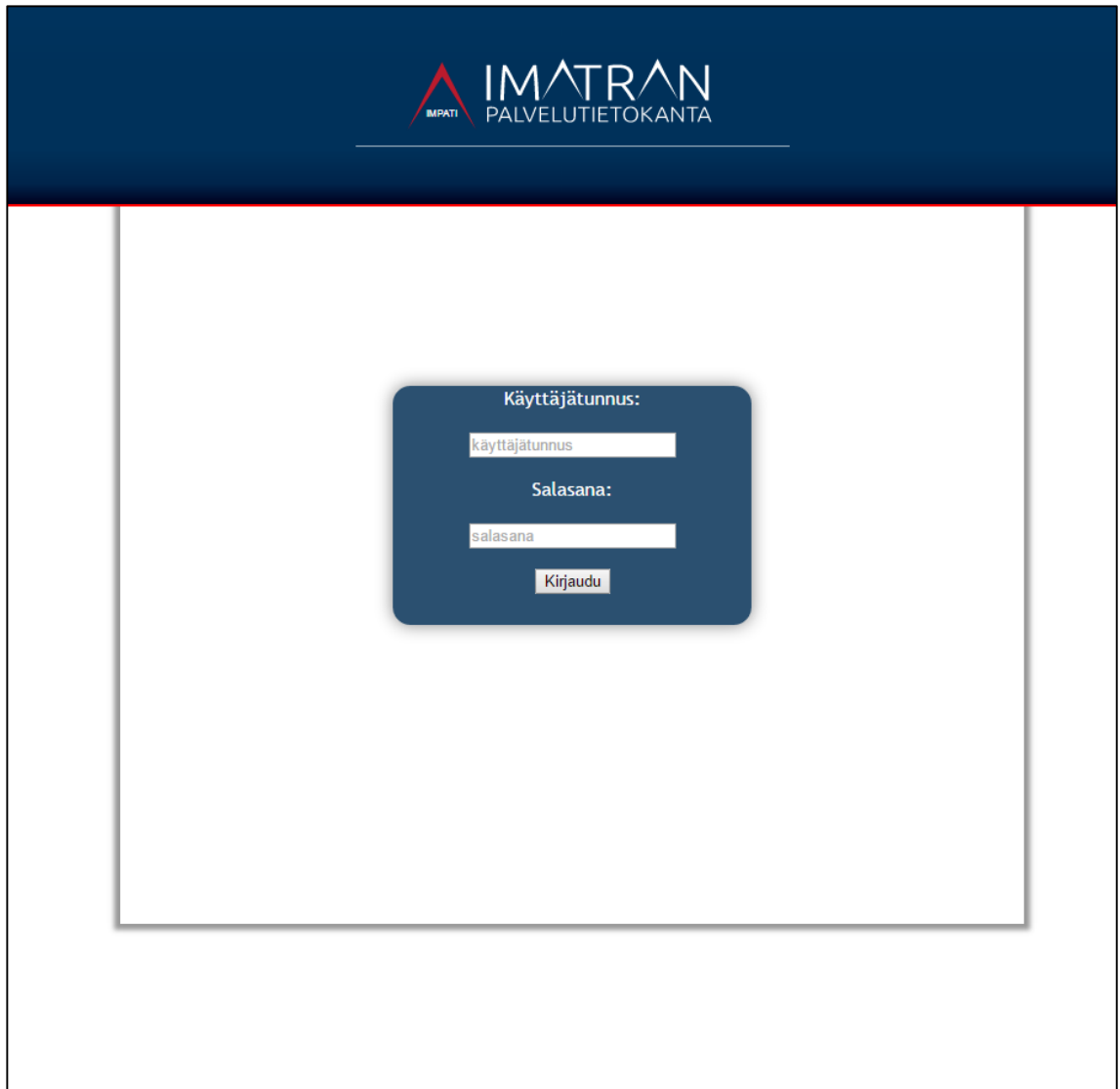


Kuva 28. Käyttötapauskaavio

Seuraavissa luvuissa esitellään järjestelmän keskeisimmät käyttötapaukset.

6.5 Sisäänkirjautuminen

Sisäänkirjautuminen järjestelmään tapahtuu syöttämällä käyttäjätunnus ja salasana sekä painamalla **kirjaudu**-painiketta (Kuva 29). Kirjautumisen yhteydessä järjestelmä tunnistaa käyttäjän peruskäyttäjäksi tai ylläpitäjäksi.



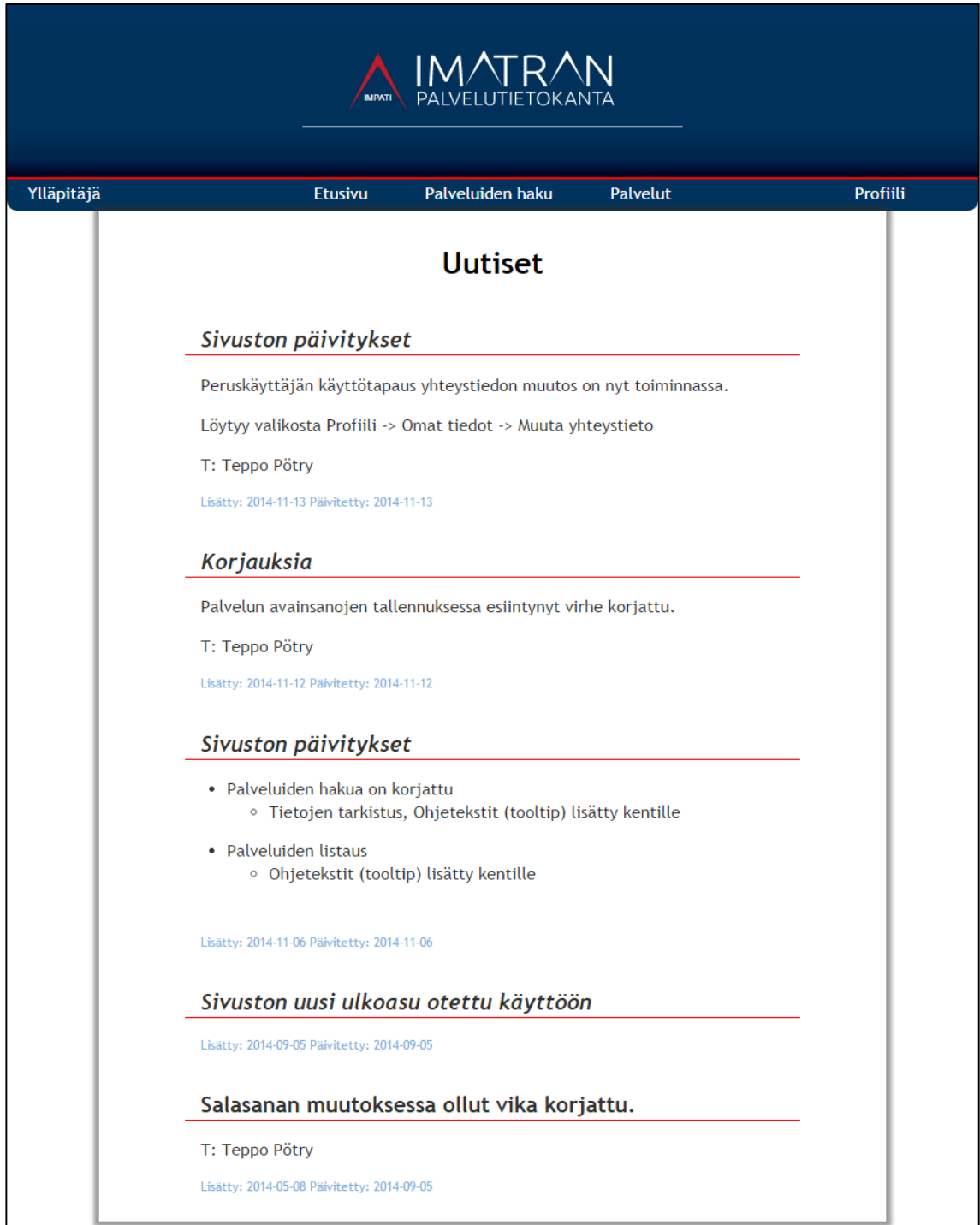
The image shows a login interface for the IMATRAN PALVELUTIETOKANTA system. At the top, there is a dark blue header with the logo on the left, which includes a red triangle and the text 'IMATRAN' and 'PALVELUTIETOKANTA'. Below the header is a white rectangular area containing a dark blue login form. The form has the following elements:

- A label 'Käyttäjätunnus:' above a text input field containing the placeholder text 'käyttäjätunnus'.
- A label 'Salasana:' above a text input field containing the placeholder text 'salasana'.
- A button labeled 'Kirjaudu' at the bottom of the form.

Kuva 29. Sisäänkirjautuminen

6.6 Etusivu

Käyttäjä ohjataan **Etusivulle** sisäänkirjautumisen jälkeen, jossa listataan järjestelmän uutiset lisäyispäivämäärän mukaan laskevaan järjestykseen (Kuva 30).



The screenshot shows the home page of the IMATRAN PALVELUTIETOKANTA system. At the top, there is a dark blue header with the logo and name. Below the header is a navigation bar with five items: Ylläpitäjä, Etusivu, Palveluiden haku, Palvelut, and Profiili. The main content area is titled "Uutiset" and contains four news items, each with a red underline:

- Sivuston päivitykset**
Peruskäyttäjän käyttötapaus yhteystiedon muutos on nyt toiminnassa.
Löytyy valikosta Profiili -> Omat tiedot -> Muuta yhteystieto
T: Teppo Pötry
Lisätty: 2014-11-13 Päivitetty: 2014-11-13
- Korjauksia**
Palvelun avainsanojen tallennuksessa esiintynyt virhe korjattu.
T: Teppo Pötry
Lisätty: 2014-11-12 Päivitetty: 2014-11-12
- Sivuston päivitykset**
 - Palveluiden hakua on korjattu
 - Tietojen tarkistus, Ohjetekstit (tooltip) lisätty kentille
 - Palveluiden listaus
 - Ohjetekstit (tooltip) lisätty kentille
Lisätty: 2014-11-06 Päivitetty: 2014-11-06
- Sivuston uusi ulkoasu otettu käyttöön**
Lisätty: 2014-09-05 Päivitetty: 2014-09-05
- Salasanan muutoksessa ollut vika korjattu.**
T: Teppo Pötry
Lisätty: 2014-05-08 Päivitetty: 2014-09-05

Kuva 30. Etusivu

6.7 Palveluiden haku

Käyttäjä syöttää asiakkaan iän ja kotipaikkakunnan lomakkeen kenttiin, valitsee **Työmarkkina-asema**-valikosta asiakkaan työmarkkina-aseman, valitsee asiakkaan **Asiakkuuspaikat** sekä **Avainsanat** haettavalle palvelulle ja painaa **Hae**-painiketta. Järjestelmä listaa löydetyt palvelut sopivuuden mukaan laskevasti (Kuva 31). Sopivuus lasketaan avainsanojen perusteella. Ikä-kenttä on pakollinen ja sen on oltava välillä 15–99.

IMATRAN
PALVELUTIETOKANTA

Ylläpitäjä Etusivu Palveluiden haku Palvelut Profiili

Palveluiden haku

Ikä:

Kotipaikkakunta:

Työmarkkina-asema:

Asiakkuuspaikat:

Aikuisosiaalityö
 IntoPajat ry
 Mielenterveyspalvelut

Sampo
 TE-toimisto
 Työn Vuoksi ry

Avainsanat:

Arjenhallinta
 Kuntoutus
 Maahanmuuttaja

Opintohaku
 Opiskelu
 Opiskeluvalmiudet

TYP
 Vammaispalvelut

Työelämä

Hae

Kuntouttava työtoiminta (TyönVuoksi ry)
Kuntouttava toiminta henkilökohtaisen aktivointisuunnitelman mukaan

Työkokeilu (TyönVuoksi)
Työkokeilu

Edelleen sijoitus (TyönVuoksi ry)
Tuettu työllistyminen yrityksiin tai yhdistyksiin jonkin mittaisen pajajakson jälkeen.

Takaisin työmarkkinoille (TyönVuoksi ry)
Valmentautumisjakson aikana asiakkaan tilanne, työkyky ja työssäsuoriutuminen arvioidaan ja laaditaan jatkosuunnitelma

Kuva 31. Palveluiden haku

6.8 Palveluiden listaus

Palvelut-sivulla käyttäjä voi listata kaikki järjestelmässä olevat palvelut. Sivulla on myös mahdollista hakea palvelua sen nimellä tai lajitella palveluita eri kriteerien mukaan (Kuva 32).



Kuva 32. Palveluiden listaus

6.9 Palvelun näyttö

Käyttäjä voi halutessaan tarkastella tarkempia tietoja palvelusta painamalla palvelun nimeä joko **Palveluiden hakutulosten** tai **Palveluiden listauksen** kautta. **Palvelun näyttö** -sivulla näkyy muuan muassa tarkempi kohderyhmä, tavoite ja yhteystiedot (Kuva 33).

**EDELLEEN SIOJITUS ELI SIIRTÄMINEN TOISEN TYÖNANTAJAN PALVELUKSEEN****TyönVuoksi ry****Honkaharju 7****55800 IMATRA****Kohderyhmä**

Edelleen sijoitus on tarkoitettu TyönVuoksi ry:n projektiin kiinnittyville, työttömille asiakkaille, joilla on oikeus palkkatukeen.

Edelleen sijoitus on tarkoitettu ensisijaisesti:

- TyönVuoksen projektilaisille (TTM)
- työttömille, palkkatuki oikeutetuille.

Tavoite

Avoimille työmarkkinoille, yrityksiin/yhdistyksiin työllistymisen avustaminen palkkatuella.

Palvelun sisältö

Asiakkaalla on mahdollisuus *tuettuun työllistymiseen* yrityksiin tai yhdistyksiin x-mittaisen pajajakson jälkeen. TyönVuoksen olemassa olevat/uudet yritys yhteistyö verkostot käytetään asiakkaan hyväksi työllistymisen portaissa. Siirtyminen pajajakson jälkeen toisen työnantajan palvelukseen tehdään pienin askelin ja kivuttomasti. Työ- ja yksilövalmennusta voidaan jatkaa vielä siirtymisen jälkeenkin yrityksissä/yhdistyksissä.

Etuudet

Työsuhde on määräaikainen, osa-aikainen tai täysipäiväinen. Edelleen sijoituksissa yrityksiin noudatamme TES:n mukaista palkkausta.

Palveluun hakeutuminen

Palveluun ohjaututaan TE-toimiston tai Työvoiman palvelukeskuksen kautta.

Lisätietoja Johanna Ovaska p. 050 435 9676

TyönVuoksi ry
Honkaharju 7
55800 Imatra

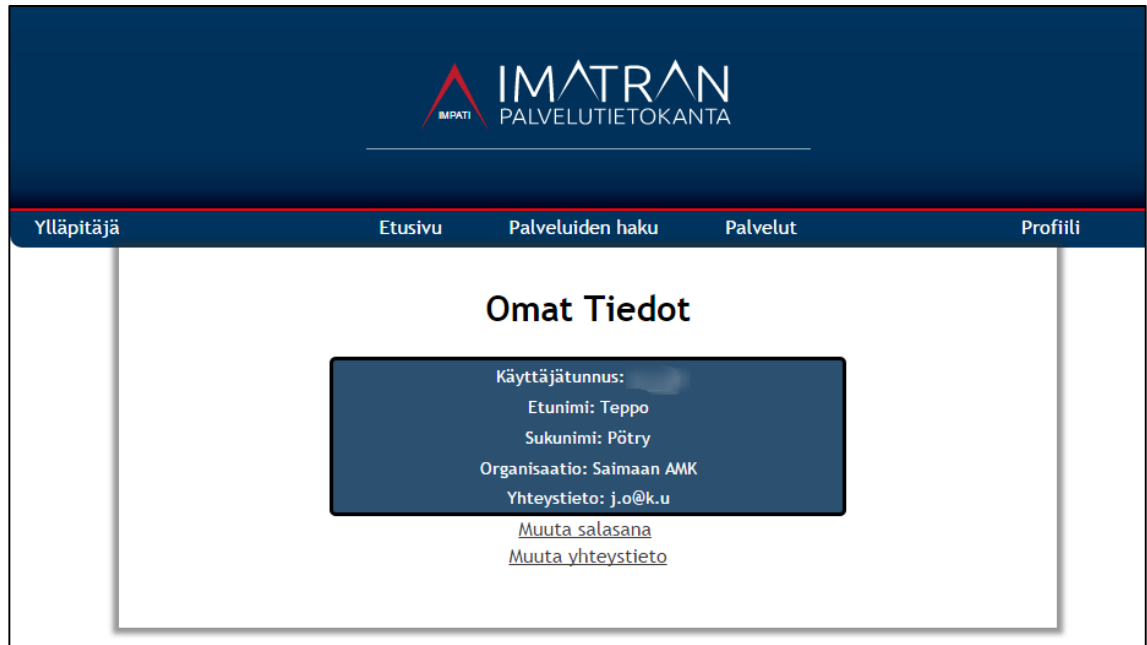
Tuloste asiakkaalle

- PDF-tiedostot tulee ladata ajaXplorerilla Impati-palvelimelle hakemistoon "Tiedostot/Pdf"
- Tiedostoviite tehdään palvelupohjaan ckEditorin "Lisää linkki"-toiminnolla ([http](#), uusi ikkuna)
- PDF-tiedostojen täydellinen osoite on

Kuva 33. Palvelun näyttö

6.10 Omien tietojen hallinta

Omat tiedot -sivulla käyttäjä voi tarkastella omia tietojaan. Sivulla näkyvät tiedot: käyttäjätunnus, etunimi, sukunimi, organisaatio ja yhteystieto (Kuva 34). Sivulla on myös mahdollisuus vaihtaa salasana tai muuttaa yhteystietoa.



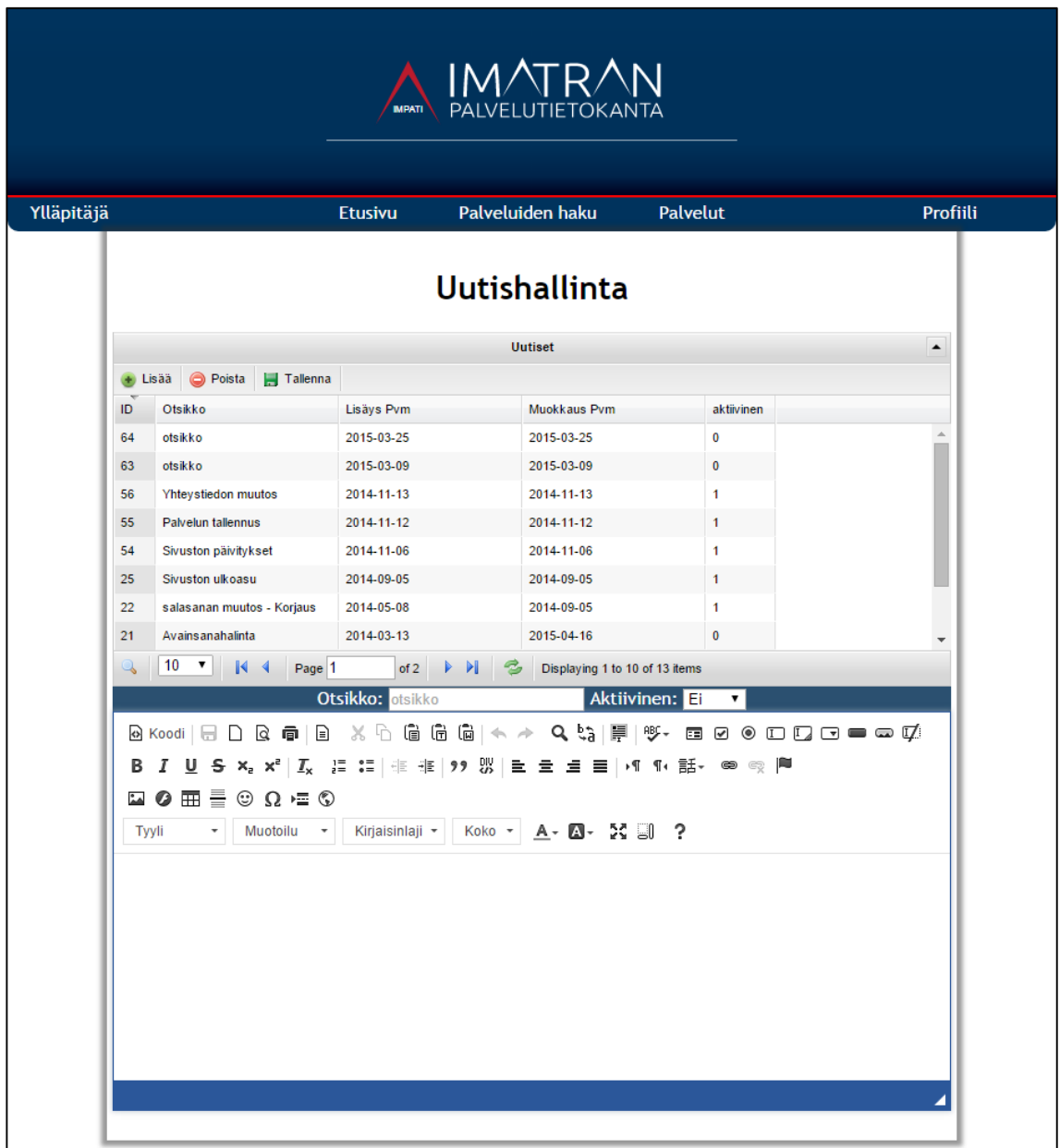
Kuva 34. Omat tiedot

6.11 Uloskirjautuminen

Uloskirjautumisen yhteydessä poistetaan käyttäjäkohtaiset asetukset ja ohjataan käyttäjä kirjautumissivulle.

6.12 Uutistenhallinta

Uutistenhallinnan (Kuva 35) avulla ylläpitäjällä on mahdollisuus lisätä järjestelmään uusi uutinen, muokata olemassa olevaa uutista tai poistaa uutinen järjestelmästä. **Lisäyspäivämäärä-** ja **Muokkauspäivämäärä-** kentät luodaan automaattisesti lisäyksen ja tallennuksen yhteydessä. Uutisen **Aktiivisuus-** kenttä vaikuttaa uutisen näkyvyyteen etusivulla. **Otsikko-** kenttä näkyy ainoastaan **Uutistenhallinta-** sivulla.



The screenshot displays the 'Uutishallinta' (News Management) interface. At the top, there is a navigation bar with links for 'Ylläpitäjä', 'Etusivu', 'Palveluiden haku', 'Palvelut', and 'Profiili'. The main content area is titled 'Uutishallinta' and contains a table of news items. The table has columns for ID, Otsikko, Lisäys Pvm, Muokkaus Pvm, and aktiivinen. Below the table, there is a search bar and a rich text editor for editing the selected item.

ID	Otsikko	Lisäys Pvm	Muokkaus Pvm	aktiivinen
64	otsikko	2015-03-25	2015-03-25	0
63	otsikko	2015-03-09	2015-03-09	0
56	Yhteystiedon muutos	2014-11-13	2014-11-13	1
55	Palvelun tallennus	2014-11-12	2014-11-12	1
54	Sivuston päivitykset	2014-11-06	2014-11-06	1
25	Sivuston ulkoasu	2014-09-05	2014-09-05	1
22	salasanan muutos - Korjaus	2014-05-08	2014-09-05	1
21	Avainsanahallinta	2014-03-13	2015-04-16	0

Kuva 35. Uutistenhallinnan pääsivu

6.13 Käyttäjähallinta

Käyttäjähallinnan kautta ylläpitäjä voi hallita järjestelmän käyttäjiä. **Käyttäjähallinnan**-pääsivulla näkyy kaikki järjestelmässä olevat käyttäjät listattuna **Flexigrid**-komponenttiin (Kuva 36).

The screenshot shows the user management interface for IMATRAN PALVELUTIETOKANTA. The page title is "Käyttäjien hallinnointi". The interface includes a navigation bar with links for "Ylläpitäjä", "Etusivu", "Palveluiden haku", "Palvelut", and "Profili". Below the navigation bar, there is a search bar and a table of users. The table has columns for ID, Tunnus, Etunimi, Sukunimi, Yhteystieto, Organisaatio, and Käyttäjä. The table lists 10 users. At the bottom of the table, there are pagination controls showing "Page 1 of 2" and "Displaying 1 to 10 of 17 items".

ID	Tunnus	Etunimi	Sukunimi	Yhteystieto	Organisaatio	Käyttäjä
1		Sami	Repo		Yksityinen	1
16		Teppo	Pötry		Saimaan AMK	1
19		Juha	Rasmus		InfoPajat ry	1
21		Juha	Rasmus		InfoPajat ry	2
22		Sami	Repo		sdsdss	1
25		Tomi	Kiiveri		Saimaan Ammattikoulu	1
26		Tuomas	Porvalli		Saimia	1

Kuva 36. Käyttäjähallinnan pääsivu

6.13.1 Käyttäjän lisääminen

Ylläpitäjä lisää käyttäjän järjestelmään Kuva 37 mukaisen lomakkeen avulla. Kaikki lomakkeella olevat kentät ovat pakollisia. **Generoi**-linkin avulla ylläpitäjä voi luoda **Salasana**-kenttiin kelpoisuusvaatimukset täyttävän salasanan kirjoittamisen sijaan. **Tulosta**-painikkeen avulla voidaan tulostaa käyttäjän tiedot ennen käyttäjän lisäämistä järjestelmään.

Lisää käyttäjä ✕

Kaikki kentät pakollisia.

käyttäjätunnus:
käyttäjätunnus

etunimi:
etunimi

sukunimi:
sukunimi

salasana:
salasana

salasana:
salasana uudelleen

Generoi
 Näytä salasana

yhteystieto:
yhteystieto

organisaatio:
organisaatio

Käyttäjätaso:
Peruskäyttäjä ▼

Tulosta Lisää Sulje

Kuva 37. Lisää käyttäjä -lomake

6.13.2 Käyttäjän tietojen muokkaaminen

Ylläpitäjä voi muokata käyttäjä tietoja Kuva 38 mukaisen lomakkeen avulla. Kaikki kentät salasanaa lukuun ottamatta ovat pakollisia. Mikäli **Salasana**-kentät jätetään tyhjäksi, vanha salasana jää voimaan.

Muokkaa käyttäjää: [User Icon] (16) x

Salasana ei ole pakollinen. Kaikki muut kentät ovat pakollisia.

etunimi:
Teppo

sukunimi:
Pötry

salasana:
salasana

salasana:
salasana uudelleen

Näytä salasana

[Generoi yhteystieto:](#)
j.o@k.u

organisaatio:
Saimaan AMK

Käyttäjätaso:
Ylläpitäjä

Tulosta Tallenna Sulje

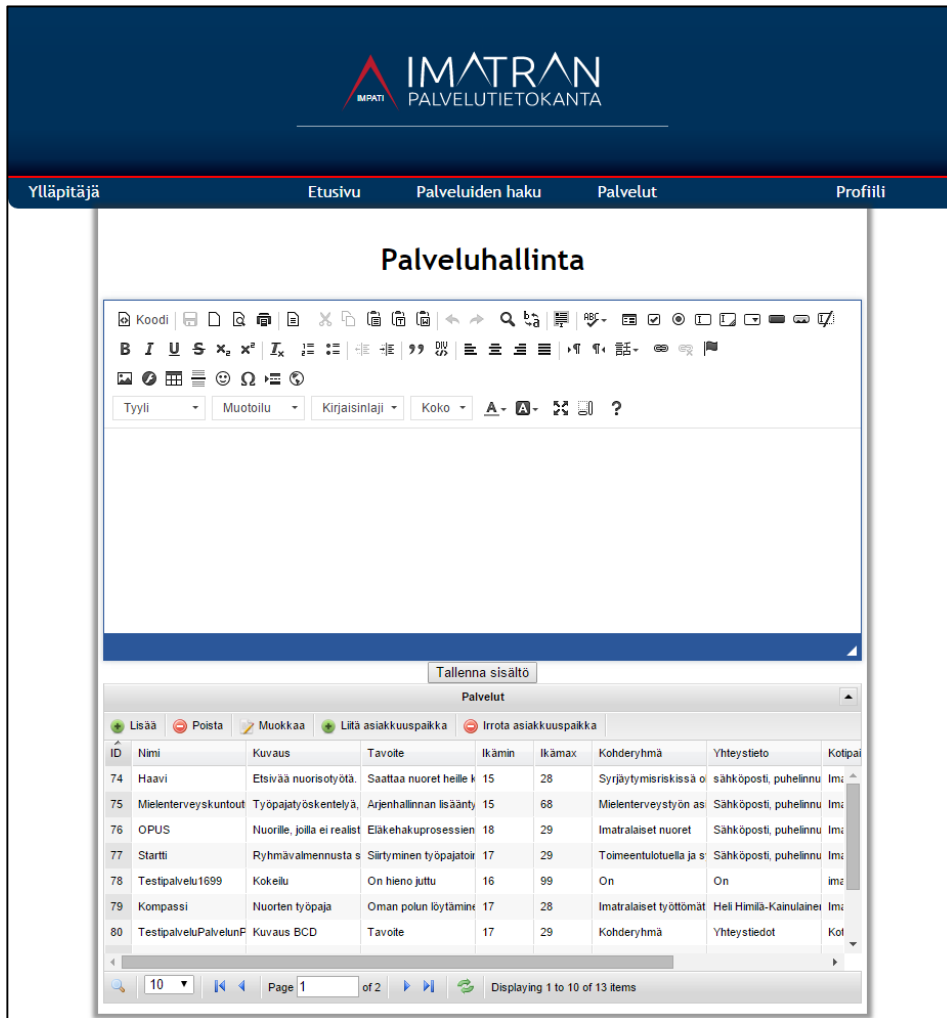
Kuva 38. Muokkaa käyttäjää -lomake

6.13.3 Käyttäjän poistaminen

Ylläpitäjä voi halutessaan poistaa käyttäjän järjestelmästä valitsemalla ensin käyttäjän ja sen jälkeen painamalla **Poista**-painiketta (Kuva 36).

6.14 Palveluhallinta

Palveluhallinnan kautta ylläpitäjä voi hallinnoida järjestelmässä olevia palveluita. Palveluhallinnan-pääsivulla näkyy kaikki järjestelmässä olevat palvelut listattuna **Flexigrid**-komponenttiin (Kuva 39). Ylläpitäjällä on mahdollisuus lisätä uusi palvelu järjestelmään, muokata olemassa olevaa palvelua, poistaa palvelu sekä hallinnoida palvelun vaatimuksena olevia asiakkuuspaikkoja.



Kuva 39. Palveluhallinnan pääsivu

6.14.1 Palvelun lisääminen

Ylläpitäjä lisää uuden palvelun järjestelmään Kuva 40 mukaisen lomakkeen kautta. **Aktiivisuus**-kenttä vaikuttaa palvelun näkyvyyteen hakutuloksissa (Luku 6.7) sekä **Palveluiden listaus** -toiminnossa (Luku 6.8). **Kotipaikkakunta** ei ole pakollinen kenttä, muut kentät ovat pakollisia.

Kuva 40. Palvelun lisäys -lomake

6.14.2 Palvelun muokkaaminen


Palvelun tietojen muokkaaminen tapahtuu valitsemalla palvelu Flexigrid-komponentista ja painamalla **Muokkaa**-painiketta. Lomake on samanlainen kuin luvussa 6.14.1 oleva (Kuva 40), joten sitä ei esitellä tässä tarkemmin. Palvelun sisältöä muokataan valitsemalla palvelu **Flexigrid**-komponentista, jonka jälkeen sisältö latautuu **CKEditor**-komponenttiin (Kuva 41). Muutokset tallennetaan **Tallenna sisältö** -painikkeella.

Palveluhallinta

Koodi

B I U S x x° I_x := : : -E »

Tyylit: Tyyli | Normaali | Kirjainlaji: | Koko: A- A+ ↺ ↻ ?


 Palvelun nimi
 IntoPajat ry
 Honkaharju 7
 55800 IMATRA

Kohderyhmä
 Ryhmä, jolle palvelu on suunnattu. Ryhmä voi rajautua esim. koulustason, työmarkkina-aseman, iän, kotipaikkakunnan tai tietyn erityisperusteen mukaan.

Tavoite
 Tavoite (oletus) ja tavoitteet listalla on palvelun mukaisesti mahdollista

Koodi: 558000

Tallenna sisältö

Lisää ➕ Poista ➖ Muokkaa ✎ Liitä asiakkuuspaikka ➕ Irrota asiakkuuspaikka ➖

ID	Nimi	Kuvaus	Tavoite	Ikämin	Ikämax	Kohderyhmä	Yhteystieto	Kotipa
77	Startti	Ryhmävalmennusta s	Siirtyminen työpajatoir	17	29	Toimeentulotuella ja s	Sähköposti, puhelinnu	Imz
78	Testipalvelu1699	Kokeilu	On hieno juttu	16	99	On	On	Imz
79	Kompassi	Nuorten työpaja	Oman polun löytämine	17	28	Imatralaiset työttömät	Heli Himilä-Kainulainei	Imz
80	TestipalveluPalvelunP	Kuvaus BCD	Tavoite	17	29	Kohderyhmä	Yhteystiedot	Kol
81	Työpajakoulu	Sampon opiskelijoille,	Varmistaa opintojen o	16	68	Sampon opiskelijat.	Heli Himilä-Kainulainei	Imz
85	Testauspalvelu	kuvaus	tavoite	16	99	kohde	yhteys	Imz
86	Työkokeilu (TyönVuo	Työkokeilu	Tavoite	16	99	Työttömät	56756756756	Imz

10 Page 1 of 2 Displaying 1 to 10 of 13 items

Kuva 41. Palvelun sisällön muokkaaminen

7 Yhteenveto

Opinnäytetyön tavoitteena oli toteuttaa asiakkaan vaatimuksien mukainen IMPATI-järjestelmä. Järjestelmän peruskäyttäjän käyttöliittymien tuli olla selkeitä ja helppokäyttöisiä, että perus ATK-aidot omaava henkilö oppii järjestelmän käytön helposti ja nopeasti. Tässä vaiheessa on vaikea sanoa, onnistuttiinko tässä, koska järjestelmää ei ole saatu vielä käyttöön otettua. Esittelytilaisuuksien perusteella palaute on kuitenkin ollut positiivista ja vaikuttaa siltä, että järjestelmälle on käyttöä tulevaisuudessa.

Omat tavoitteeni olivat web-palveluiden toteutustekniikoiden oppiminen, projektin läpi vieminen ja siihen liittyvä projektinhallinta sekä opinnäytetyön tekeminen. Projektin aikana opin, että ohjelmistoprojektiin liittyy paljon muutakin kuin koodin kirjoittaminen. Tällä hetkellä näyttää siltä, että projekti saadaan vietyä läpi kevään 2015 aikana.

Vaikeaa projektissa oli aikataulussa pysyminen, koska kaikilla kehittäjillä oli vähän kokemusta käytetyistä tekniikoista, ja tästä johtuen työmääräarviot olivat aikamoista arvailua. Kunnollisen versionhallinnan puuttumisesta seurasi, että kaikilla kehittäjillä ei ollut aina oikeaa lähdekoodin versiota käytössä, ja tästä seurasi sekaannuksia. Aluksi oli myös vaikeaa ymmärtää järjestelmän toiminta laajemmasta näkökulmasta, mutta tästä selvittiin säännöllisten viikkopalaverien ansiosta. Työmäärän arviointi oli myös hankalaa ja etenkin projektin alkuvaiheessa ne osoittautuivat liian optimistisiksi. Tästä syystä päätettiin, että projektiin ei otettu enempää tekijöitä, joka myöhemmin osoittautui vääräksi ratkaisuksi. Toteutuksessa käytetyt ohjelmointikielot ja -tekniikat olivat uusia, joten opiskeluun piti varata aika paljon aikaa.

Järjestelmään on tarkoitus tehdä viimeiset testaukset sekä korjaukset 24.4.2015 mennessä ja järjestelmä otetaan käyttöön toukokuun 2015 aikana.

Projekti tuotti asiakkaalle toiminnallisen määrittelyn, ylläpitäjän käyttöohjeen ja testatun IMPATI-järjestelmän.

Kuvat

- Kuva 1. Vesiputousmalli, s. 13
- Kuva 2. UML kaaviohierarkia, s. 21
- Kuva 3. Aktiviteettikaavio, s. 22
- Kuva 4. Ajoneuvojen vuokrausjärjestelmän käyttötapaukskaavio, s. 22
- Kuva 5. Tilakaavio, s. 23
- Kuva 6. Ajoituskaavio, s. 23
- Kuva 7. Kokoava vuorovaikutuskaavio, s. 24
- Kuva 8. Kommunikointikaavio, s. 25
- Kuva 9. Sekvenssikaavio, s. 25
- Kuva 10. Komponenttikaavio, s. 26
- Kuva 11. Koostekaavio, s. 26
- Kuva 12. Luokkakaavio, s. 27
- Kuva 13. Oliokaavio, s. 27
- Kuva 14. Pakkauskaavio, s. 28
- Kuva 15. Sijoittelukaavio, s. 28
- Kuva 16. Internetin fyysinen rakenne, s. 31
- Kuva 17. TCP/IP-protokollaperhe, s. 31
- Kuva 18. Adobe Dreamweaver CS 6 käyttöliittymä, s. 37
- Kuva 19. StarUML käyttöliittymä (versio 1), s. 38
- Kuva 20. Notepad++ käyttöliittymä, s. 39
- Kuva 21. MySQL Workbench käyttöliittymä, s. 40
- Kuva 22. cPanel käyttöliittymä, s. 41
- Kuva 23. phpMyAdmin käyttöliittymä, s. 42
- Kuva 24. IMPATI-projektin organisaatiokaavio, s. 43
- Kuva 25. Arkkitehtuurikaavio, s. 48
- Kuva 26. Laitteistokaavio, s. 51
- Kuva 27. Tietokantakaavio, s. 52
- Kuva 28. Käyttötapauskaavio, s. 53
- Kuva 29. Sisäänkirjautuminen, s. 54
- Kuva 30. Etusivu, s. 55
- Kuva 31. Palveluiden haku, s. 56
- Kuva 32. Palveluiden listaus, s. 57
- Kuva 33. Palvelun näyttö, s. 58
- Kuva 34. Omat tiedot, s. 59
- Kuva 35. Uutistenhallinnan pääsivu, s. 60
- Kuva 36. Käyttäjähallinnan pääsivu, s. 61
- Kuva 37. Lisää käyttäjä -lomake, s. 62
- Kuva 38. Muokkaa käyttäjää -lomake, s. 63
- Kuva 39. Palveluhallinnan pääsivu, s. 64
- Kuva 40. Palvelun lisäys -lomake, s. 65
- Kuva 41. Palvelun sisällön muokkaaminen, s. 66

Lähteet

2Kmediat. CSS-opas. <http://www.2kmediat.com/css/johdanto.asp>. Luettu 27.4.2015

Etelä-Karjalan työvoimanpalvelukeskus.
<http://www.tyovoimanpalvelukeskus.fi>. Luettu 15.4.2014

Finlander J-E. Pilviteknologia. <http://colibrix.net/cloud/wiki/pilvipalvelu/>. Luettu 24.4.2015

Haikala I & Mikkonen T. 2011. Ohjelmistotuotannon käytännöt. Helsinki: Talentum

Haikala I & Märijärvi J. 2004. Ohjelmistotuotanto. Helsinki: Talentum

Heinisuo, R & Rauta I. 2007. PHP ja MySQL Tietokantapohjaiset verkkopalvelut. Helsinki: Talentum

Häkkinen A. 2013. Mallintaminen. Luentokalvo. Metropolia ammattikorkeakoulu.
<http://users.metropolia.fi/~hakka/Ohma/Ohma-02.pdf>. Luettu 29.4.2015

IntoPajat ry 2014. Tietoa yhdistyksestä. <http://www.intopajat.fi/3>. Luettu 9.4.2015

IntoPajat ry 2014. Työpajat. <http://www.intopajat.fi/11>. Luettu 9.4.2015

Kuntaliitto 2015. Internet Governance. <http://www.kunnat.net/fi/asiantuntijapalvelut/tyk/tietohallinto/tietoverkot/internet/Sivut/default.aspx> . Luettu 24.4.2015
Kuntakokeilu. Hanke.
<http://www.kuntakokeilu.fi/hanke>. Luettu 15.4.2015

Kuntakokeilu. Taustaorganisaatiot.
<http://www.kuntakokeilu.fi/taustaorganisaatiot>. Luettu 15.4.2015

Laaksonen A. 2009. PHP-opas. <http://www.ohjelmointiputka.net/op-paat/opas.php?tunnus=mysqlphp12>. Luettu 19.4.2015

Lehdonvirta P & Korpela J. 2012. HTML5-kirja. Mitä HTML5-sovellukset ovat?.
<http://html5kirja.fi/2012/08/31/mita-html5-sovellukset-ovat/>. Luettu 27.4.2015

Luukkainen M & Laine H. 2010. Ohjelmistojen mallintaminen. Luentomoniste. Helsingin yliopisto. Tietojenkäsittelyn laitos

MySQL Workbench. <https://www.mysql.com/products/workbench/>. Luettu 19.4.2015

Netmarketshare 2015. <https://www.netmarketshare.com/>. Luettu 27.4.2015

Sand A & Hyrskykari A. JWT 2015 - Johdatus www-tekniikoihin.
<http://www.sis.uta.fi/~jwt/15/javascript.html>. Luettu 27.4.2015

StarUML. User guide. [http://staruml.sourceforge.net/docs/user-guide\(en\)/toc.html](http://staruml.sourceforge.net/docs/user-guide(en)/toc.html) Luettu 19.4.2015

StarUML. <http://staruml.io/>. Luettu 19.4.2015

TOIMARI-hanke 2014. Esite.
<http://www.imatra.fi/djulkaisu/kokous/20142057-5-3.PDF>. Luettu 16.4.2015

Työllisyyden toimijakartta. Imatra.
<http://www.toimijakartta-etelakarjala.fi/imatra>. Luettu 10.4.2015

TyönVuoksi ry. Valmennuspalvelut. <http://www.tyonvuoksi.fi/valmennuspalvelut.html>. Luettu 15.4.2015

TyönVuoksi ry. Osuuskunta TyönVuoksi. <http://www.tyonvuoksi.fi/valmennuspalvelut.html>. Luettu 15.4.2015

Vahtiohje 2009. Tekninen internet-infrastrukturi ja tietoturvallisuus.
<https://www.vahtiohje.fi/web/guest/tekninen-internet-infrastrukturi-ja-tietoturval-lisuus>. Luettu 24.4.2015

Varjo J. 2010. Laitteistotietokannan ja sen käyttöliittymän suunnittelu. Insinööri-tyo. Metropolia ammattikorkeakoulu. Helsinki. Luettu 28.4.2015

Wikipedia. HTTP. <http://fi.wikipedia.org/wiki/HTTP>. Luettu 24.4.2015.

Wikipedia: Notepad++. <http://en.wikipedia.org/wiki/Notepad%2B%2B>. Luettu 18.4.2015

Wikipedia. Palvelin. <http://fi.wikipedia.org/wiki/Palvelin>. Luettu 26.4.2015

Wikipedia: Verkkopalvelu. <http://fi.wikipedia.org/wiki/Verkkopalvelu>. Luettu 24.4.2015

Wikipedia. Webhotelli. <http://fi.wikipedia.org/wiki/Webhotelli>. Luettu 27.4.2015

Wikipedia. UML-mallinnus. <http://fi.wikipedia.org/wiki/UML-mallinnus>. Luettu 29.4.2015