

Tallinna Ülikool
Matemaatika-loodusteaduskond
Informaatika osakond

Eve Keerus-Jusupov

**Tarkvara arendusetapid Lääne-Virumaa Kutsekõrgkooli
õppeinfosüsteemi üliõpilasmooduli näitel**

Magistritöö

Juhendaja: Jaagup Kippar

Autor: “ ” 2005
Juhendaja: “ ” 2005
Osakonna juhataja: “ ” 2005

Tallinn 2005

Sisukord

Sissejuhatus.....	4
1. Eeluuring.....	6
1.1 Kutseõppeasutused ja õppeinfosüsteemid.....	6
1.1.1 Lääne-Virumaa Kutsekõrgkool.....	8
1.1.2 Tartu Kutsehariduskeskus.....	8
1.1.3 Võrumaa Kutsehariduskeskus.....	10
1.2 Seadused.....	11
2 Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi kavandamine.....	16
2.1 Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi üliõpilasmooduli kasutajate vajadused	16
2.2 Kasutajad.....	19
2.3 Olulised õppeinfosüsteemi parameetrid kasutajate jaoks.....	20
2.4 Nõuded.....	21
2.5 Kasutajaliides.....	24
3 Tarkvaraarenduse teooria.....	27
3.1 Tarkvaraarenduse metoodika.....	28
3.2 Projektide ebaõnnestumiste põhjused.....	29
3.3 Riskid.....	30
3.4 Tarkvara isetegemise ja väljastpoolt tellimise plussid ja miinused.....	31
4 Tarkvara loomise tehniline tagapõhi.....	33
4.1 Klient-server rakenduse tööpõhimõte ning PHP ja MySQL.....	33
4.2 Vorm ja vormi kohustuslike väljade kontrollimine	37
4.3 Andmebaas ja päringute tegemine.....	41
4.3.1 Päringud.....	45
4.4 Kasutaja autentimine ning HTTPS.....	47
5 Testimine.....	53
5.1 Kasutatavus.....	53
Kokkuvõte.....	58

Kasutatud kirjandus.....	60
Summary.....	63
Lisad.....	64
Lisa 1	
Intervjuu küsimused.....	64
Lisa 2	
Lääne-Virumaa Kutsekõrgkooli organisatsiooniline struktuur.....	65
Lisa 3	
VKHK õppetöö moodul.....	66
Lisa 4	
Andmebaasile mõjuvad ohud (ISKE).....	68
Lisa 5	
Andmebaasile mõjuvaid ohtusid vähendavad turvameetmed (ISKE).....	69
Lisa 6	
Õppeinfosüsteemi üliõpilasmooduli andmebaasi andmemudel.....	71
Lisa 7	
Kasutajajuhend.....	73
Lisa 8	
Kasutajate hinnangud.....	82

Sissejuhatus

Infotehnoloogia pidev areng loob võimalusi igapäevase töö lihtsustamiseks, samuti aitab leida kasutajatel vajaminevat informatsiooni kiiremini ja tulemusrikkamalt. Tänapäeva kasvavates hariduskeskustes ei ole mõeldav, et nende efektiivseks töötamiseks puudub õppeinfosüsteem, süsteem, mis tagab vajaminevate andmete kogumise, töötlemise, säilitamise ja väljastamise. Lähtudes Lääne-Virumaa Kutsekõrgkooli arengukava tänastest väljakutsetest, on õppeinfosüsteemi üliõpilasmooduli olemasolu ja rakendusvõimalused aktuaalne teema.

Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteem on loomisjärgus, selle esimene etapp – üliõpilasmoodul – valmis käesoleva töö käigus. Tarkvara arendusteooria läbitöötamine ja praktikasse rakendamine aitavad paremini mõista tarkvara arenduse elutsüklit, rakenduse loomise struktuuri ning tööprotsesside juhtimist.

Käesoleva magistritöö teemaks on “Tarkvara arendusetapid Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi üliõpilasmooduli näitel”.

Magistritöö eesmärk on:

- anda ülevaade tarkvara arendusetappidest Lääne-Virumaa Kutsekõrgkooli näitel;
- rakendust loova uurimuse kaudu analüüsida olukorda enne ja pärast üliõpilasmooduli loomist/rakendamist;
- õppeinfosüsteemi üliõpilasmooduli loomise/rakendamise kaudu lihtsustada ja kiirendada õppeosakonna tööülesannete täitmist.

Hetkeolukorda ning kasutajate vajadusi analüüsides võib kindlalt väita, et käesoleva rakenduse loomine on aktuaalne valdkond Lääne-Virumaa Kutsekõrgkoolile.

Magistritöö koosneb viiest peatükist. Töö esimene peatükk annab ülevaate tarkvaraarenduse esimesest etapist – eeluuringust. Enne rakenduse loomist on vajalik saada selge ülevaade probleemist ning võimalikest lahendustest, tutvuda analoogsete rakendustega, uurida kehtivaid õigusakte ning standardeid.

Magistritöö teine peatükk, Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi kavandamine, sisaldab tarkvaraarenduse teist etappi – analüüsi. Hetkeolukord, kasutajate vajaduste ning nõuete

analüüs, üliõpilasmooduli struktuur, õppeinfosüsteemi parameetrid, kasutajaliidese olulised omadused – need on märksõnad, mida käsitletakse teises peatükis.

Tarkvaraarendusteooria ja metoodika, projektide ebaõnnestumiste põhjused, riskid ning tarkvara isetegemise ja väljastpoolt tellimise plussid ja miinused ning miks valida väiksemahulistes tarkvaraarendustes protsesside paremaks juhtimiseks mitme-etapiline mudel? Tarkvaraarenduse teooria on kolmas peatükk ning käsitleb eelpool nimetatud teemasid.

Magistritöö neljas peatükk annab ülevaate järgmistest teemadest: klient-server rakenduse üldine põhimõte, PHP/MySQL-i eelised ja puudused, vorm ning vormi kohustuslike väljade kontrollimine, andmebaas, päringud, kasutaja autentimine. Tarkvara loomise tehniline tagapõhi käsitleb rakenduse disaini ning realisatsiooni, mis on tarkvaraarenduse kolmandaks etapiks.

Tarkvara testimine kui arenduse neljas etapp on tarkvaratootmise protsessi üks osa. Käesolevas üliõpilasmoodulis rakendatakse kasutatavuse testi, mille puhul pööratakse tähelepanu kasutamise õppimise lihtsusele, kasutamise lihtsusele, konkreetsete ülesannete täitmise lihtsusele, kasutajaliidese mugavusele, loogilisele ülesehitusele ning välisele disainile.

Kasutatavuse testimine ning kasutajate hinnangud on viienda peatüki teemakäsitluseks.

Magistritöö teoreetilise osa olulisteks lähteallikateks on Ian Sommerville raamat “Software Engineering” ning õigusaktid: Isikuandmete kaitse seadus, Andmekogude seadus ja Infosüsteemide kolmeastmelise etalonturbe süsteem ISKE rakendamisjuhend, praktilise poole tähtsamaks allikmaterjaliks on veebipõhine PHP käsiraamat.

1. Eeluuring

Eeluuringu eesmärgiks on õppeinfosüsteemi üliõpilasmooduli loomisel ja rakendamisel vajamineva informatsiooni kogumine, mis annab vastused küsimustele:

- Kuidas on loodud analoogsed toimivad süsteemid teistes kutseõppeasutustes (Võrumaa Kutsehariduskeskuses ja Tartu Kutsehariduskeskuses):
 - tarkvara valik;
 - olemasolevad probleemid;
 - struktuur.
- Kas ja millised õigusaktid reguleerivad üliõpilasmooduli loomisel tekkida võivaid probleeme:
 - hinnete kui isiku võimeid ning isiksuseomadusi kirjeldavate andmete töötlemine ja avalikustamine;
 - tervikluse, konfidentsiaalsuse ja käideldavuse tagamine.

Eelanalüüs kui üliõpilasmooduli loomise esimene faas annab ideid ja informatsiooni käesoleva rakenduse efektiivsemaks loomiseks ning rakendamiseks, aitab otsustada edasiste arendustööde planeerimisel ning kasutusele võtmisel.

Eeluuringu tulemustest annavad ülevaate järgmised alapeatükid, mis käsitlevad vastuseid eelpool püstitatud küsimustele ning probleemidele.

1.1 Kutseõppeasutused ja õppeinfosüsteemid

Esimese alapeatüki eesmärgiks on anda ülevaade õppeinfosüsteemi mõistest ning funktsioonidest, Lääne-Virumaa Kutsekõrgkooli struktuurist ning õpetatavatest erialadest, Võrumaa Kutsehariduskeskuse (VKHK) ja Tartu Kutsehariduskeskuse (TKHK) õppeinfosüsteemidest. Nimetatud kutsehariduskeskuste kohta esitatud informatsiooni ja kirjelduste algallikas on intervjuud TKHK infotehnoloogia talituse juhataja Rauno Jakobsoniga ja VKHK arendusdirektori Vadim Mitroškiniga. Intervjuu küsimused on esitatud lisas 1.

Õppeinfosüsteem (ÕIS)

Järgnevas alalõigus käsitletakse mõisteid infosüsteem ja õppeinfosüsteem ning selgitatakse nimetatud süsteemide eesmäärke.

Infosüsteem – omavahel sidestatud ja mitmel tasemel eksisteerivate andmete, funktsioonide, protsesside, sündmuste, asukohtade ja väärtuste süsteem (Mikli, 1998, lk 10).

Õppeinfosüsteem – õppetöö protsesside juhtimise abivahend. ÕIS-i all mõeldakse komplekti arvutiprogramme, mis toetavad õppeprotsessi läbiviimist ja õppetöö tulemustest kokkuvõtete tegemist (Tartu Ülikooli õppeinfosüsteemi veebipõhine abiinfo, 2005).

Käesolevas töös lähtutakse arvutiseeritud infosüsteemist, süsteemist, mille keskseks osaks on arvuti oma andmebaaside, elektroonsete asjaajamisvõimaluste ja dokumendihalduse ning veebilehtedega. Õppeinfosüsteemi all mõistetakse õppeasutuse infosüsteemi, mille ülesandeks on tagada vajaminevate andmete kogumine, töötlemine, säilitamine ja väljastamine.

Igal süsteemil, sealhulgas igal infosüsteemil, on tema funktsioneerimise eesmärk. Organisatsiooni funktsioonid on primaarsed, infosüsteemi funktsioonid sekundaarsed ja neid võib vaadelda kui abinõusid organisatsiooni funktsioonide täitmiseks.

Üldiselt võib õppeinfosüsteemi peamisteks funktsioonideks lugeda järgmised punktid:

- andmete kogumine, salvestamine, korrastamine, säilitamine, muutmine, nende kohta päringute teostamine, nendest väljavõtete tegemine, andmete kasutamine, üleandmine, ühendamine, kustutamine (Andmekogude seadus, RTI 2004, 30, 204, 28.04.2004);
- info väljastamine, levitamine.

Järgmiseks õppeinfosüsteemi funktsioonide tasemeks võib lugeda andmete riskasutusse viimist – andmete ülekandmist ühest andmekogust teise või mitmes andmekogus sisalduvate andmete ühist infotehnoloogilist töötlemist (Andmekogude seadus, RTI 2004, 30, 204, 28.04.2004). Eelnev kokku määratleb õppeinfosüsteemi peamise ülesande – eksisteerida ja tegutseda kui üks ühtne tervik, tagamaks informatsiooni kogumise, töötlemise, säilitamise ning väljastamise.

Ettevõtte juhi seisukohalt vaadates on infosüsteemi põhiülesanne kindlustada organisatsiooni juhtimiseks vajaliku info olemasolu: õige info õigel ajal, õiges kohas ja õigel kujul (IT juhtimise käsiraamat, 1999, lk 1).

(Õppe)infosüsteemide funktsioonide määramisel tuleb lähtuda kasutaja vajadustest. Paljude infosüsteemide sh õppeinfosüsteemide ebaõnnestumise põhjuseks on see, et eelnevalt ei ole piisava põhjalikkusega uuritud kasutaja vajadusi. Alapeatükis 2.1 esitatakse Lääne-Virumaa

Kutsekõrgkooli õppeinfosüsteemi üliõpilasmooduli kasutaja vajaduste loetelu ning üldine struktuur.

1.1.1 Lääne-Virumaa Kutsekõrgkool

Lääne-Virumaa Kutsekõrgkool on Lääne-Virumaal kutsekeskharidust, rakenduskõrgharidust ning täiskasvanute täiend- ja ümberõpet pakkuv riiklik kutseõppeasutus. Lääne-Virumaa Kutsekõrgkool moodustati 2001. aastal kahe kooli – Mõdriku Põllumajandustehnikumi ja Rakvere Pedagoogikakooli baasil. Koolis õpetatakse ärijuhtimise, raamatupidamise, sotsiaaltöö, tegevusjuht-kasvataja, kaubandusökonomika, müügiassistendi, äriinfotehnoloogia ja majandusinfo töötlemise erialasid. Ärijuhtimise ja sotsiaaltöö erialadel on loodud eraldi päevased õppegrupid, kus saavad õppida ka need, kes keskhariduse on omandanud venekeelses koolis. Lisaks on õppegrupp füüsilise puudega isikutele, kes õpivad sotsiaaltööd. Kõiki erialasid on võimalik omandada nii päevases kui kaugõppes. 2004/2005 õppeaastal õpib Lääne-Virumaa Kutsekõrgkoolis 1072 üliõpilast, töötajate arv on 64, neist põhikohaga pedagooge 32, ülejäänud on administreeriv ning abipersonal. Lisaks on 28 lepingu alusel töötavat pedagoogi (Lääne-Virumaa Kutsekõrgkool, 2004, lk 5).

Lisas 2 on esitatud Lääne-Virumaa Kutsekõrgkooli organisatsiooniline struktuur. Koolis on kõrgeim võim kooli nõukogul, millele allub direktor. Direktori otseses alluvuses töötavad juhiabi, pearaamatupidaja, direktori asetäitja, täiendkoolituse osakonna juhataja-personalijuht ning arendusspetsialist.

1.1.2 Tartu Kutsehariduskeskus

Tartu Kutsehariduskeskuse infosüsteem on loodud eesmärgiga parandada teabe kogumise, töötlemise ja edastamise kvaliteeti. Kuna keskus on suur - umbes 250 töötajat ja ligi 3000 kutsekoolituse õpilast, lisaks veel eelkoolituse ja täiendkoolituse õpilased, siis vajab keskuse juhtimine täpset ja kiiret informatsiooni. Süsteemi töökindluse tagavad 3 GHz Intel P4, FreeBSD server, MySQL andmebaasiserver ja Apache veebiserver (Tartu Kutsehariduskeskus, 2005).

Infosüsteemi väljund on valdavalt veebipõhine, jagunedes avalikeks ja mitteavalikeks lehtedeks. Infosüsteemi on võimalik jagada ka klientide lõikes: kutsekoolitus, eelkoolitus, nõustamine, täiendkoolitus, töötajad ning eraldi õpilaste portaal.

Rakendus on kirjutatud PHP/MySQL abil. Süsteemi on arendatud selles keskkonnas 4 aastat. Infotehnoloogia talituse juhataja Rauno Jakobson esitab PHP/MySQL-i valikuks järgmised põhjused: töökindel, kiire, odav ning positiivsed kogemused.

Kokku on antud hetkel 9 andmebaasi, mis jagunevad järgmiselt:

- kutsekoolituse andmebaas – 40 tabelit;
- nõustamise andmebaas – 6 tabelit;
- täiendkoolituse andmebaas – 8 tabelit;
- biblio andmebaas – 5 tabelit;
- üldised andmebaas – 44 tabelit;
- dokumentide andmebaas – 2 tabelit;
- portaali andmebaas – 15 tabelit.

Töölepingut sõlmides saab uus kooli töötaja oma kasutajanime ja parooli. Parooli saab töötaja hiljem muuta. Kui tavaõpetaja logib oma kasutajanime ja parooli abil sisse, siis näeb ta ainult neid õpilastegruppe, kellele tunde annab. Tavaõpetaja saab muuta ja sisestada hindeid, lisada ainekava vastava vormi abil ning õppematerjale. Kursusejuhataja näeb kõiki oma kursuse üliõpilaste hindeid ning saab neid lisada ja muuta, lisaks veel saab täiendada/lisada õpilaste andmeid, tõendeid, käskkirju, kompensatsioone, koondaruandeid, konsultatsiooniaegu, vahehindeid, koondhindeid, kokkuvõtvaid hindeid. Samuti sisestavad kursusejuhatajad iga nädala lõpus puudumised, selleks on oma vorm, kus saab lisada põhjusega ning põhjuseeta puudumised. Hiljem on võimalik sisestatud puudumisi muuta. Antud süsteem võimaldab printida lõpudokumente. Kasutusjuhendid on olemas hinnete lisamise juures. Tunniplaane on võimalik vaadata nii õpetaja, ruumi kui ka grupi järgi. Kõik päringud, mis veebis tehakse, salvestatakse andmebaasi. Kuna kasutajad on eelnevalt autoriseeritud, siis on võimalik vaadata näiteks, kes luges missugust õppematerjali ning mis kell. Kõik muutused säiluvad, kui kasutaja muudab oma parooli, siis tema andmed dubleeritakse. Infosüsteemi tehakse keskmiselt 50-60 päringut sekundis, kuna iga sõna lehelt teeb pöördumise andmebaasi poole.

Andmebaasis on umbes 3 miljonit kirjet. Kokku on 323 aktiivset kasutajat, lokaalseid kasutajaid on 2, veebipõhiselt kasutatakse andmete turvaliseks edastuseks HTTPS kanalit.

Probleemidest on tekkinud vaid kasutajast tulenevad probleemid – kasutajad ei lisa hindeid, ei täida esitatud ülesandeid.

Süsteem uueneb iga päev. Olemasolev infosüsteem on alalises täiustamisfaasis, pidevalt avastatakse uusi funktsioone, et muuta kasutajate elu lihtsamaks. Seega toimub järjepidev arendus, vajaminevate moodulite ja funktsioonide juurdekirjutamine.

1.1.3 Võrumaa Kutsehariduskeskus

Võrumaa Kutsehariduskeskuses töötab 40 kutseõpetajat ning õpib 825 õppurit. VKHK infosüsteem jaguneb järgmiselt: üldinfo, regulatsioonid, õppetöö, korraldus, haldus, dokumendid, raamatukogu, lingid ja kooli nõukogu. Antud infosüsteem on kirjutatud ASP/MSQL abil.

ASP (*Active Server Pages*) on Microsofti poolt välja töötatud tehnoloogia, mis koos IIS (Internet Information Service) veebiserveriga võimaldab dünaamilisi veebilehti luua. Skriptide kirjutamiseks kasutatakse peamiselt VBscripti, mis on tuletatud programmeerimiskeelest Visual Basic (Viigipuu, 2004, lk 7).

ASP puuduseks on platvormist sõltuvus, töötades üksnes Microsoft Windows platvormil. VKHK arendusdirektor Vadim Mitroškin tõi kasutatava tarkvara eeliseks Windows kasutatavuse ning väga hea klienditoe, mida vabavara tootjad alati pakkuda ei suuda.

Olemasolevat õppeinfosüsteemi on arendatud aastast 1997. Tarkvara valiku põhjused esitas VKHK kvaliteetidirektor Vadim Mitroškin järgmised: töökindlus, töötav kasutajatugi ning isiklikud oskused/kogemused.

Infosüsteemis on üks andmebaas ning kokku 79 tabelit. Õppeinfosüsteemiga tegeleb üks töötaja, seega jäävad ära kasutajaprobleemid, kasutajate gruppide loomine ning õiguste jagamine.

Süsteem on vajadusel uuendatav.

Käesoleva magistritöö raames vaadeldakse õppetöö moodulit. Õppetöö moodul struktureeritud puuna on esitatud lisas 3. Lühidalt esitades on õppetöö mooduli põhipunktid ilma alajaotuseta järgmised: õppurite register, tunniplaanid, õppekavad, õppematerjalid, tulemused, sessioonid, õppepraktika, sisseastumine, täiendkoolitus, aruanded.

VKHK õppeinfosüsteemil on lisaks veebipõhine väljund. Kogu järgnev informatsioon asub avalikul veebilehel. Võimalik on teha õppeinfo kohta järgmisi päringuid:

- õpilaste ja üliõpilaste nimekirjad;
- õppekavad;
- tunniplaanid;
- õppematerjalid;
- hinnetelehed;
- praktikakohad;
- õppe- ja eksamissessioonid (Võrumaa Kutsehariduskeskus, 2005).

Võrumaa Kutsehariduskeskuse õppeinfosüsteemi positiivseteks külgedeks on terviklik töötav lahendus, struktureeritus, arendatav, ühe isiku poolt hallatav, veebipõhine väljund kajastab kogu vajaminevat informatsiooni ning integreeritus RIKS-iga (Raamatukogude Info- ja Kataloogisüsteem).

1.2 Seadused

Järgneva alapeatüki eesmärk on anda ülevaade üliõpilasmooduli loomisel arvestatud seadustest. Isikuandmete töötlemine, hinnete avalikustamine, andmebaasile mõjuvad ohud ja nende vähendamiseks ning likvideerimiseks kasutatavad turvameetmed on probleemid, millele otsitakse käesolevas alapeatükis vastuseid.

Avaliku teabe seaduse (ATS) §37 lg 3 p 3 kohaselt on eraelulisteks isikuandmeteks isiksuseomadusi, võimeid või muid iseloomuomadusi kirjeldavad andmed (ATS, RTI, 30.11.2004, 81, 542, 10.12.2004). Antud kategooria alla tuleb paigutada ka üliõpilaste akadeemilist edasijõudmist kirjeldavad hinded, kuna nende kaudu saab hinnata isiku võimeid ning isiksuseomadusi (Tikk, 2003, lk 20). Seega on tegemist piiratud juurdepääsuga teabega ning avalikustada seda ilma isiku nõusolekuta ei tohi (*Ibid.*, lk 20).

Isikuandmete kaitse seaduse (IAKS) mõttes on eksamihinded mittedelikaatne teave, mille töötlemise lubatavust käsitleb §8 (IAKS RT 2003, 26, 158, 01.05.2004). Õppeasutused töötlevad oma üliõpilaste andmeid seadusega ettenähtud kohustuse täitmiseks, kuid sellisel alusel

töödeldavate andmete üleandmine kolmandale isikule (sellega on hõlmatud ka avalikustamine piiramata isikute ringile) on lubatud ainult juhul, kui nende töötlemine kolmanda isiku poolt toimub samadel eesmärkidel (Tikk, 2003, lk 20). Isikuandmete avalikustamise korral näiteks stendil või veebilehel ei ole aga võimalik kolmandate isikute poolt andmete kasutamise eesmärke kindlaks teha, seega ei tohi neid sel teel kättesaadavaks muuta. Seadusest tulenevalt ei tohi üliõpilase akadeemiline edasijõudmine olla igaühele jälgitav. Seadusega kooskõlas on hinnete väljapanemine kooli stendil või näiteks avalikustamine Interneti vahendusel juhul, kui isiku nimi asendada koodiga, õpinguraamatu numbriga vms. Kuna esitatud rakendus ei ole ligipääsetav mittevolitatud isikutele, siis ei teki vastuolu isikuandmete kaitse seadusega. Samas on käesolevasse moodulisse laiendamisvõimalust arvestades lisatud õpinguraamatu number, mille abil on võimalik üliõpilasel hiljem esitada vastavasisuline päring, näiteks isikliku hinnete lehe kuvamiseks.

Vastavalt isikuandmete kaitse seadusele § 5 on isikuandmete töötlemine iga isikuandmetega tehtav toiming, sealhulgas isikuandmete kogumine, salvestamine, korrastamine, säilitamine, muutmine, juurdepääsu võimaldamine, päringute teostamine, väljavõtete tegemine, kasutamine, edastamine, riskasutamine, ühendamine, sulgemine, kustutamine või hävitamine või mitu eeltoodud toimingut, sõltumata toimingute teostamise viisist või kasutatavatest vahenditest (IAKS RT 2003, 26, 158, 01.05.2004). Isikuandmete töötlemine on lubatud üksnes andmesubjekti nõusolekul (IAKS RT 2003, 26, 158, 01.05.2004). Kui üliõpilane soovib asuda õppima Lääne-Virumaa Kutsekõrgkooli, siis avaldust täites kinnitab ta allkirjaga esitatud andmete õigsust ning annab loa töödelda oma isikuandmeid käesolevas rakenduses.

Andmekaitse Inspektsiooni inspektor Bert Blösi sõnul on üliõpilase hinnete näol tegemist tavaliste isikuandmetega ning piisab kui:

- esitatud rakendus on kättesaadav asutuse volitatud töötajatele, kelle töölepingus on sätestatud konfidentsiaalsuskohustus;
- üliõpilaste hinded on näha õpinguraamatunumbri alusel ning üliõpilane näeb ainult oma õppetulemusi;
- andmete edastuseks kasutatakse turvalist HTTPS-protokolli.

Lisaks toonitas inspektor isikuandmetekaitse seaduse § 19 lg1 ja lg2 olulisust. Vastav paragrahv kohustab isikuandmete kaitseks kasutusele võtma organisatsioonilisi, füüsilisi ja infotehnilisi

turvameetmeid andmete tervikluse osas – juhusliku või tahtliku volitamata muutmise eest, andmete käideldavuse osas – juhusliku hävimise ja tahtliku hävitamise eest ning õigustatud isikule andmete kättesaadavuse takistamise eest ning andmete konfidentsiaalsuse osas – volitamata töötlemise eest (IAKS RT 2003, 26, 158, 01.05.2004).

Tagamaks isikuandmete kaitset peab vältima:

- kõrvaliste isikute ligipääsu isikuandmete töötlemiseks kasutatavatele seadmetele;
- andmete omavolilist lugemist, kopeerimist, salvestamist, muutmist ja kustutamist andmetöötlussüsteemis, samuti andmekandjate omavolilist teisaldamist;

ning tagama, et:

- igal andmetöötlussüsteemi kasutajal oleks juurdepääs ainult temale töötlemiseks lubatud isikuandmetele ja temale lubatud andmetöötluseks;
- isikuandmete edastamisel andmesidevahenditega ja andmekandjate transportimisel ei toimuks isikuandmete omavolilist lugemist, kopeerimist, muutmist või kustutamist (IAKS RT 2003, 26, 158, 01.05.2004).

Esitatud õppeinfosüsteemi üliõpilasmooduli loomisel lähtutakse lisaks isikuandmete automatiseeritud töötlemisel isiku kaitse konventsiooni ratifitseerimise seadusest. § 5 sõnastab automatiseeritult töödeldavate isikuandmete kogumise, täiendamise ning säilitamise eesmärgid:

- tuleb hankida ja töödelda ausal ning seaduslikul teel;
- tuleb koguda seaduspärasel ja täpselt määratletud eesmärkidel ning kasutada vastavalt nendele eesmärkidele;
- peavad olema adekvaatsed, asjakohased, piisavad vastavalt kogumise eesmärkidele;
- peavad olema õiged ja vajadusel täiendatavad;
- peavad olema säilitatud vormis, mis lubab andmesubjekti teha kindlaks mitte kauemaks, kui perioodiks, mis on vajalik andmete kogumise eesmärkide saavutamiseks (Isikuandmete automatiseeritud töötlemisel isiku kaitse konventsiooni ratifitseerimise seadus RTII, 03.01.2001, 1, 3, 01.06.2002).

Esitatud rakendust luues on arvestatud infosüsteemide kolmeastmelist etalon turbe süsteemi ISKE. Vastavalt Riigikogu poolt 28. augustil 2004 vastu võetud infosüsteemide turvameetmete süsteemi kehtestamise määruse §10 kohaselt on isikuandmete töötlemisel teabe konfidentsiaalsuse turvaosaklassiks S2 (Infosüsteemide turvameetmete süsteemi kehtestamine RTI, 26.08.2004, 63, 443, 29.08.2004). Teabe konfidentsiaalsus (S) on töödeldava teabe kättesaadavus ainult selleks

volitatud isikutele ja asutustele (Infosüsteemide turvameetmete süsteemi kehtestamine RTI, 26.08.2004, 63, 443, 29.08.2004). S2 – andmete avalikustamine häirib riigi või asutuse funktsioneerimist või rikub inimese privaatsust (*Ibid.*).

Käesolevas töös on arvestatud moodul B7.2 ehk andmebaasile mõjuvaid ohtusid ning rakendatud vastavaid turvameetmeid. Ohtude ning turvameetmete loetelu vastavalt ISKE juhendile on esitatud lisades 4 ja 5.

Järgnevas loetelus esitatakse olemaolevale rakendusele mõjuvad potentsiaalsed ohud ning rakendatud turvameetmed. Ohtude loetelu algallikas on ISKE rakendamisjuhend versioon 1.0 (Infosüsteemide kolmeastmelise etaloniturbesüsteem, 2003).

- Andmebaasi turvamehhanismide puudumine või puudulikkus (paroolid jms).
 - Andmebaasi turvamehhanismid on järgmised: arvuti ja serveri paroolkaitse, enne rakenduse kasutamist toimub kasutaja autentimine, algparoolid on muudetud ning lisaks toimub pidev andmebaasisüsteemi logifailide kontroll (kasutajate seansside ajad ja kestused, andmebaasiühenduste arv, nurjunud ühenduskatsed).
- Andmebaasi kasutajate vahetumise halb korraldus (baasi "lahtijätmine").
 - Kui kasutaja unustab lahkudes andmebaasi kasutajaliidest sulgeda või pole 5 minutit rakendust kasutanud, siis sessioonide abil suunatakse kasutaja uuesti sisselogimislehele.
- Andmebaasisüsteemi hooletu haldus (liigsed õigused, seire puudumine, harv varukopeerimine).
 - Kaks kasutajat saavad andmeid lisada ning muuta, liigseid õiguseid pole antud. Regulaarne andmevarundus. Varukoopiate loomine on andmete sisestamise faasis igapäevane protseduur, hiljem kord nädalas, vajadusel sagedamini (näiteks: eksamisessioonide ajal).
- Andmete kogemata manipuleerimine.
 - Enne programmi tegelikku kasutamist toimub vastavasisuline koolitus. Oskamatusest, teadmatusest, liigsetest õigustest ning hooletusest ei saa kasutaja andmebaasi rikkuda/kahjustada. Kasutajatel puuduvad vastavad paroolid, et ligi pääseda Apache veebiserveris olevale MySQL andmebaasile.
- Andmete kadu andmebaasis (kogemata, rünne).

- Andmebaasi tervikluse tagamiseks kasutatakse kasutaja autentimist sissepääsu reguleerimiseks, samuti pole võimalik sisestada kasutajal koodijuppe, üleliigseid tühikuid, sümboleid, lisaks kontrollitakse ka andmete sisestusel, et telefoninumber oleks õigel kujul, et e-mailiaadressis oleks @-märk ja punkt ning samuti isikukoodi korrektsust. Kogemata ei ole võimalik andmeid kustutada.
- Andmete kadu andmebaasis salvestusruumi puudumise tõttu.
 - Kõvakettamaht andmebaasis on 200 GB, seega andmete kadu ei saa tekkida salvestusruumi puudumise tõttu.
- Kaughoolde portide väärkasutus (kräkkerite ründeobjekt).
 - Serverile puudub ligipääs kaughoolde portide kaudu.
- Süstemaatiline paroolide mõistatamine (D. Kleini uuring 1990, 15000 kontot: 24,2%).
 - Paroolid on süstemaatiliseks mõistatamiseks liiga keerulised. Paroolides kasutatakse suuri ja väikeseid tähti, numbreid ning sümboleid, lisaks on paroolide minimaalne lubatud pikkus vähemalt 8 sümbolit.

Kui andmebaasi poole saab pöörduda võrgu kaudu, võivad kaugprotseduuride puudulikud turvameetmed võimaldada volitamatu manipuleerimist või sirvimist, rikkudes terviklust ja konfidentsiaalsust (Infosüsteemide kolmeastmelise etalonturbe süsteem, 2003, Lisad, lk 3).

Eelnev peatükk andis ülevaate tarkvara arendusetapi esimesest faasist – eeluuringust. Eeluuringu käigus otsiti vastuseid küsimustele: kuidas on loodud analoogsed toimivad süsteemid teistes kutseõppeasutustes (Võrumaa Kutsehariduskeskuses ja Tartu Kutsehariduskeskuses) ning kas ja millised õigusaktid reguleerivad üliõpilasmooduli loomisel tekkida võivaid probleeme? TKHK ja VKHK õppeinfosüsteemidega tutvumine andis ülevaate kasutatavatest rakendustest ning ideid ja informatsiooni käesoleva üliõpilasmooduli loomiseks, rakendamiseks ja edasiarenduseks. Vastavasisuliste õigusaktide läbitöötamine andis vajamineva informatsiooni tagamaks käesoleva rakenduse kooskõla kehtiva seadusandlusega.

2 Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi kavandamine

Käesoleva peatüki eesmärgiks on anda ülevaade üliõpilasmooduli kasutajatest ning nende vajadustest, nõuetest, olulistest parameetritest kasutajate jaoks, üliõpilasmooduli struktuurist ning kasutajaliidese olulistest omadustest. Tarkvara arendusetapina on käesoleva peatüki sisuks analüüs – täpsustatakse millistele nõuetele rakendus peab vastama. Uut süsteemi kavandades ning luues on tarkvara arendajal vaja kõigepealt aru saada probleemi olemusest, et loodav rakendus vastaks tulevaste kasutajate ootustele. Nõuete ebatäpne koostamine tähendab, et (Sherafat, 2005):

- tellijad ja lõppkasutajad pole süsteemiga rahul;
- süsteem on ebausaldusväärne;
- süsteemi edasine kasutamine, hooldus ja arendus muutub tavaliselt väga kulukaks.

Vältimaks olukorda, kus kasutajad pole rahul loodud rakendusega, on oluline uurida kasutajate vajadusi, hetkeolukorda ning nõudeid.

2.1 Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi üliõpilasmooduli kasutajate vajadused

Järgmises alapeatükis tutvustatakse LVKK ÕIS-i üliõpilasmooduli kasutajate vajadusi ning momendi olukorda. Esitatud magistritöö on oluline Lääne-Virumaa Kutsekõrgkoolile, kuna aitab kiirendada ja lihtsustada üliõpilasandmeid haldava personali tööd.

Momendi olukord

- Otsitakse paberkanalilt arvestuste ja eksamitulemuste protokolle ning iga kursusejuhataja esitab semestri lõpus aruanded, mis kajastavad üliõpilase keskmist hinnet ning õppekava täitmise protsenti.
- 2004/2005 õppeaasta kevadsemestril vaatas õppeosakond 600 täiendava õppetoetuse avaldust läbi, et kontrollida, kas nad elavad Vinni vallaga mittepiirnevates valdades.
- Hinnete protokollid täidetakse käsikirjas ning mitte alati tähestikulisel järjekorras.

- Üliõpilaste võlgnevustest selge ülevaate saamiseks on vaja läbi vaadata mitme semestri (kuni seitse) paberkandjatel asuvad protokollid.
- Lõpudokumentide andmed asuvad MS Exceli failis ning akadeemilised õiendid trükitakse MS Wordi hulgi-postituse abil.
- Statistikaamet küsib üliõpilaste vanuseid 01. septembri seisuga, see tähendab, et õppeosakond arvutab rohkem kui 1000 üliõpilase vanused nende avalduses oleva isikukoodi alusel.
- Samamoodi otsitakse paberitest inglise või saksa keelt esimese võõrkeelena õppivate üliõpilaste arvu ja mitmeid teisi statistikaametile vajalikke andmeid.
- Üliõpilaste andmed on laiali – paberitel, failides, erinevates arvutites.

Vajadused

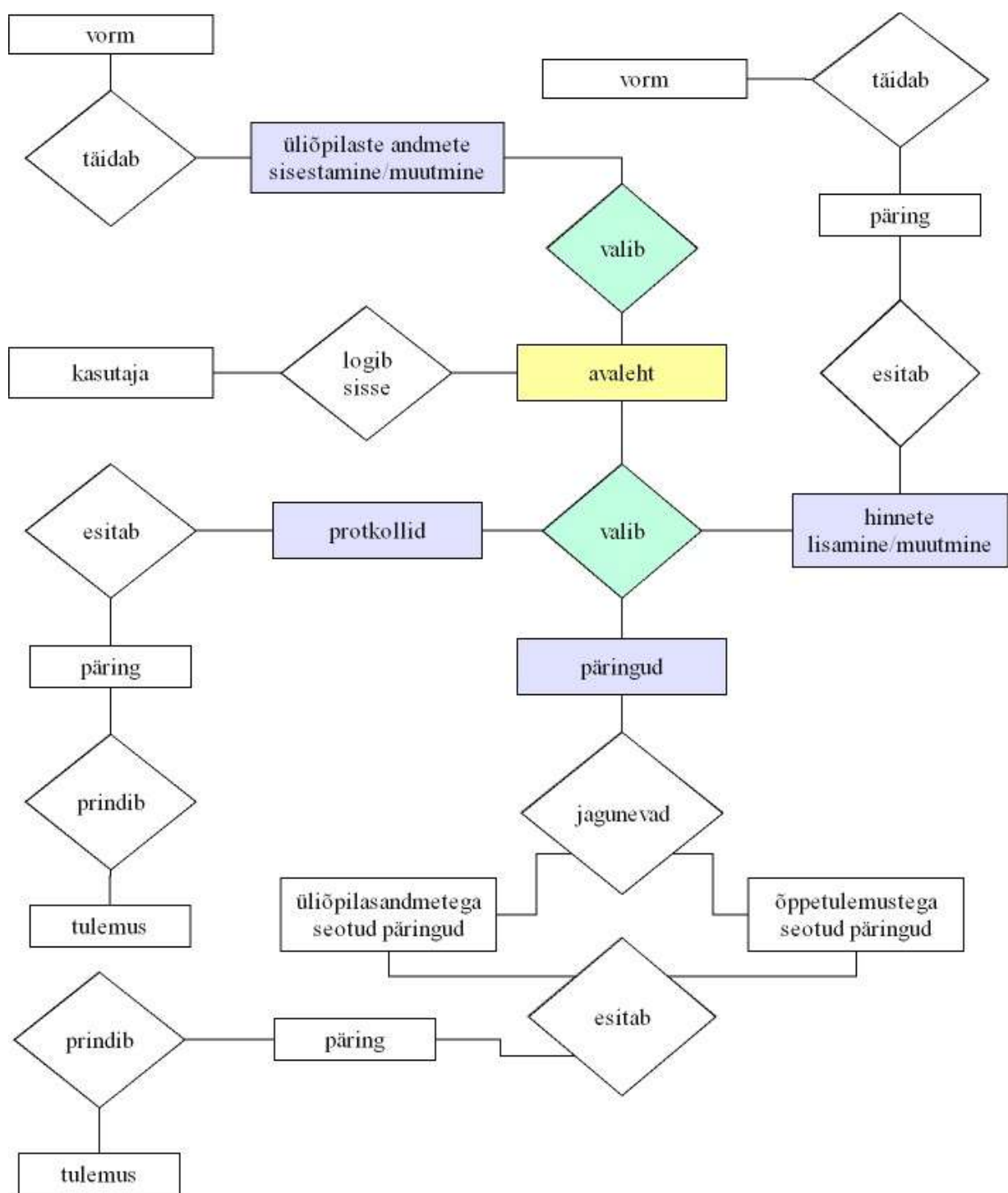
- Lihtsustada hinnete protokollide täitmist, kaotades ära kursusejuhatajate kohustuse esitada üliõpilaste keskmised hinded ja õppekava täitmise protsendi.
 - Muuta üliõpilaste andmed kiirelt kättesaadavaks selleks volitatud isikutele.
 - Koostöös tellijaga valmisid nõuded, missugused andmed on vajalikud.
- Vajaminevatest andmetest annab ülevaate alapeatükk 4.3.1 Päringute tegemine.
- Kergendada personali tööd suhtlemisel statistikaametiga.

Koostöös õppeosakonna ning direktori asetäitjaga valmis ÕIS-i kasutajate vajaduste loetelu, mis on struktureeritult esitatud joonisel 1.

Andmete sisestajale kuvatakse õige kasutajanime ja parooli trükkimise korral kasutajaliidese avaleht ning kasutajale esitatakse järgmised valikud:

- üliõpilaste andmete sisestamine;
- üliõpilaste andmete muutmine;
- protokollid;
- hinnete lisamine;
- hinnete muutmine;
- päringud.

Järgnevat joonist võib vaadelda kui õppeinfosüsteemi üliõpilasmooduli struktuuri, mis annab ülevaate tegevuste jadast.



Joonis 1. Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi üliõpilasmooduli struktuur.

Järgnev alapeatükk tutvustab käesoleva rakenduse kasutajaid, nende õigusi ja kohustusi rollide kaupa.

2.2 Kasutajad

Üliõpilasmooduli kasutajad on organisatsiooni töötajad, kes süsteemi poolt pakutavaid teenuseid kasutavad tööülesannete täitmiseks, informatsiooni kogumiseks, töötlemiseks ning väljastamiseks.

Käesoleva rakenduse loomisel on lähtutud Võrumaa Kutsehariduskeskuses rakendatud ideest, mida vähem õigustega kasutajaid, seda vähem probleeme, sellepärast on antud rakenduses kaks töötajat, kes saavad üliõpilasandmeid sisestada ja muuta ning 3 töötajat, kes saavad teostada päringuid.

Antud õppeinfosüsteemi kasutajad on Lääne-Virumaa Kutsekõrgkooli:

- süsteemi administraator;
- õppeosakonna juhataja;
- õppeosakonna sekretär;
- direktori asetäitja;
- juhiabi;
- direktor.

Täpsete rollide määramine eeldab kasutajate tuvastamist (autentimist) ja neile vajalike õiguste määramist (autoriseerimist). Vajalikud õigused ja piirangud on otstarbekas rühmitada kasutusvajaduste järgi – selliseid kasutusvajaduste gruppe nimetatakse rollideks (Rebane, 2003, lk 10).

Kasutajate õigused ja kohustused rollide kaupa:

1. Süsteemi administraator. Isik, kelle vastutada on süsteemi üldine haldamine. Süsteemi administraator saab määrata kõigi kasutajate õigusi ja volitusi.
2. Andemete sisestaja. Saab andmeid lisada, muuta, väljastada, ei saa kustutada. Õppeosakonna juhataja tegeleb päevase õppevormi üliõpilastega – sisestab ja muudab üliõpilase andmed ning hinded, teostab vajaminevaid päringuid, väljastab protokolle.

Õppeosakonna sekretäri tööülesannete hulka kuulub kaugõppe üliõpilaste andmete ja hinnete lisamine ning parandamine, protokollide väljastamine, päringute teostamine.

3. Kasutaja. Ei saa andmeid lisada, muuta, kustutada. Direktori asetäitja, juhiabi ja direktor saavad päringuid teostada, vajadusel väljastada.

Seega on esitatud rakendusel 2 kasutajat, kes saavad andmeid sisestada, muuta ja päringuid teostada. Ülejäänud 3 kasutajat – direktori asetäitja, juhiabi ja direktor – saavad teha päringuid, kuid ei saa andmeid lisada/parandada.

2.3 Olulised õppeinfosüsteemi parameetrid kasutajate jaoks

Käesolevas alapeatükis esitatakse õppeinfosüsteemi olulised parameetrid kasutajate jaoks ja need on järgmised:

- Kiirus – kui kiiresti kuvatakse kasutajale vajalik info (süsteemi elemendid peavad omavahel suhtlema nii vähe kui võimalik ja nii palju kui vajalik).
- Mugavus ehk käideldavus – teabe õigeaegne ning mugav kättesaadavus.
- Kujundus – kasutajaliidese väline disain.
- Turvalisus ehk terviklus – informatsiooni usaldusväärsus ning täpsus.
- Konfidentsiaalsus – andmete kättesaadavus ainult selleks volitatud isikutele.

Andmeterviklus on omadus, mis näitab, et andmeid ei ole volitamatul viisil muudetud ega hävitatud (Veebipõhine raamatukogusõnastik, 2005). Esitatud rakenduses on andmeterviklus oluline kriteerium, kuna peab olema välistatud näiteks olukord, kus üliõpilase hindeid on volitamatult muudetud. Käideldavus on samuti käesoleva üliõpilasmooduli oluline nõue. Käideldavus on omadus olla volitatud olemi nõudmisel kättesaadav ja kasutuskõlblik ehk omadus väljendub nende andmete õigeaegses ja hõlpsas kättesaadavuses volitatud isikutele (Eesti Standardiamet, 1999, lk 2). Näiteks üliõpilaste andmete sisestamiseks vajaminevad vormid ja päringu tulemused peavad olema kättesaadavad selleks volitatud isikutele. Töökindluse nõue tähendab ka sisuliselt teatud tasemel käideldavust. Konfidentsiaalsus on omadus, mis näitab, et informatsioon ei ole tehtud kättesaadavaks volitamata isikuile, olemitele või protsessidele ega neile avalikustatud (*Ibid.*, lk 2). Esitatud rakendus peab olema mitte kättesaadav volitamata

isikutele. Kuna antud hetkel on üliõpilasmoodul mõeldud asutuse töötajatele, siis üliõpilastel puudub ligipääs õppeinfosüsteemile.

2.4 Nõuded

Nõuete spetsifikatsioon on iga tarkvaraprojekti kohustuslik osa. Selle koostamise eesmärgiks on saada võimalikult täielik ülevaade tarkvarale esitavatest nõuetest enne programmeerimise algust. Nõuete analüüsi etapil püstitatakse disainiülesanded ning määratletakse sihtgrupp, tarkvaraplatvorm, tehnilised nõuded (Põldoja, 2003, lk 26). Nõuete spetsifikatsioon peab sisaldama võimalikult detailset kirjeldust, mida süsteem peab tegema ning ette andma parameetrid, millele süsteem peab vastama.

Nõuete kirjeldamisel võib eristada kolme tüüpi nõudeid (Haamer, 2001, lk 22):

1. Funktsionaalsed nõuded – tegevused, mida erinevad süsteemi kasutajad teha peaks saama.
2. Mittefunktsionaalsed nõuded – parameetrid, millele peab süsteemi installatsioon ja käitumine vastama: reaktsiooniajad, mahud, töökindlus, kasutatavus, efektiivsus, turvalisus, hooldatavus, laiendatavus ja usaldusväärsus.
3. Omadused, mida süsteemil ei tohi olla – tihti on nõudeid läbi eituse lihtsam kirja panna.

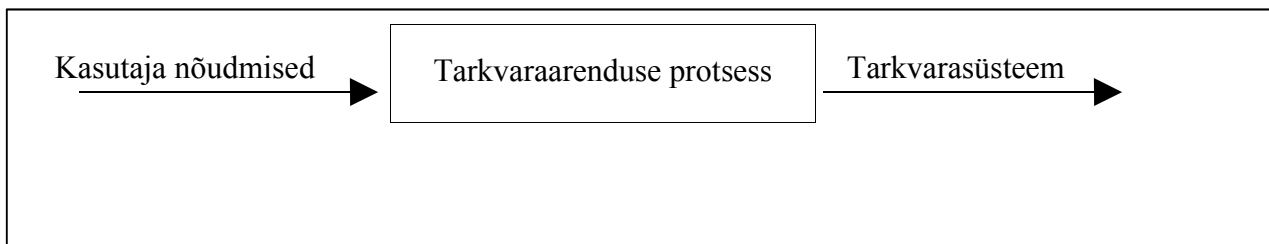
Nõuete dokumenti tuleb testida. Testida tuleb (*Ibid.*, lk 22):

- nõuete õigsust, vastavust tegelikele vajadustele;
- nõuete vahelist kooskõla;
- nõuete täielikkust, kas kõik on kirja pandud;
- reaalsust, kas nõudeid on võimalik täita.

Iga nõude kohta tuleks eraldi kontrollida (*Ibid.*, lk 23):

- kontrollitavust, kas hiljem on võimalik tõestada, et nõue on täidetud;
- arusaadavust, et poleks mitmeti tõlgendatavaid nõudeid;
- lokaalsust, muudetavust – kas nõuet saab muuta ilma suurt osa nõuete dokumendist muutmata.

Nõuete muutumine on väga sagedane tegevus, sest süsteemi nõuete koostamine ja ehitamine toimuvad pika aja jooksul. Selle aja vältel võivad muutuda väliskeskkond, seadused, kliendi arusaam oma vajadustest ja IT keskkond.



Joonis 2. Tarkvaraarenduse protsess.

Tarkvaraarendusprotsess koosneb hulgast tegevustest, mida on vaja läbida kasutaja vajaduste põhjal tarkvarasüsteemi väljatöötamiseks. Joonisel 2 on näidatud, et tarkvarasüsteemi aluseks on kasutajate nõuded/vajadused, mis realiseeritakse tarkvaraarenduse protsessi kaudu.

Nõuete dokumendi koostamisel on abistavaks dokumendiks IEEE standard, mis soovitab järgmist struktuuri (Sommerville, 2001, 117):

1. Sissejuhatus
 - 1.1. Nõuete dokumendi eesmärk
 - 1.2. Toote ulatus
 - 1.3. Definitsioonid, akronüümid, lühendid
 - 1.4. Ülevaade sissejuhatuses
2. Põhiosa
 - 2.1. Toote perspektiiv
 - 2.2. Toote funktsioonid
 - 2.3. Kasutaja karakteristikud
 - 2.4. Põhilised piirangud
 - 2.5. Eeldused ja ootused
3. Spetsiifilised nõuded sisaldavad funktsionaalseid, mitte-funktsionaalseid ja kasutajaliidese nõudeid. See on dokumendi kõige olulisem osa, kuid lähtudes organisatsioonide vahelistest erinevusest, ei esita standard kolmandas punktis kindlat struktuuri.
4. Lisad
5. Register

Kathryn L. Heninger'i poolt sõnastatud tarkvara nõuete dokument peab (Briggs, 1998):

- kirjeldama ainult välist süsteemi käitumist;
- andma ülevaate teostuse kitsaskohtadest;

- olema lihtne muuta;
- olema alusmaterjal süsteemi kasutajatele ja hooldajatele;
- sisaldama arenduse plaani;
- kirjeldama aktsepteeritavaid reaktsioone ebasoovitavatele sündmustele.

James ja Suzanne Robertson esitavad Volere malli, mille põhjal töötada välja nõuete spetsifikatsioon. Selle järgi jagunevad nõuded 27 erinevasse tüüpi, mis omakorda liigitatakse viide kategooriasse (Volere nõuete spetsifikatsioon, 2003).

Käesolevas rakenduses valmis nõuete dokument koostöös tulevaste kasutajatega ning koostamisel on arvestatud Volere mallis esitatud jaotust. Nõuete spetsifikatsioon sisaldab järgmisi punkte:

1. Projekti suunajad

- Rakenduse loomise eesmärk:

eesmärgiks on luua Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi üliõpilasmoodul.

- Kliendid, tellijad, huvirühmad, toote kasutajad:

Lääne-Virumaa Kutsekõrgkooli õppeosakond, direktori asetäitja, juhiabi, direktor.

2. Projekti piirajad

- Määratud piirangud (piirangud seoses tarkvaraplatvormiga, ajaga jne):

loodava rakenduse valmimisaeg on 01.juuni 2005. Kasutatakse vabavara: programmeerimiseks PHP, MySQL andmebaas, Apache veebiserver, operatsioonisüsteem SuSe.

3. Funktsionaalsed nõuded

- Funktsionaalsed ja andmetega seotud tegevused:

koostöös kasutajatega valmis tegevuste loetelu, mida erinevad süsteemi kasutajad teha peaks saama – missugune informatsioon peab olema kättesaadav selleks volitatud kasutajatele. Ülevaate vajaminevast informatsioonist annab alapeatükk 4.3.1 Päringud.

4. Mittefunktsionaalsed nõuded

- Rakenduse kasutajaliides:

kasutajaliidese loomisel on tähelepanu pööratud järgmistele märksõnadele: nähtavus, ennustatavus, järjepidevus, taastatavus. Lähemalt on selgitatud kasutajaliidese disaini olulisemaid soovitusi alapeatükis 2.5.

- Kasutatavus:

kasutatavuse kontrolliks rakendatakse vastavad testid. Põhjalikuma ülevaate annab alapeatükk 5.1.

- Rakendatavus:

tarkvara testimiseks kasutatakse sfinks.pri.ee serverit ning rakendatakse LVKK serveris.

- Turvalisus:

toimub kasutajate autentimine, paroolid on krüpteeritud, esialgu toimub rakenduse kasutamine sisevõrgus, edaspidi toimub andmete edastamine turvalise protokolliga HTTPS kaudu, olemasolevale üliõpilasmoodulile rakendatud turvameetmeid kirjeldatakse alapeatükis 1.4.

- Juriidilised / Õiguslikud nõuded:

käesoleva rakenduse loomisel on arvestatud nii isikuandmete kaitse seadusega, avaliku teabe seadusega, isikuandmete automatiseeritud töötlemisel isiku kaitse konventsiooni ratifitseerimise seaduse kui ka infosüsteemide kolmeastmelise etalonturbe metoodikaga. Põhjalikuma ülevaate annab alapeatükk 1.4.

5. Projekti nõuded

- Maksumus:

eraldi eelarve käesoleva rakenduse loomiseks puudub.

- Riskid:

riskidest annab ülevaate alapeatükk 3.3.

- Kasutajate koolitus:

enne olemasoleva üliõpilasmooduli rakendamist toimub kasutajate koolitus.

2.5 Kasutajaliides

Kasutajaliides (*user interface*) on süsteemiosa, mis aitab kasutajal arvutiga suhelda (W.M. Newman, M.G. Lamming, 1995, lk 6). Kasutajaliidese kujundamisel võib juhinduda järgmistest soovistest, näiteks (Sommerville, 2001, lk 330):

1. Õpitavus – kui lihtne on kasutajaliidest kasutama õppida

- Nähtavus ja ennustatavus – kasutaja näeb, kus ta asub ja saab hinnata võimalikke edasisi tegevusvariante (kõik võimalikud tegevused on ekraanil nähtavad ja valitavad).
- Sünteesitavus – kasutaja näeb oma eelmiste tegevuste tulemust.

- Tuttavlikkus – välimus sarnane varemkasutatud süsteemidega, visuaalsete objektidega saab teostada objekti tüübile loomulikke operatsioone (nupp – vajutada, liugur – nihutada).
- Üldistatavus – sarnastes situatsioonides annab sarnane käitumine sarnase tulemuse.
- Järjepidevus – ühtne stiil, värvid, nuppude asukohad, nimed, menüüd jne.

2. Paindlikkus

- Kasutaja initsiatiiv dialoogis.
- Paralleelsus – mitu dialoogi korraga.
- Kohandatavus ja iseadapteerumine.

3. Töökindlus – võime vältida kasutajate vigu

- Jälgitavus – näeb või saab küsida süsteemi olekut, süsteemi teated on järjekindlad (ei kao ära).
- Taastatavus – kasutaja vea korral *undo/redo* võimalus või võimalikult lihtne edasi parandamine.
- Reaktsioon – kiire tagasiside hiire ja klaviatuuri kasutamisele, stabiilne reaktsiooniaja pikkus.

Esitatud rakenduses on kasutajaliidese disainietapis tähelepanu pööratud järgmistele punktidele:

- Nähtavus ja ennustatavus – kasutaja saab igalt veebipõhiselt lehelt liikuda lingi abil avalehele, raadionuppude juures on selgitav tekst, kuhu liigutakse järgmisel lehel. Probleemi võib tekitada PDF-fail, kust tagasi saab minna *back*-nuppu kasutades.
- Järjepidevus – kasutatud on ühtset stiili, sama taustavärvi, nuppude nimesid ja asukohti.
- Taastatavus – näiteks veateate-lehelt tagasi tulles on sisestatud andmed alles, samuti korra *forward* või *back* nuppu kasutades.

Lisaks õpitavusele, paindlikkusele ja töökindlusele on kasutajaliidese disaini osas oluline tähelepanu pöörata ka värvide valikule. Järgnevas loetelus esitatakse olulisemad põhimõtted kasutajaliidese värviteooriast (Sommerville, 2001, lk 339):

- Värvide arvukus. Ühes aknas ei tohiks olla rohkem kui neli-viis erinevat värvi ning kasutajaliidesel mitte rohkem kui seitse. Värve peab kasutama valikuliselt ja järjepidevalt.

- Värvid kui süsteemi oleku näitajad ja teadete edastajad. Värvid võivad juhtida tähtsa informatsiooni juurde, aidata organiseerida veebilehe sisu ning hõlbustada lehel liikumist. Punane on veateade, roheline näitab, et võib edasi liikuda järgmisele lehele. Uus informatsioon esitatakse heledates toonides, oluline lühiteade intensiivsemates toonides.
- Värvide paar punane-sinine. Silma füsioloogia tõttu ei suuda inimesed üheaegselt vaadata punast ja sinist värvi.
- Taustavärvi valikul peab arvestama põhimõtet, et tekstivärv ja taustavärv oleksid omavahel kontrastis.



The image shows a web interface for 'Mõdriku'. At the top left is a logo with a building icon and the text 'Mõdriku'. To its right is a photograph of a large, light-colored building. Below these elements is a yellow rectangular box containing the text 'Palun sisestage kasutajanimi ja parool:' (Please enter username and password:). Under this text are two input fields: 'Kasutajanimi:' (Username:) and 'Parool:' (Password:). Below the password field is a button labeled 'Edasi' (Next).

Joonis 3. Näide kasutajaliideses kasutatavatest värvidest.

Joonisel 3 on näide käesoleva rakenduse kasutajaliidesest. Kasutatavad värvid on järgmised: taust helekollane, logo hall ja helesinine, kiri valdavalt must ja tumesinine, nupud hallid, veateated punased.

Eelnev peatükk andis ülevaate Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi kasutajatest, nende vajadustest, momendiolukorrast ja nõuetest, kasutajatele olulistest parameetritest, üliõpilasmooduli struktuurist ning kasutajaliidese kujunduseks etteantud soovitudest.

3 Tarkvaraarenduse teooria

Tarkvaraarendus omab tsüklilist elukäiku, mis algab rakenduse loomisega ning lõpeb rakenduse mahakandmisega ning uue loomisega. Klassikaline elutsükel koosneb kuuest faasist (nimetuste algallikas: Haamer, 2001, lk 27):

1. Eeluring – esimene faas, mis annab ideid ja informatsiooni analoogsetest rakendustest, reguleeritavast seadusandlusest, et luua uus süsteem efektiivne, töökindel ja paindlik laiendustele ning edasiarendustele.
2. Analüüs – teine etapp, kus tehakse kindlaks, millistele nõuetele peab rakendus vastama. Toimub kasutajate vajaduste, hetkeolukorra ning nõuete kaardistamine ja analüüs.
3. Tarkvara disain ja realiseerimine – kolmas etapp, kus suuremate rakenduste loomisel enne tarkvara realiseerimise disainitakse prototüüp – süsteemi projekteerimise, töönäitajate ja kasutuspotentsiaali mõistmiseks sobiv mudel või esialgne teostus (Prototüübi mõiste algallikas: Septer, Liikane, 2000, lk 348).
4. Testimine – neljandas faasis kontrollitakse süsteemi töökindlust ning kasutuslikkust.
5. Juurutamine – viiendas etapis toimub tarkvara rakendamine, algab andmete sisestamine ning rakenduse reaalne kasutamine.
6. Hooldus – viimases faasis süsteem töötab ning tehakse vajadusel jooksvaid hooldustöid ja muudatusi. Süsteemi eluea produktiivne osa.

Olemasolevas üliõpilasmoodulis läbitakse eelpool nimetatud tarkvaraarenduse etapid, kasutuselevõtt toimub funktsioonide kaupa. Esimeste tegevustena on võimalik üliõpilaste andmete sisestamine ja muutmine, õppetulemuste lisamine ja parandamine, päringute teostamine, järgmised töö etapid on: üliõpilaste ligipääs oma hinnetehtedele, õppekavade sisestamine, lõpudokumentide printimine ning üliõpilaste võlgnevuste päringu genereerimine vastavalt õppekavale ning üliõpilase sooritatud tulemustele.

Käesoleva peatüki eesmärgiks on anda ülevaade üliõpilasmooduli loomisel lähtunud tarkvaraarendusmetoodikast, projektide ebaõnnestumiste põhjustest, riskidest ning tarkvara isetegemise ja väljastpoolt tellimise plussidest ja miinustest.

3.1 Tarkvaraarenduse metoodika

Esitatud tarkvararakendus on valminud koostöös kasutajatega. Enne andmebaasi loomist hindasid tulevased tarbijad valmisolevaid vorme ning täpsustasid vajaminevaid andmeid, lisasid vajalikke andmevälju ning esitasid päringute loetelu. Tihe koostöö ja järjepidevad konsultatsioonid kasutajatega on aluseks, et järk-järgult ehitada vajaminevat ning kasutajasõbralikku rakendust. Käesolevas üliõpilasmoodulis kasutatakse tarkvara protsessi paremaks juhtimiseks mitme-etapilist mudelit, kus tegevused jaotatakse omavahel lõikuvateks etappideks. Töö iseloomu poolest toimub liikumine avastuslik-uurimuslikult tegevuselt (näiteks kasutaja vajaduste kindlakstegemine) loomingulisele (näiteks tarkvara arhitektuuri kujundamine) ja edasi rakenduslikule tegevusele (Normak, 2004, lk 80). Tarkvara üksikud komponendid realiseeritakse järjestikuliselt ja omavahel seotuna.

Mitme-etapilist mudelit kujutab järgmine tegevuste jada:

1. Analoogete rakenduste analüüs.
2. Nõuete väljatöötamine ja analüüs (üliõpilasmooduli struktuur).
3. Etapp 1: vormide loomine, koostöös kasutajatega vajalike andmeväljade lisamine ja andmebaasitabelite koostamine, prooviandmete sisestamine, testimine.
4. Etapp 2: koostöös kasutajatega vajaminevate päringute täpsustamine, päringute loomine, päringutulemuste testimine.
5. Etapp 3: õppekava sisestamiseks ja muutmiseks vajalike vormide loomine ning üliõpilaste võlgnevust ning hinnetelevi kajastava päringu loomine.
6. Kasutajate autoriseerimine – kasutajanimed, paroolid, turvalisuse testimine – kasutajaõiguste, kohustuslike väljade ning sisu kontroll (tühikud, koodijupid).
7. Testimine.
8. Tarkvara väljastamine.

Esitatud mooduli puhul on võimalik, kuid ei väljastata iga etapi valmimise järel tarkvara kasutajatele, kuna tegemist on mahuliselt väikese rakendusega ja puudub vajadus väljastada iga etapp üksikult. Ent mitme-etapiline koskmudel on antud projekt laiemas plaanis – üliõpilasmoodul kuulub teiste moodulite hulka Lääne-Virumaa Kutsekõrgkooli

õppeinfosüsteemis ning üliõpilasmooduli rakendamine leiab aset enne järgmiste moodulite kasutuselevõttu. Seega aitab antud tarkvaraprotsessi mudel vähendada riske – kasutaja saab tarkvara katsetada enne kogu õppeinfosüsteemi kasutuselevõttu, kuna programmi silumine toimub pidevalt ning välditud on olukord, kus projekt on 90%-liselt täidetud ja sellest 0% funktsioneerib.

3.2 Projektide ebaõnnestumiste põhjused

Järgnevas alapeatükis antakse ülevaade millistel asjaoludel võib projekti kulg viia ebaõnnestumisele. Ebaõnnestumiste põhjused tuleb võimalikult üksikasjalikult läbitöötada, sest ainult nii saab vähendada eksimuste tõenäosust. Projekti ebaõnnestumise põhjused on järgmised (Seeba, 2004, lk 11, Normak, 2004, lk 130):

- ebatäpne arusaamine lõppkasutajate vajadustest;
- võimetus ohjata muutuvaid nõudmisi;
- tarkvara, mida on raske hooldada või laiendada;
- tõsiste projekti vigade hiline avastamine;
- tarkvara kehv kvaliteet;
- testimine on ebapiisav;
- projekti seisu hindamine on subjektiivne;
- riskide lahendamine on pealiskaudne;
- kasutajate mittekaasamine;
- ressursside nappus;
- mitterealistlikud ootused;
- asutuse tippjuhtkonna mittepiisav toetus;
- tehnoloogiate mittevaldamine.

Käesolevas rakenduses pööratakse tähelepanu eelpool nimetatud projekti ebaõnnestumiste põhjustele ning püütakse vältida analoogseid olukordi. Tihe koostöö tulevaste kasutajatega aitab vältida olukordi, kus teostaja ei saa aru lõppkasutajate vajadusest või ei suuda toime tulla muutuvate nõudmistega. Kasutajate poolne testimine aitab avastada koodi- ja loogikavigu. Samas on oht, et projekti seisu hindamine on subjektiivne ja riskide haldus pealiskaudne.

3.3 Riskid

Esitatud üliõpilasmooduli arendamise riskid on teatava tõenäosusega aset leidvad sündmused, mis võivad arenduses või tema rakenduses aset leida. Andmebaasile mõjuvaid riske, see tähendab võimalusi, et vaadeldav oht kasutab ära mingi vara või vararühma nõrkused, põhjustades varade kaotuse või kahjustuse, on lähemalt selgitatud alapeatükis 1.2 Seadused, kus täpsemalt räägitakse ISKE metoodikaga kooskõlas olevatest riskide vähendamise võimalustest (Riskide mõiste algallikas: Eesti standardiamet, 1999, lk 7). Tabelis 1 esitatakse käesoleva üliõpilasmooduli planeerimise etapis valminud riskide tabel, kus esimeses veerus on riskide nimetused, teises mõjud ning kolmandas vähendamise võimalused.

Tabel 1. Riskid

Riski nimetus	Riski mõju	Riski vähendamise võimalused
Ajagraafikust mittekinnipidamine	Töö valmimine hilineb	Toimub järjepidev tegevuste analüüs lähtudes ajagraafikust
Riist- ja tarkvarast tingitud probleemid (rike, viirus, pöördumatud kahjustused)	Töö valmimine hilineb	Tagavarakoopiad, nende turvaline hoidmine ning pidev uuendamine
Väljastatud vigane tarkvara	Vead andmetöötlemises, autoriseerimises – ebausaldusväärne tarkvara	Tarkvara järjepidev testimine ja kohene vigade parandus
Ainuisikuline üliõpilasmooduli teostus	Töö valmimine hilineb, vähesest programmeerimiskogemusest tingitud vead tarkvara koodides	Koostöö kolleegide ja magistritöö juhendajaga
Suutmatus toime tulla muutuvate ning kasvavate vajaduste ja nõuetega	Töö ei valmi tähtaegselt, kiirustamisest, ning töökoormusest tingitud vead, k.a. loogikavead uute moodulite sidumisel olemasolevatega	Tööülesannete delegeerimine, et vähendada ainuisikulisest teostusest tulenevat töökoormust ja vastutust

Väiksemahulistes rakendustes puudub praktiline vajadus teostada riskianalüüsi iga tarkvaraarenduse etapi kohta eraldi. Käesolev rakendus valmis tihedas koostöös tulevaste kasutajatega, kes igapäevase töö käigus juhtisid teostaja tähelepanu erinevatele tarkvara protsessi etappide käigus asetleidnud ja tekkida võivatele probleemidele, vähendades seega riskide esinemistõenäosust. Eraldi tähelepanu on pööratud andmebaasile mõjuvate riskide ennetamisele ning vähendamisele, et vältida olukorda, kus andmebaasis olevad andmed on hävinenud. Riskide vähendamise üheks loetelu punktiks võib lugeda asutusesisese tarkvara isetegemise, kuna teostajal on väga selge ülevaade hetkeolukorrast, nõuetest ning kasutajate vajadustest. Järgnevas alapeatükis esitatakse tarkvara isetegemise ning väljastpoolt tellimise plussid ja miinused.

3.4 Tarkvara isetegemise ja väljastpoolt tellimise plussid ja miinused

Tarkvara isetegemine:

- lõpptulemus vastab rohkem varjatud vajadustele;
- hea kasutajatugi;
- kiire ja odav muudatuste teostamine;
- parem integreeritus teiste ettevõtte infosüsteemidega;
- võimalus arendusmeeskonda mõjutada (see võib olla positiivne kui ka negatiivne asjaolu);
- halvem tarkvara kvaliteet (ei dokumenteerita, kirjutatakse “põlve peal” oma vabast ajast);
- tööjõuga seotud riskid, tarkvara võib mitte olla renditav ja hallatav pärast võtmeisikute lahkumist;
- odav (oma töötajad teevad väljaspool tööaega, lisatasu).

Tarkvara väljastpoolt tellimine:

- tarkvara kvaliteet parem;
- teostajate kvalifikatsioon parem (tegijat saab valida);
- programmides vähem vigu;
- kulusid on võimalik täpsemalt planeerida;
- väiksemad tööjõuga seotud riskid (suures tarkvarafirmas leiab kiiresti asendaja);
- saab keskenduda ettevõtte põhitegevusele, mitte IT arendusele;

- muutused võtavad rohkem aega;
- partner ei tunne ettevõtte spetsiifikat;
- partneri kadumise risk (pankrot);
- töö hind on kallim.

Esitatud rakenduses ei kaalutud ei väljastpoolt ostetava valmisrakenduse ega tellimustöö võimalust. Valmistöö negatiivseteks külgedeks on kõrge hind ning 100% mitteühilduvus ettevõtte spetsiifikaga, tellimustöö puhul on määrav samuti suur maksumus.

Käesolev rakendus valmis magistritöö raames ning tarkvara isetegemise positiivsed ja negatiivsed küljed on järgmised:

- ettevõtte spetsiifika hea tundmine ning kasutajate vajaduste täpne arvestus;
- rakenduse testimis- ja samuti kasutusfaasis kohalolev ning kompetentne kasutajatugi (*help desk*);
- odav (esitatud rakendus valmis põhitoo kõrvalt);
- tarkvara koodide halb loetavus võõrale isikule, puudulik dokumentatsioon.

Käesolev rakendus on mõeldud eelkõige õppeosakonna töö kiiremaks ja lihtsamaks muutmiseks. Kokkuvõtteks võib öelda, et eelnev tarkvaraarenduse teooria annab ülevaate kasutatavast metoodikast ning võimalikest riskidest. Eespool käsitletud eeluuring ning kavandamine on eelduseks, et liikuda järgmise tarkvaraarenduse etapi juurde, milleks on rakenduse disain ja realiseerimine.

4 Tarkvara loomise tehniline tagapõhi

Käesolevas peatükis tutvustatakse klient-server rakenduse tööpõhimõtet ning vastatakse küsimusele miks eelistada tarkvara loomisel PHP/MySQL-i. Alapeatükk 4.2 annab ülevaate vormidest ning funktsioonidest, mida rakendatakse vormi kohustuslike väljade kontrollimisel. Neljandasse peatükki kuulub ka andmebaasi ja tabelite loomine, päringute tegemine ning protokollide genereerimine, samuti kasutaja autentimine, HTTPS ning kokkuvõtte töö realiseerimise kitsaskohtadest. Neljas peatükk annab ülevaate tarkvara loomise tehnilisest tagapõhjast.

4.1 Klient-server rakenduse tööpõhimõte ning PHP ja MySQL

Klient-server põhimõtet kasutatakse tänapäeva rakendustes üsna tihti. Üldiselt tähendab see rakenduse jaotamist kahte ossa:

- server, mis tegeleb andmehalduse, andmete säilitamise ja üldise töötlusega;
- klient, mis suhtleb vahetarkvaraga ja laseb tal andmeid töödelda, edastades päringud serverile.

Kliendi põhifunktsioonid on (Majak, 2005):

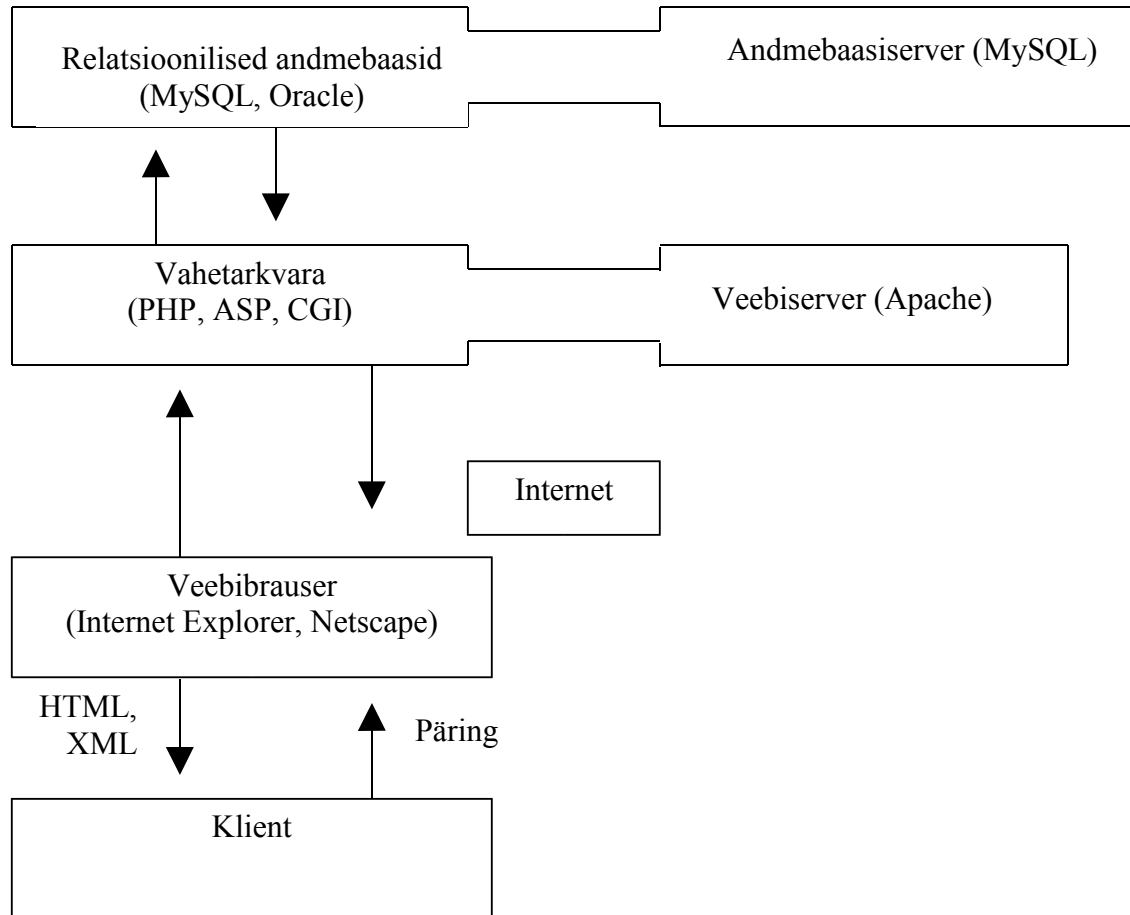
- kasutajaliidese hoidmine;
- kasutajalt andmete vastuvõtmine ja andmebaasi(desse) saatmine;
- serveri teenindust (serveri andmebaasi kasutust) nõudvate protsesside käivitamine, nende nõuete (*requests*) saatmine serverile;
- tulemuste vastuvõtmine serverilt;
- tulemuste teisendamine kasutaja jaoks sobivasse vormi.

Serveri põhifunktsioonid on (Majak, 2005):

- klientidelt saabuvate teenindusnõuete (*requests*) vastuvõtmine;
- teenindusnõuete töötlemine (andmebaasi päringud, andmeuuendus);
- tulemuste teisendamine kliendi jaoks sobivasse vormi ning nende ülekandmine kliendile;
- andmete õigsuse ja andmebaasi terviklikkuse kontroll;

- andmetele juurdepääsu ja kasutajate õiguste kontroll.

Järgnev joonis annab ülevaate klient-server rakenduse põhimõttest.



Joonis 4. Klient-server rakenduse tööpõhimõte.

(Greenspan, Bulger, 2001, lk 22, Castagnetto, Rawat jt, 2001, lk 693)

Joonisel 4 esitab klient päringu veebisirvijale. Päringute teostamiseks veebiserveri ja andmebaasiserveri vahel on vaja programmeerimiskeelt, mida nimetatakse vahetarkvaraks. Vahetarkvaraks nimetatakse kaht või mitut tüüpi tarkvara (nt rakenduse ja operatsioonisüsteemi) vahelist tarkvara (Hanson, Tavast, 2000, lk 169). Vahetarkvara vajatakse mõlemas suunas liikumiseks, nii andmebaasiserverile mineva kui ka andmebaasiserverist tuleva info töötlemiseks. Andmebaasiserver salvestab kõik andmed ja täidab vajalikud päringud. Selline on üldine klient-server rakenduse tööpõhimõte. Õppeinfosüsteemi rakenduse printsiip on oma olemuselt analoogne.

Võimalusi realiseerimiseks on mitmeid, nagu ASP (*Active Server Pages*) ja Java, antud lahenduseks on valitud PHP (*Hypertext Preprocessor*) ja MySQL (*My Structured Query Language*). Nimetus 'PHP' pärineb keele kõige esimesest versioonist PHPst, mida nimetati *Personal Home Page Tools* ja hiljem *Personal Home Page Construction Kit*; praegu tõlgendatakse lühendit PHP ka kui *Hypertext Preprocessor* (Ilitson, 2004).

PHP kuulub klassi vahetarkavara. Vahetarkvara töötab tihedas koostöös veebiserveriga, et tõlgendada HTTP (*Hypertext Transfer Protocol*) tehtavad päringud, töödelda neid ja teha päringute täitmiseks koostööd teiste programmidega serveris. PHP on HTML-i lisatud (*embedded*) serveripõhine (erinevalt brauseripõhisest keelest nagu Javascript) skriptikeel. Enamus keele koostisosadest on laenatud mujalt (C, Java, Perl). PHP on laialt kasutatav ning avatud lähtekoodiga, mis on eelkõige mõeldud veebirakenduste koostamiseks (*Ibid.*).

PHP käsud lisatakse harilikule HTML-veebilehele eraldi PHP-märgenditena, mis võivad paikneda ükskõik kus veebilehe tekstis ning täidetakse veebiserveris ja brauserisse saadetakse lõpptulemus. PHP abil on võimalik koguda infot, genereerida dünaamilisi (kasutaja tegevusele vastavalt muutuvaid) veebilehti ning saata või vastu võtta "küpsiseid" (cookie; veebiserveri poolt kasutaja arvutisse salvestatav väike andmekogum, mida harilikult kasutatakse lehe olekute meeldejätmiseks) (*Ibid.*).

PHP suurimaks plussiks on võimalus kirjutada väga kiirelt ja lihtsalt andmebaasiga seotud veebilehti.

Andmebaasina kasutatakse firma MySQL AB poolt arendatud MySQL andmebaasi. MySQL andmebaasi server on maailmas enim kasutusel olev avatud lähtekoodiga andmebaasiserver (*Ibid.*). Tarkvara on oma arhitektuuri ülesehituse poolest kiire ja see on kergesti kohaldatav.

Kokkuvõtvalt võib esitada järgmised põhjendused:

Miks eelistada PHP-d?

- toetab erinevaid andmebaase – MySQL, Oracle, ODBC;
- sisseehitatud raamatukogud – näiteks PDF, GIF genereerimiseks, elektronposti saatmiseks jne;
- vaba tarkvara;
- lihtsalt ümber paigutatav – toetab erinevaid operatsioonisüsteeme ja on kergesti üleviiday;
- kiiresti ja pidevalt arenev (Majak, 2005).

Miks eelistada MySQL-i?

- kõrge tootlikkus, kiire ja võimas – väikese jõudlusega server suudab teenindada miljoneid rakendusi;
- vabavara;
- mitmeplatvormiline;
- pidevalt arenev (Majak, 2005).

Negatiivse poole pealt võib esile tuua järgmised punktid:

- kogu töö toimub serveris, ka veatöötlus tuleb seal realiseerida. Valesti täidetud vorm peab minema serverisse ja sealt tagasi kliendini;
- turvalisusega seotud probleemid;
- viitetervikluse kontroll puudub – tavaliselt andmebaasi tarkvaras kontrollitakse automaatselt kirje lisamisel, kas võõra võtme väljale salvestatud väärtus on olemas teises tabelis, millele ta peab viitama. Kui teises tabelis vastavat võtme väärtust pole, siis ei lasta antud tabelisse kirjet lisada. MySQL-i andmebaasis selline viiteterviklikkuse kontroll puudub;
- MySQL ei toeta otseselt valiku lauses alamvalikuid.

PHP ja MySQL koos moodustavad kiire süsteemi relatsiooniliste andmebaaside kasutamiseks.

PHP võimaldab suhelda lihtsalt MySQL-ga, luua ühendust serveriga (*mysql_connect*), ühendust sulgeda (*mysql_close*), teha serveris päringuid (*mysql_query*) ja kasutada päringu tulemusi (*mysql_result*, *mysql_fetch_row*) jne (Sibola, 2000, lk 38). Mugav on ka kliendiga suhtlemine, nimelt on kõik kliendi poolt saadatud vormi muutujad PHP poolt eeldefineeritud kui programssed muutujad.

Näiteks kui vormil sisaldub tekstiväli nimega eesnimi

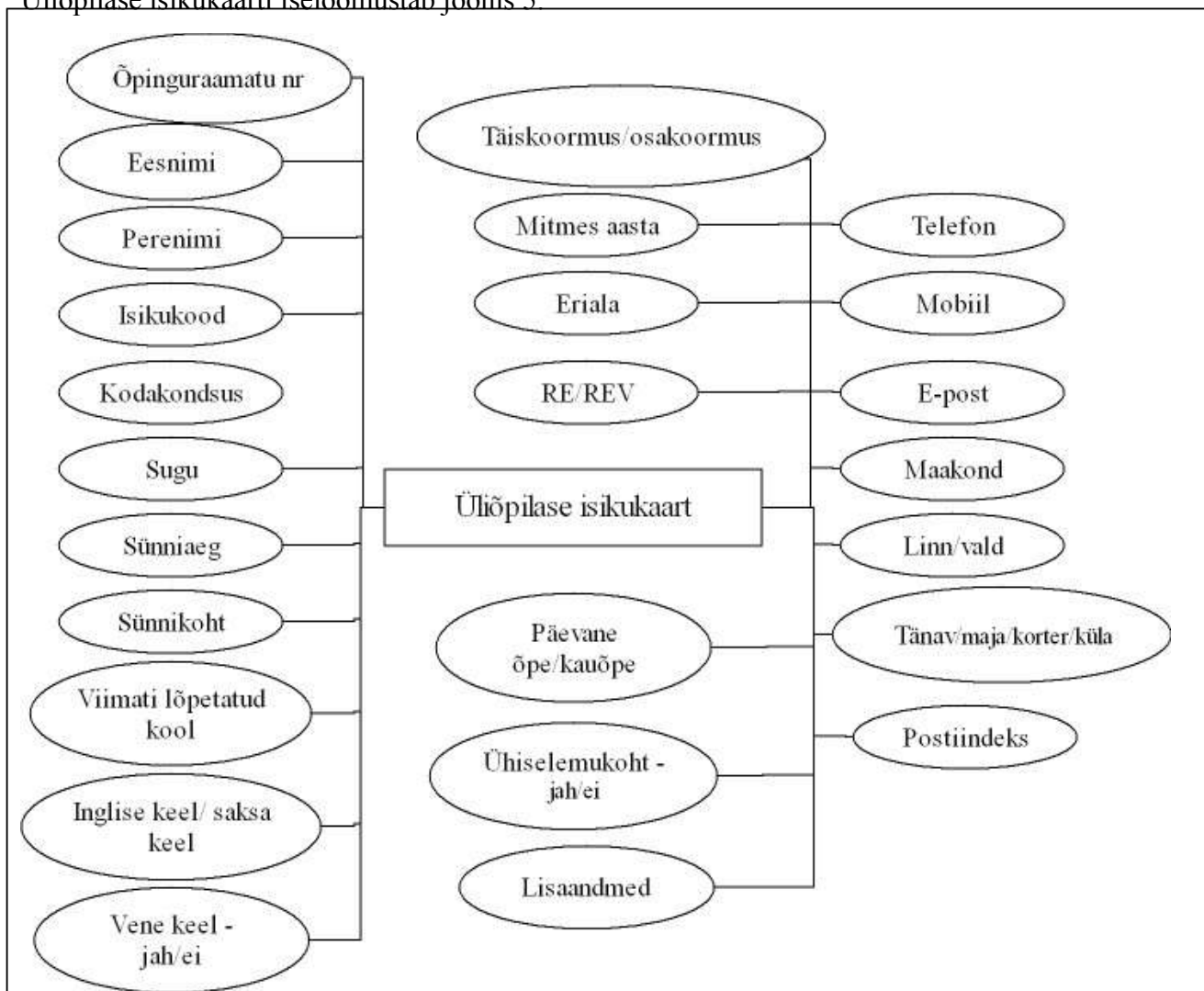
```
<input type="text" name="eesnimi">
```

ja klient saadab selle PHP skriptile, siis võib skript selle vormi muutuja väärtuse poole pöörduda lihtsalt muutuja *\$eesnimi* kaudu.

Kokkuvõtteks antud alapeatüki kohta võib öelda, et kliendi osa nagu vormi tegemine, mis eeldab HTML-i (*HyperText Markup Language*) tundmist ning andmete edastamine kliendilt serverisse, kus vajalik on teada PHP-d, toimub suhteliselt lihtsa tööpõhimõtte järgi, samas töötava klient-server rakenduse realiseerimine on keerulisem osa.

4.2 Vorm ja vormi kohustuslike väljade kontrollimine

Üliõpilase isikukaarti iseloomustab joonis 5.

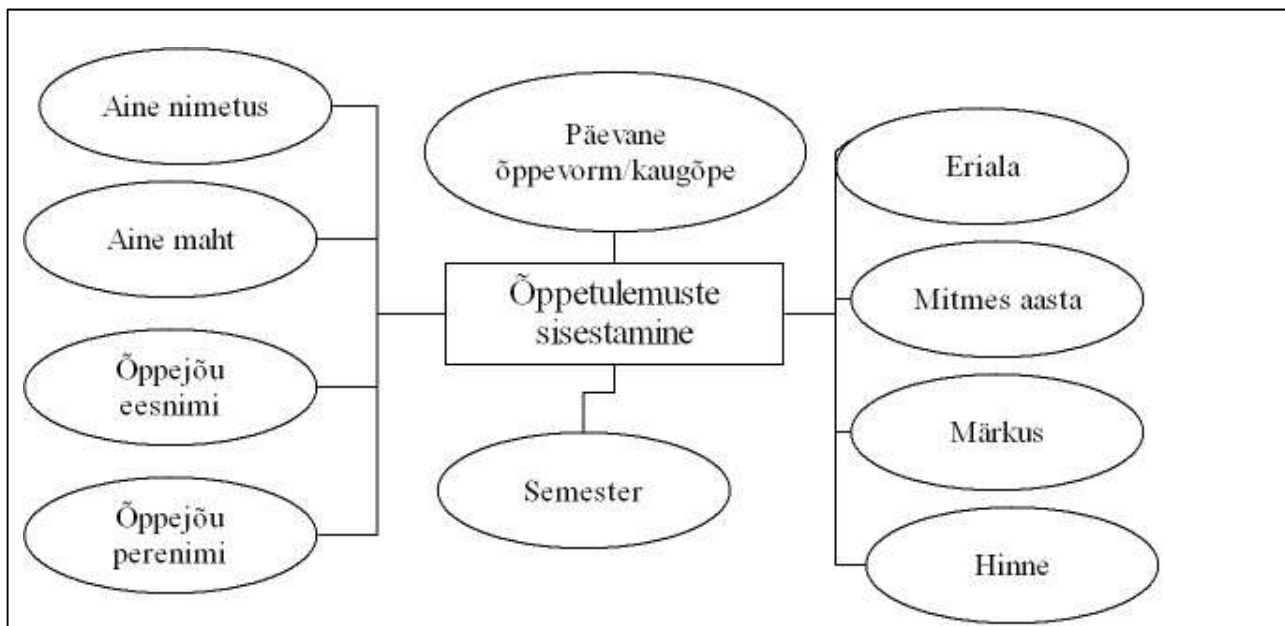


Joonis 5. Üliõpilase isikukaart.

Vastuvõtu avalduse põhjal kantakse andmed andmebaasi. Veebipõhisele üliõpilase isikukaardile sisestatakse: õpinguraamatu number, eesnimi, perenimi, isikukood, kodakondsus, sugu, sünniaeg, sünnikoht, viimati lõpetatud kool, telefon, mobiilnumber, e-maili aadress, maakond, linn/vald, tänav/maja/korter/küla, postiindeks, eriala, mitmes aasta, riigieelarveline/riigieelarveväline (RE/REV), päevane õpe/kaugõpe, täiskoormus/osakoormus, põhivõõrkeel – inglise keel/saksa keel, vene keelt õppinud – jah/ei, vajab ühiselamukohta – jah/ei ning lisaandmed. Viimasesse lahtrisse saab sisestada üliõpilase kohta täiendavaid andmeid.

Klikk nupul “Sisesta” saadab sisestatud andmed serverisse, kus kontrollitakse PHP abil andmete õigsust (mõistlikkust). Kui andmed on korras, siis salvestatakse need andmebaasi.

Õppetulemuste sisestamist iseloomustab joonis 6. Õppetulemuste sisestamiseks trüüb andmete sisestaja aine nimetuse, aine mahu, õppejõu ees- ja perekonnanime, valib semestri ning esitab kursuse päringu (eriala, mitmes aasta, päevane õppevorm või kaugõpe).



Joonis 6. Õppetulemuste sisestamine.

Klikkides nuppu “Edasi”, avaneb hinnete sisestamise vorm. Antud vormil kuvatakse kasutajale valitud kursuse nimekiri tabelina. Andmete sisestaja lisab igale üliõpilasele rippmenüü abil hinde ja vajadusel kirjutab märkuse.

Vormi kohustuslike väljade kontrollimine

Selleks, et andmebaasi ei salvestataks poolikut ja ebakorrektset informatsiooni, on vajalik kohustuslike lahtrite kontroll. Vormi kohustuslike väljade täitmise kontrollimine on võimalik nii *JavaScripti* abil kui ka PHP-d kasutades. *JavaScripti* eeliseks on kiirus ja mugavus, kontrollimine toimub kliendi arvutis. Kuna paljud kasutajad on oma arvutis keelanud *JavaScriptide* kasutamise, seda eelkõige turvalisuse huvides ning samuti ei lase mõned tulemüürid rakendada kõiki *JavaScripte*, siis võimalike probleemide ennetamiseks ja vältimiseks kasutatakse antud

rakenduses väljade kontrollimiseks PHP abi, kuigi see vähendab töökiirust, sest väljade kontroll toimub serveris.

Antud rakenduses kuuluvad kohustuslikule täitmisele järgmised väljad: õpinguraamatu number, eesnimi, perenimi, eriala, mitmes aasta ja õppevorm. Selleks, et kindlaks teha, kas kasutaja on sisestanud teksti nõutud lahtritesse, on mugav kasutada funktsiooni *empty()* (Greenspan, Bulger, 2001, lk 124).

Kontrollimiseks, kas andmete sisestaja on sisestanud üliõpilase perenime, kasutatakse lauset:

```
if (empty ($perenimi))
```

```
echo "Sisesta perenimi, see on nõutud v&auml;li";
```

Olukorra vältimiseks, kus kasutaja sisestab nõutud lahtrisse vaid tühiku, kasutatakse antud rakenduses funktsiooni *trim()*. Funktsioon *trim()* on vajalik tühikute eemaldamiseks stringi algusest ja lõpust. Lisaks tühikutele eemaldatakse ka stringi algusest ja lõpust reavahetused (*\n*), rea algusesse liikumised (*\r*), horisontaalsed tabulatsioonid (*\t*), vertikaalsed tabulatsioonid (*\v*) ja stringi lõpu tunnused (*\0*). Funktsiooni *trim()* süntaks on järgmine:

```
string trim ( string str [, string charlist])
```

(Veebipõhine PHP käsiraamat, 2005).

Programmikoodis kontrollitakse tühikute sisestamist järgmiselt:

```
elseif (empty ($eesnimi) || trim($eesnimi)=="")
```

```
$viga.="Sisesta eesnimi, see on nõutud v&auml;li";
```

Telefoni ja isikukoodi sisestamise korral kontrollitakse ka lahtrite sisu, kas lahtrite sisu koosneb lubatud sümbolitest - numbritest, vastasel korral esitatakse veateade. Samuti toimub e-posti aadressi kontroll, siis on nõutavaks sümboliks @-märk, millest ees- ja tagapool võivad olla tähed, numbrid, sidekriips, alakriips ja punkt. Sümbolite kontrollimiseks on vajalik funktsioon *ereg()*, mis on regulaaravaldisele vastava alamstringi otsimiseks stringist ning antud funktsiooni süntaks on:

```
int ereg(string pattern, string source, array [regs]);
```

(Castagnetto, Rawat jt, 2001, lk 218).

```
Int ereg(string reg_avaldis, string lähtestring, array [vasted]);
```

Lähtestringist otsitakse regulaaravaldise abil kirjeldatud alamstringi. Viimaseks argumendiks on leitud tulemuste ehk vastete massiiv, mille iga element on üks leitud alamstring. Funktsioon *ereg()* sarnaneb funktsiooniga *eregi()*, ainsaks erinevuseks on asjaolu, et *eregi()* ei ole tõstetundlik (*ereg()* on tõstetundlik).

Telefoninumbri kontrollimiseks on vajalik koodi kirjutada järgmised laused:

```
elseif (($telefon!="") && !ereg("[0-9]+$",$telefon))
```

```
$viga.="Telefoninumber võib sisaldada ainult numbreid 0-9, telefoninumber on n&aiteks  
kujul: 04420789";
```

Kui nõutud väli pole täidetud või on sisestatud mittelubatud sümbolid, ilmub veateade, mis avatakse uuel lehel ning kust tagasi saab klient siis, kui klikib lingil “Tagasi”. Tulles tagasi andmete sisestamise vormile, jätkab klient väljade täitmist, varem sisestatud andmed on alles.

Näide:

```
if ($viga!=""){  
echo $viga;  
$a = "isikukaart.php";  
$a.="?eesnimi=". $eesnimi . "&perenimi=". $perenimi;  
if(session_is_registered("sess_isiku"))  
{ $sess_isiku["viga"]=$viga;  
$sess_isiku["a"]=$a;  
header("Location: viga1.php");  
exit();}
```

Eelpool näites olevad muutujate väärtused kirjutatakse kokku URL-i (*Uniform Resource Locator*) reale. Lisaks kasutatakse muutujate väärtuste hoidmiseks ning vealehele liikumiseks ja pealehele tagasitulekuks sessioonide ja *header()* funktsiooni abi. Sessioonidest annab ülevaate alapeatükk 4.4. Esitatud rakenduses kasutatakse *header()* funktsiooni, et liikuda vajalikule vormile. Koodi kirjutamisel peab jälgima, et enne funktsiooni *header()* poleks ühtki rida HTMLi. Kood algab PHP sissejuhatava taagiga.

Koodinäide *header()* funktsiooni süntaksist:

```
<?php  
header("Location: http://www.example.com/"); /* Brauseri suunamine */  
exit; /*Tagab, et edasisuunamise korral järgnevat koodi ei täideta */  
?>
```

(Veebipõhine PHP käsiraamat, 2005).

Antud rakenduses on vajalik kirjutada koodi algusesse järgnev lõik:

```
<?php
if ($submit=="Edasi")
header("Location:".$vorm);?>
```

Vastavalt raadionupu valikule kuvatakse kasutajale ette soovitud vorm. Valides üliõpilaste andmete sisestamise, saab õppeosakonna töötaja täita üliõpilase isikukaarti.

Muutujate väärtuste kirjutamisel URL-i reale on oht, et kui kasutaja sisestab mitmest sõnast muutuja, siis vealehelt tagasi tulles on andmed alles kuni esimese tühikuni ja kõik ülejäänud väärtused on kadunud. Eelpool kirjeldatud olukorra vältimiseks on funktsioon *urlencode()*, mille süntaks on järgmine:

```
string urlencode ( string str)
```

(Veebipõhine PHP käsiraamat, 2005).

Funktsioon *urlencode()* kodeerib stringi URL-ile loetavaks. Tähtedest ja numbritest erinevad sümbolid (*non-alpha-numeric*), v a ‘_’, asendatakse protsendi (%) märgiga, millele järgneb kaks 16-süsteemis numbrit ja eraldajaks on pluss (+) märk. Näiteks tühiku asemel on %20, pluss-märgi asemel %2B jne.

Programmikoodi kirjutatakse sümbolite kodeerimiseks järgmine lause:

```
$a."&telefon=" . urlencode($telefon) . "&epost=" . urlencode($epost);
```

Selleks, et sisestatud andmed oleks alles vormi väljadel, on vaja koodi kirjutada järgmine lause:

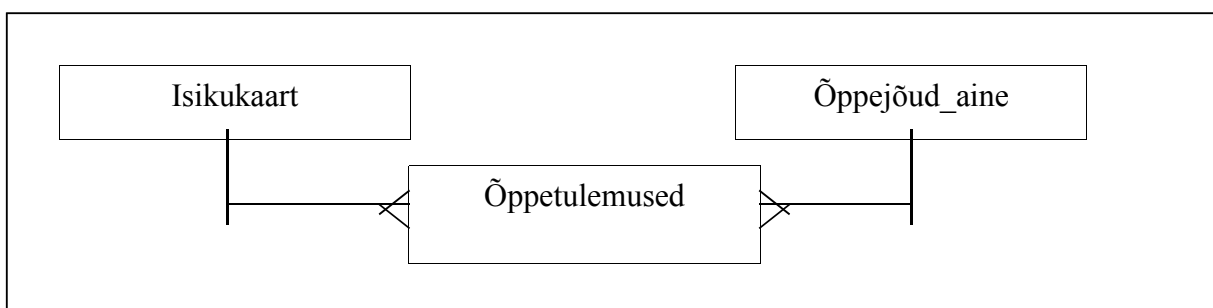
```
<input name="perenimi" type="text" value="<?php if ($viga!="") echo $perenimi; ?>">
```

Vormi vastavate lahtrite täitmist kontrollitakse peale seda kui andmete sisestaja on klikkinud nupul “Sisesta”. Kui kõik vajalikud lahtrid on täidetud, salvestatakse informatsioon andmebaasi.

4.3 Andmebaas ja päringute tegemine

Õppeinfosüsteemi üliõpilasmooduli tabelitest annab ülevaate andmemudel. Andmemudel kirjeldab andmebaasisüsteemis kasutatavat andmestruktuuri (Kivisild, 2002, lk 26). Lisas 6 esitatud õppeinfosüsteemi üliõpilasmooduli andmebaasi andmemudelisse kuulub kolm tabelit. Esitatud andmemudeli pealkirjadeks on välja nimi, kirjeldus, täitmiskohustus, tüüp ning võtmed. Tabelitele kuuluvad atribuudid on esitatud tulbas, mille pealkiri on välja nimi. Järgnevas tulbas on esitatud atribuutide kirjeldused, seejärel on oluline märkida täitmiskohustus. Igal atribuudil on

vastav tüüp, näiteks tekstiväli, kuupäev või mõni muu väljatüüp. Viimaseks tulbaks on võtmed/võõrvõtmed. Võti on oma olemuselt unikaalne identifikaator, tunnus, mille väärtus määrab üheselt kõik objektid. Võti võib olla primaarvõti või võõrvõti. Vastavalt ISO/IEC 2382-17 andmebaasi standardile on primaarvõti võti, mis üheselt identifitseerib ühe kirje ning võõrvõti relatsiooni atribuut, mis vastab teise relatsiooni primaarvõtmele (Eesti Standardiamet, 1998, lk 13). Esitatud tabelitel on võtmeks järjekorranumber, mis üheselt identifitseerib iga sisestatud kirje. Andmetabelite vahel on üks mitmele seos – ühele kirjele lähtetabelis vastab üks või mitu kirjet sihttabelis. Lähtetabel on tabel, mis omab võõrvõtit, sihttabel on tabel, millele võõrvõti viitab (Kivisild, 2002, lk 25). Joonisel 7 on näha, et ühele kirjele Isikukaardi tabelis vastab üks või mitu kirjet Õppetulemuste tabelis ning ühele kirjele Õppejõud_aine tabelis vastab üks või mitu kirjet tabelis Õppetulemused.



Joonis 7. Tabelite vahelised seosed.

Andmebaasi “infosysteem” kuulub kolm tabelit: “isikukaart”, mis koosneb 27 väljast, “hinnesis”, kuhu kuulub 5 välja ning “oppejoud_aine” – 9 välja. Tabelite võtmeks on ID, mis on andmetüübilt *auto numeric*.

Järgnevalt tutvustatakse funktsioone, mis on vajalikud ühenduse loomiseks, andmebaasi ja tabelite moodustamiseks, andmete sisestamiseks ja päringute tegemiseks.

Enne kui tööle asuda andmebaasiga, on vaja luua ühendus ning autoriseerida kasutaja. Ühenduse loomiseks on funktsioon *mysql_connect()*

```
int mysql_connect ([string $hostname] [,string $username] [,string $password])
```

(Koterov, 2001, lk 364).

Antud rakenduses näeb ühenduse loomine välja selline:

```
$yhendus=mysql_connect("localhost","eve","123456"), mis tähendab, et ühendust võetakse lokaalse arvutiga ning registreerimine näitab, et kasutajanimi on “eve” ja parool “123456”.
```

Kui ühendus on olemas, on järgmine oluline tegevus andmebaasi loomine, selleks sobib funktsioon: *mysql_create_db()*

```
int mysql_create_db(string database name, int [link_identifier] );
```

(Castagnetto, Rawat jt, 2001, lk 296).

Andmebaas, mille nimi on “infosüsteem”, luuakse järgmise lause abil:

```
$abi=mysql_create_db ("infosysteem")
```

Andmebaasi kasutamiseks on vaja luua tabel, et oleks koht, kuhu andmeid vormilt saata.

Tabel luuakse järgmise lausega:

```
$abi1="create table isikukaart
```

```
(eesnimi varchar (20) null,
```

```
perenimi varchar (20) null,)";
```

Klient-server rakenduses sisestatakse andmed vormi väljadele ja kirjutatakse PHP abil kood, mis võtab need muutujad serveris vastu ning kirjutab andmebaasi tabelisse.

MySQL-is on andmete lisamise süntaks järgmine:

```
insert into TabeliNimi (veerunimi1, veerunimi2...) values (väärtus1,väärtus2)
```

(Koterov, 2001, lk 371).

Andmed lisatakse tabelisse järgmise lause abil:

```
$abi2="insert into isikukaart (ID,eesnimi, perenimi) values ('$_POST [ID]', '$_POST [eesnimi]','$_POST[perenimi]'";
```

Juhtsümbole, PHP-koodi, *JavaScriptide* ja HTML-i andmebaasi kirjutamise vältimiseks kasutatakse enne salvestamist funktsioonide *addslashes()*, *strip_tags()* ja *htmlspecialchars()* abi. Funktsioon *addslashes()* lisab stringis teatud sümbole ette kaldkriipsu. Antud funktsiooni süntaks on:

```
string addslashes ( string str)
```

(Veebipõhine PHP käsiraamat, 2005).

Andmete salvestamisel MySQL-i andmebaasi on oht, et teatud kasutaja poolt sisestatud sümboleid võib tõlgendada juhtimissümboolitena (eritähenduslikult). Selle vea vältimiseks kasutatakse funktsiooni *addslashes()*. Antud funktsioon lisab kaldkriipsu (\) apostroofi (‘), jutumärgi (“), tagurpidi kaldkriipsu (\) ja NUL (NULL-baidi) ette.

Esitatud rakenduses olev koodinäide funktsioon *addslashes()* kasutamise kohta on järgmine:

```
$eesnimi=addslashes($eesnimi);
```

Hiljem andmebaasist andmete sisselugemisel peab kõrvaldama lisatud täiendavad kaldkriipsud, selleks kasutatakse funktsiooni *stripslashes()* abi. Funktsioon *htmlspecialchars()* konverteerib spetsiaalsed märgid HTML-le sobilikuks. Teatud sümbolitel on eriline tähtsus HTML-is ja nende sümbolite tähenduse säilitamiseks kasutatakse funktsiooni *htmlspecialchars()*.

Teisendatakse järgmisi sümboleid:

'&' (ampersand) muudetakse '&';

""" (jutumärk) muudetakse '"';

" (apostroof) muudetakse ''';

'<' (väiksem kui) muudetakse '<';

'>' (suurem kui) muudetakse '>';

(Veebipõhine PHP käsiraamat, 2005).

Funktsiooni süntaks on:

```
string htmlspecialchars ( string string [, int quote_style [, string charset]])
```

(Veebipõhine PHP käsiraamat, 2005).

Sümbolite teisendamiseks kirjutatakse koodi järgmised laused:

```
$eesnimi=htmlspecialchars($eesnimi);
```

```
$perenimi=htmlspecialchars($perenimi);
```

Funktsioon *strip_tags()* kõrvaldab HTML-i ja PHP taagid sisestatud väljadelt. Seega kasutajal pole võimalik sisestada HTML-koodi, PHP-d ja *JavaScript*e.

Funktsiooni süntaks on:

```
string strip_tags ( string str [, string allowable_tags])
```

(*Ibid.*, 2005).

Funktsioon *strip_tags()* aitab muuta antud rakenduses HTML-i, *JavaScripti* ja PHP elemendid ohutuks ehk keelata nende kasutamine. Programmikoodi on kirjutatud järgmised laused:

```
$eesnimi=strip_tags($eesnimi,"");
```

Käesoleva alapeatüki lõpetuseks võib öelda, et andmebaas ja tabelid on loodud, andmed on tabelitesse lisatud, järgmise osana on vaja genereerida sisestatud andmete põhjal päringud, et vajaminev informatsioon oleks kasutajatele kättesaadav.

4.3.1 Päringud

Kasutajal on võimalik teha üliõpilasandmete ja õppetulemustega seotud päringuid, lisaks kasutatakse päringuid protokollide loomisel ning hinnete lisamisel/muutmisel.

Käesoleva rakenduse kõige olulisemaks osaks on päringud. Vajaminev informatsioon peab olema kättesaadav volitatud isikutele ühest kohast.

Koostöös tellijaga valmis järgmine loetelu vajaminevatest päringutest:

- Hinded
 - Üliõpilase keskmine hinne (aritmeetiline)
 - Semestri kaupa
 - Kõikide hinnete keskmine
 - Kursuse keskmine hinne (aritmeetiline)
 - Semestri kaupa
 - Kõikide hinnete keskmine
- Õppekava täitmise protsent
 - Üliõpilase õppekava täitmise protsent
 - Kursuse õppekava täitmise protsent
- Täiendav toetus
 - Kursuste kaupa, kes elavad väljaspool Vinni valda ja Vinni vallaga piirnevaid valdu
 - Erialade kaupa, kes elavad väljaspool Vinni valda ja Vinni vallaga piirnevaid valdu
- Üliõpilaste arv
 - Naissoost üliõpilaste arv
 - Meessoost üliõpilaste arv
 - Koolis õppivate üliõpilaste arv
 - Akadeemilisel puhkusel viibivate üliõpilaste arv
 - Päevases õppevormis
 - Kaugõppes

- Kursuste kaupa
- Üldarvuna kokku
- Päevases osakonnas õppivate üliõpilaste arv
- Kaugõppes õppivate üliõpilaste arv
- Üliõpilaste arv erialade kaupa
- Üliõpilaste arv maakondade kaupa
- Ühiselamus elavate üliõpilaste arv
- Mitte ühiselamus elavate üliõpilaste arv
- Inglise keelt esimese võõrkeelena õppivate üliõpilaste arv
 - Erialade kaupa
 - Üldarvuna
- Saksa keelt esimese võõrkeelena õppivate üliõpilaste arv
 - Erialade kaupa
 - Üldarvuna
- Tabelina vanuste kaupa (sünniaastate kaupa)
 - Päevane õppevorm
 - Kaugõpe
- Riigieelarveliste ja riigieelarveväliste üliõpilaste arv
 - Kursuste kaupa
 - Erialade kaupa
 - Lääne-Virumaalt
 - Ida-Virumaalt
 - Üldarvuna

Päringu tulemused esitatakse kasutajale PDF-formaadis. Protokollide loomiseks ning päringutulemuste kuvamiseks kasutatakse FPDF abi, et genereerida PDF-faile. FPDF on mittekommertslikul eesmärgil kasutatav vabavara, mis võimaldab kasutajal genereerida PHP faile PDF-failideks.

Dokumendi formaat PDF (*Portable Document Format*) on firma Adobe Systems poolt loodud universaalne faili formaat, mis säilitab sõltumata kasutatavast platvormist ja versioonist (nii riistvaralisest kui tarkvaralisest) algdokumendi esialgse väljanägemise – säilib esialgne kirja pilt,

vorming, graafika ja värvid. Seega tagatakse lehekülje alati ühesugune väljanägemine ja paigutus nii ekraanil kui paberile trükituna, seejuures ei sõltu trükitulemus kasutatava printeri tüübist. PDF-faili lugemistarkvara on tasuta, failid on väikesemahulised, fail on alati ühesuguse kujuga. PDF on lõppformaad, seega mõeldud info esitamiseks mitte redigeerimiseks (Helinurme, 2003). PDF-formaad sisaldab mugavaid navigeerimisvahendeid, dokumendi kaitsmise võimalusi, PDF-fail on kompaktne, ülekantav erinevatesse süsteemidesse ja töökindel (Septer, 2001, lk 68). Esitatud rakenduses kasutatakse FPDF abi protokollide ning päringutulemuste printimiseks või salvestamiseks.

4.4 Kasutaja autentimine ning HTTPS

Vältimaks mittevoolitatud isikute ligipääsu andmetele on vajalik kasutaja autentimine. Antud alapeatükk annab ülevaate autentimisest ja vajaminevatest funktsioonidest.

Autentimine on olemi (isiku, grupi, protsessi, sõnumi vms) väidetava identsuse automaatne verifitseerimine (Hanson, V, Buldas, A, jt, 1998, lk 58). Autentimist rakendatakse peamiselt:

- pääsu reguleerimise mehhanismides (taotleja autentimine);
- kommunikatsiooni tervikluse tagamise mehhanismides (andmeallika või partneri autentimine) (*Ibid.*, lk 58).

Autentimise protsessi tulemuseks on kasutaja identsuse kontrollimine ja tõendamine. Antud rakenduses on vajalik ühepoolne autentimine, mida rakendatakse pääsu reguleerimise mehhanismides. Mehhanism kontrollib pääsu taotleja autentsust, taotleja aga ei kontrolli mehhanismi autentsust ehk siis usaldab mehhanismi “ametlikku ilmet” (Hanson, V, Buldas, A, jt, lk 62, 1998). Vajadus kasutaja autentimise järele tuleneb sellest, et osa infost, mis arvutisüsteemis liigub, peab olema kaitstud volitamata kasutamise eest. Autentimine tähendab tegevust, kus üks osapool tõestab teisele oma identiteedi. Kasutaja autentsuse tagamiseks on kõige levinum viis kasutada paroolide (salasõnade) süsteemi. Parool on volitatud kasutaja käsutusse antud salastatud tekstijada, mis võib üldjuhul sisaldada lisaks suur- ja väiketähtede ka numbreid ja erimärke (Hanson, V, Buldas, A, jt, lk 66, 1998). Kasutaja tuvastab ennast serverile, sisestades veebilehel kasutajanime ja parooli. Server kontrollib sisestatud info õigsust ning vastavalt kontrolli tulemustele garanteerib kasutajale ligipääsu või keeldub sellest ning teavitab ebaõnnestunud sisselogimiskatses kasutajat ennast ning vajadusel ka serveri haldajaid. Kasutaja seisukohast on

oluline, et autentimine oleks lihtne ja põhitööd vähe segav tegevus. Paroolide kasutamine paraku selle seisukohaga väga hästi kokku ei sobi, paroolid ei tohi olla liiga lühikesed ega liiga lihtsad ning neid tuleb vahetada.

Antud rakenduses on vajalik kasutaja autentimine, et vältida mittevoolitatud isikutele andmete sisestamist.

Enne kui avaneb vorm, kus kasutaja saab valida protokollide, päringute, hinnete lisamise/muutmise ning üliõpilaste andmete sisestamise/muutmine vahel, toimub kasutaja autentimine. Töötaja peab sisestama vastava vormi väljadele oma kasutajanime ja parooli. Sisestatud andmete õigsuse korral avaneb kasutajale vastav vorm, vale kasutajanime või parooli korral kuvatakse siseneda soovijale veateade.

Turvalisuse huvides on väga oluline paroolide krüpteerida. Üheks võimaluseks on kasutada selleks funktsiooni *md5()*, mis teeb suvalise pikkusega tekstist krüptograafilise lühendi, nn hash lühendi, hash on 32-märgiline heksadetsimaalne number. Funktsioon *md5()* on algoritm informatsiooni krüpteerimiseks, kasutatakse näiteks paroolide kontrollimiseks või failide saatmisel kogu faili kohale jõudmise kinnitamiseks. Funktsioon *md5()* on ühesuunaline.

Näiteks:

md5 “parool” on 69fb46f4c18463dd25002aeffc0257d1

Algoritmi süntaks on järgmine:

String md5 (string str)

(Veebipõhine PHP käsiraamat, 2005).

01. märts. 2005 kirjutasid Arjen Lenstra, Xiaoyun Wang ja Benne de Weger raporti, kus nad väitsid, et suudavad genereerida kaks 1024-bitise pikkusega bitistringi, mille MD5-hashid kattuvad (Lenstra, A, Wang, X, Weger, B, 2005). Seega on tegemist kollosiooniga – on leitud kaks avateksti, mille tulemuseks on ühesugune krüptograafiline sõnumilühend. Samas mõlemad stringid genereeriti katse käigus, seega ei ole tegu päris MD5 pööramisega, kuid võib väita, et pragu *md5()* turvalisuses siiski on tekkinud. Kui funktsioon *md5()* genereerib 32-märgilise heksadetsimaalse numbri, siis funktsioon *sha1()* väljastab 40-märgilise heksadetsimaalse numbri (Veebipõhine PHP käsiraamat, 2005). Käesolevas rakenduses kasutatakse krüpteerimiseks *sha1()* funktsiooni ning koodinäide andmete paroolitabelisse kirjutamise kohta on järgmine:

\$abi=sha1('salajane');

```
$abi4="insert into parool (kasutajanimi,parool)  
values ('nimi',password('$abi'))";
```

Parool “salajane”, mida krüpteeritakse *sha1()* funktsiooniga, omistatakse abimuutujale “abi”. Seejärel lisatakse tabelisse “parool” väljad kasutajanimi ja parool ning antakse neile väljadele väärtusteks “nimi” ja muutuja “abi” väärtus, kusjuures viimane veel omakorda krüpteeritakse kasutades MySQL-i funktsiooni *password()*. Eelmises versioonis MySQL 4.0 oli funktsiooni *password()* väljund 16 sümboli pikkune jada. Versioonis 4.1 aga muutus see 41 sümboli pikkuseks (Veebipõhine MySQL käsiraamat, 2005).

Paroolivormil kontrollitakse sisestatud andmete õigsust. Kui kasutaja klikib nuppu “Edasi”, siis muutujale „parool“ omistatakse funktsiooniga *sha1()* krüpteeritud parool ning valitakse tabelist “parool“, kus parool on omistatud funktsiooniga *password()* krüpteeritud paroolile ning kasutajanimele omistatakse muutuja “nimi”. Eelnev selgitus näeb koodis välja järgmine:

```
if ($submit=="Edasi" )  
{ $parool=sha1($parool);  
$abi5 = "select *from parool where parool = (password('$parool')) and  
kasutajanimi = '$nimi'";  
$tulem = mysql_query($abi5);  
if ($tulem) { list($admin_ok) = mysql_fetch_row($tulem); }
```

Õige parooli ja kasutajatunnuse sisestamise korral kuvatakse kasutajale avaleht. Vältimaks olukorda, kus teades faili nime, oleks võimalik avada autentimata erinevaid vorme, on kõikidele kasutaja identifitseerimist nõudvatele vormidele lisatud programmikoodi tingimus, mis väldib mittevolitatud isikutel andmete sisestamise ja edastamise.

Koodinäide:

```
if (!session_is_registered("admin_ok")){  
    header("Location: paroolivorm.php");  
    exit();}
```

Eelneva koodiosa realiseerumine kuvab kasutajale, kes soovib avada näiteks andmete muutmiseks olevat vormi, teades õiget failinime, ette sisselogimise akna.

Antud rakenduses liigutakse paroolivormi lehelt valiku vormile sessioonide abil. Järgnev tekstilõik tutvustab sessioonide vajalikkust ja kasutamist.

Sessioonide abil saab "kaasas kanda" väärtusi, mis on antud mingite toimingute sooritamisel. Sessioonid on info salvestamiseks mitme WWW (World Wide Web) lehekülje vahetamisel või

sama lehekülje uuendamisel. Sessiooni abil hoitakse salvestamiseks vajalikku infot serveris tekstifailis. Kliendil on ainult sessiooni number, mille järgi vastav fail identifitseeritakse.

Kõigepealt määratakse sessiooni nimi, mida tahetakse registreerida. Näiteks:

```
session_register("admin_ok");
```

Sessiooni alustamiseks on vaja järgmist koodilauset:

```
session_start();
```

Sessioonide puhul on oluline ka see, et enne registreerimist ei tohi olla mitte ühtegi rida HTML-i, küll aga võib olla PHP-koodi, kus puuduvad funktsioonid *print()* või *echo()*.

Sessiooni numbrit hoitakse cookie abil (seda pole vaja kasutajal teha, salvestatakse automaatselt) või siis URL-is (?järel). Sessiooni number on tegelikult üks juhuslik arv kodeeritud kujul. Sessiooni lõpetamisel kustutatakse sessiooni numbrile vastav tekstifail.

Olemasoleva üliõpilasmooduli kasutamine toimub asutuse sisevõrgus, kuid edaspidi hakatakse andmete turvaliseks edastamiseks kasutama käesolevas rakenduse edasiarenduses HTTPS-protkoll. HTTPS on krüpteerimis- ja autentimisvahendeid sisaldav laiendatud HTTP protokoll, sideprotokoll firmalt Netscape, mis on sisse ehitatud Netscape'i brauseritesse ning võimaldab juurdepääsu turvatud veebiserveritele (E-teatmik, 2005). Kui internetiaadressis seisab `http://` asemel `https://`, siis suunatakse sõnum veebiserveri pordile 443, mitte pordile 80 nagu tavaliselt ning seanss toimub üle turvasoklite kihi (SSL) (*Ibid.*). See tähendab, et kogu andmevahetus kasutaja ja veebiserveri vahel toimub krüpteeritult ning brauser dekrüpteerib serverist saabuval veebilehed enne kuvamist (*Ibid.*). HTTP – *HyperText TransferProtocol* – hüperteksti edastuse protokoll ehk standardite kogum, mis võimaldab World Wide Webi kasutajatel veebilehekülgedel informatsiooni vahetada (Septer, Liikane, 2000, lk 198). SSL – *Secure Socets Layer* – programmikiht, mis haldab võrgu kaudu edastavate andmete turvalisust (*Ibid.*, lk 382). HTTPS ehk HTTP üle SSL ühendus erineb tavalisest HTTP ühendusest selles osas, et sõnumi sisu on krüpteeritud vastava sessiooni transpordivõtmega. Sessiooni loomisele eelneb hulk sõnumeid klientprogrammi (veebilehitseja) ja serveri vahel, mille käigus lepatakse kokku näiteks kasutatavas protokollivariandis ja transpordivõtmes. HTTPS enamlevinud variandid on SSLv3 ja TLSv1 (Sinivee, *i.a.*). Konkreetse HTTPS versiooni, kasutatava krüptoalgoritmi, transpordivõtmed ja sertifikaadid lepivad veeebilehitseja ja veebiserver ühenduse alguses kokku

(*Ibid.*). Server saadab veebilehitsejale oma sertifikaadi, kasutaja saab vaadelda serveri sertifikaati ja kontrollida, kes selle välja andnud on (*Ibid.*). Kui kasutaja ei usalda seda väljaandjat, siis ta katkestab ühenduse.

Tavalise klient-server rakenduse korral võtab kasutaja arvutis töötav klientprogramm ühendust serveriga. Läbi avaliku võrgu kulgev liiklus on avalik – ründaja võib jälgida kasutaja ja serveri vahelist andmevahetust ja/või muuta edastatavaid andmeid. Turvaliseks ühenduseks luuakse privaatvõrgu tunnelid üle avaliku võrgu.

Esitatud rakendus hakkab paiknema Lääne-Virumaa Kutsekõrgkooli Apache veebiserveris, kus operatsioonisüsteemiks on SuSE Linux 9.2. (Süsteemi töökindluse tagavad AMD Opteron 242 1,6 GHz 1MB L2 cache, 2 GB ECC).

Nii Apache kui ka SuSe on avatud lähtekoodiga (*Open Source*) tarkvarad. See tähendab, et programmi lähtetekst on kõikidele soovijatele vabalt kättesaadav ning huvilised võivad tarkvara arenduses kaasa aidata.

Apache serverit populaarseks tegevad omadused on (Sibola, 2000, lk 10):

- multiplatvorm – Apache serverit saab edukalt kasutada suvalisel platvormil suvalises operatsioonisüsteemis;
- toetab täielikult HTTP/1.1 protokoll;
- võimaldab väga võimsat konfigureerimist ja laiendamist;
- võimaldab lisada mooduleid laiendamaks serverit; selleks saab kasutada spetsiaalset liidest Apache module Application Programming Interface (Apache API);
- kogu algkood on avalik ja litsentsitingimused ei piira selles muudatuste tegemist;
- võimaldab piisavat kasutajatuvastuse ja õiguste mehhanismi;
- kohandatavad serveripoolsed veateated;
- mitme veebilehe ühes serveris hoidmine;
- omab väga võimsat logi süsteemi, mis on vastavalt vajadustele kohandatav;
- areneb pidevalt edasi.

Kokkuvõtte töö realiseerimise kitsaskohtadest

Eelnev peatükk andis ülevaate tarkvaraarenduse kolmandast etapist – tarkvara tehnilisest teostusest. Seega on nüüd võimalik üliõpilaste andmeid ning õppetulemusi sisestada ja muuta ning leida vajaminev informatsioon päringute abil.

Tehnilise teostuse etapis oli vaja leida vastuseid järgmistele küsimustele:

- Kui kursusele on hinded sisestatud, siis samal semestril ei tohi sama õppeaine hindeid enam uuesti sisestada, vaid sel juhul saab ainult muuta juba lisatud hindeid.

Eelmise probleemi lahendusena kontrollitakse, kas tabelis `oppejoud_aine` on olemas kirje, mille väärtused on samad valitud aine nimetuse, eriala, kursuse (mitmes aasta), õppevormi ja semestriga. Kui on, siis kuvatakse kasutajale teade: “Valitud kursusele on hinded juba lisatud!”. Kui mitte, siis liigub kasutaja hinnete sisestamise vormile.

- Kuidas üliõpilase andmed viia järgmisele kursusele, et ei peaks käsitsi üliõpilase kaupa muutma kursuse aastat?

Valiku “Kursuse üleviimine järgmisele kursusele” abil toimub andmete uuendus. Kasutaja esitab päringu kursuse kohta ning valib rippmenüü abil millisele kursusele üleviimine toimub (I, II, III, IV) ning vajutab nuppu Muuda. *Update()* funktsiooni abil uuendatakse andmed.

- Kui üliõpilane lõpetab kooli, siis kuhu liiguvad tema andmed?

Eelnevaga samas rippmenüüs on võimalus valida: Lõpetanud üliõpilased. Tehes eelpool kirjeldatud valiku, lisatakse üliõpilaste andmed uude tabelisse, mille nimi on lõpetanud.

Järgmine tarkvaraarenduse etapp on testimine, milles pööratakse tähelepanu kasutatavuse testimisele – kui lihtne on olemasolevat üliõpilasmoodulit kasutada.

5 Testimine

Tarkvara testimine on tarkvaratootmise protsessi üks osa. Testimise eesmärk ei ole üksnes avastada tarkvarast vigu, kontrollida tarkvara toimimist vastavalt püstitatud nõuetele ja kontrollida tarkvara töökindlust selleks ettenähtud keskkonnas, vaid ka leida juba tarkvaraarenduse protsessi esimestes etappides selliseid vigu nagu ebatäpne dokumentatsioon, täitmata või osaliselt täidetud nõuded, disainivead, funktsionaalsuse vead (Kaljula, 2004, lk 6). Tarkvara testimisega tuleb tegeleda tarkvara arenduse protsessiga paralleelselt, sest nii saab avastada vigu kõigist tarkvara arenduse protsessi etappidest.

Süsteemi nõuded saab jagada laias laastus funktsionaalseteks ja mittefunktsionaalseteks. Funktsionaalsusnõuded saadakse otseselt süsteemile püstitatud eesmärkidest ja need on tihedalt seotud süsteemi toimimisvaldkonnaga ning neid suudavad kõige efektiivsemalt testida kasutajad, kes antud valdkonnas orienteeruvad. Mittefunktsionaalsed nõuded tagavad eesmärkide saavutamise ja siia kuuluvad nõuded sellistele aspektidele nagu jõudlus, turvalisus, paindlikkus, portatiivsus vms. Need on tehnilisemat laadi nõuded ning rohkem arendajate ja sõltumatute testijate pärusmaa. (Hülp, 2004, lk 127).

5.1 Kasutatavus

Käesolevas üliõpilasmoodulis rakendatakse kasutatavuse testi. Kasutatavuse testimine tunneb huvi selliste kriteeriumide vastu nagu kontseptuaalne selgus, õpitavus, kasutusmugavus. Testide tulemused mõjutavad sageli tarkvara disaini, muutes seeläbi programmi kasutamise lihtsamaks, efektiivsemaks ja meeldivamaks. Kasutatavuse all mõistetakse seda, et inimesed, kes toodet kasutavad, saavad seda teha kiiresti ja lihtsalt, et täita oma tööülesandeid.

Kuidas tagada kasutatavust:

- kasutajate kaasamine arendusse;
- rakenduse kasutuslikkuse testimine.

Kasutuslikkuse testimisel kogutakse:

- jõudluse mõõte – vaadeldud tegevuste arv;
- subjektiivseid mõõte – inimeste arvamused ja hinnangud.

Näiteid jõudluse mõõtudest (Roo, 2003):

- ülesande täitmisele kulunud aeg;
- kasutajajuhendi lugemisele kulunud aeg;
- valede menüüvalikute arv;
- valede klahvivajutuste arv;
- kasutajajuhendi poole pöördumiste arv;
- segadusse sattumiste arv;
- rahuldustunde väljendamiste arv.

Näiteid subjektiivsetest mõõtudest (*Ibid.*):

- hinnangud toote kasutamise õppimise lihtsusele;
- hinnangud toote kasutamise lihtsusele;
- hinnangud konkreetse ülesande täitmise lihtsusele;
- spontaansed märkused - “See oli lihtne!”, “Ma ei saa mitte midagi aru!”

Tähelepanu tuleb suunata sellele, et kasutajad ütleksid: “See on suurepärane toode! Seda on lihtne kasutada!” Eesmärgiks ei ole see, et toode läbiks testi edukalt, vaid toote muutmine selliseks, mida inimestel meeldib kasutada.

Kasutatavuse testimiseks koostatakse võimalikult täpsed küsimused, millele testimine peab vastuse andma. Nende põhjal täpsustatakse ülesanded, mida tuleks testimise käigus testitava tarkvara abil lahendada. Kasutatavuse testimisel rakendatakse ka jälgimist. Jälgitakse inimesi, kes kasutavad mingit toodet oma töökeskkonnas. Kasutatavuse testimine on kõige efektiivsem, kui üritatakse määrata, kui palju vajab kasutaja aega, et täita mingi ülesanne või ülesannete hulk, ja kui raske täitmine tundub. Tarkvaraarendajaid huvitavad testimise reaalsed tulemused, näiteks ülesande täitmise stopperiga mõõdetud ajad. Selle põhjal saab arvutada keskmise aja mingi ülesande täitmiseks. Samuti saab registreerida, kui palju kasutaja teeb vigu ja kui palju abisoove tal tekib. Arendajad saavad olulist teavet ka kasutaja vaatlemise kaudu. Vaatlejad talletavad täpselt, millisel kohal ülesande täitmise käigus kasutaja tegevus aeglustub, peatub või kaldub ülesandest kõrvale. Registreeritakse, millal kasutaja hakkab proovima midagi uut. Vaatlejad

julgustavad kasutajaid ülesande täitmisel valjusti rääkima. Kasutatavuse mistahes tüüpi testimisel tuleb silmas pida, et testitakse tarkvara, mitte kasutajat.

Käesoleva rakenduse testimises osalesid direktor, õppeosakonna juhataja, direktori asetäitja, juhiabi ja õppeosakonna sekretär. Testimisel võis kasutada kasutajajuhendit (esitatud lisas 7) ning test koosnes järgmistest protseduuridest:

A. Praktilised ülesanded

1. Täita üliõpilase isikukaart prooviandmetega.
2. Muuta sisestatud üliõpilase andmed:
 - õppevorm vastupidiseks (kui enne oli päevane, siis nüüd kaugõpe ja vastupidi);
 - sisestada (juurde) lisaandmeid.
3. Esitada järgmine protokoll:
 - õppeaine: matemaatika;
 - ainepunktide arv: 2;
 - teadmiste kontrolli liik: hindeline arvestus;
 - õppejõu eesnimi: Kati;
 - õppejõu perekonnanimi: Karu;
 - kursus: I äriinfotehnoloogia, päevane õppevorm.
4. Lisada hinded I kaugõppe raamatupidajatele:
 - õppejõu eesnimi: Mati;
 - õppejõu perekonnanimi: Karu;
 - semester: kevad 2005;
 - õppeaine: makroökonomika;
 - ainepunktide arv: 2;
 - lisada hinded ja vajadusel märkused.
5. Muuta eelnevalt I kaugõppe raamatupidajatele sisestatud hinded:
 - kahel vabalt valitud üliõpilasel muuta makroökonomika hinded.
6. Teostada järgmised päringud:
 - esitada üliõpilase Rein Rebane semestri sügis 2004 keskmine hinne (aritmeetiline keskmine);
 - koostada I päevase äriinfotehnoloogia kursuse kõikide hinnete aritmeetiline keskmine;

- esitada I päevase äriinfotehnoloogia kursuse õppekava täitmise protsent koos hinnete aritmeetilise keskmisega;
- esitada üliõpilase Siiri Susi õppekava täitmise protsent;
- esitada kaugõppe II sotsiaaltöö kursuse nimekiri, kes elavad väljaspool Vinni valda ja Vinni vallaga piirnevaid valdu;
- koostada raamatupidajate päevase õppevormi nimekiri, kes elavad väljaspool Vinni valda ja Vinni vallaga piirnevaid valdu;
- esitada üliõpilaste arvuga seotud päring.

B. Hinnangud

1. Hinda tarkvara kasutamise lihtsust 10 palli süsteemis, kus hinne 10 on väga lihtne ja hinne 1 äärmiselt keeruline.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

2. Hinda konkreetse ülesande täitmise lihtsust 10-palli süsteemis.

- Üliõpilase isikukaardi täitmine prooviandmetega.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

- Sisestatud üliõpilase andmete muutmine.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

- Protokolli koostamine.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

- Hinnete lisamine.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

- Hinnete muutmine.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

- Päringute teostamine.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

3. Hinda käesoleva rakenduse ülesehituse loogilisust 10-palli süsteemis.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

4. Hinda käesoleva rakenduse välist kujundust 10-palli süsteemis.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

5. Teie ettepanekud üliõpilasmooduli kasutajasõbralikumaks muutmise kohta.

6. Kas käesolevas rakenduses oli midagi mis häiris ja oleks vaja teisiti teha?

Kokkuvõtte kasutatavuse testimisest

Kasutajad rääkisid ülesannete täitmise juures valjusti kaasa, probleemi ilmnemisel küsisid ning otsisid ise vastused kasutajajuhendist. Käesoleva töö autor ei sekkunud testimise käigus kasutajate tegevusse, vaid osales kui välisvaatleja ning tegi märkmeid. Praktiliste ülesannete lahendamise käigus tekkis nõ oodatud probleem neil kasutajatel, kes nägid rakendust esmakordselt. Raskust valmistas PDF-failist tagasi pöördumine, kuid sealkohal saadi abi kasutajajuhendist ning intuitsioonist – vaja on vajutada rohelist *back*-nuppu. Rohkem probleeme ei tekkinud, kiideti kasutamise lihtsust, arusaadavust, loogilisust. Hinnangud 10-palli süsteemis olid valdavalt kümned, mõned üheksad.

Kasutatavuse subjektiivsete hinnangute näitajatena kõlasid väljendid: “Meie töö läheb nüüd väga lihtsaks!”, “Enam ei ole vaja teha sellisel hulgal käsitööd!”, “Missugune aja- ja närvikulu kokkuhoid!”, “Nii lihtne kasutada, teistkordsel kasutamisel oskan PDF-failist tagasi minna!”

Ettepanekutena soovitati lisada värve, et tekiks parem töötegemise tunne – kasutajaliideses on nüüd kasutatud kooli peahoone pilti, logo, halli ja sinist värvi. Lisaks rõhutati üliõpilasandmetega seotud päringute grupeerimist ning pealkirjastamist, et genereeritud tulemus oleks paremini jälgitav ning leitav. Päringu tulemused on nüüd grupeeritud ja pealkirjastatud.

Käesolev üliõpilasmoodul vajab andmete õigsuse testimist, et vältida olukorda, kus päringu tulemused on ebausaldusväärsed.

Kokkuvõte

Magistritöö eesmärk oli:

- anda ülevaade tarkvara arendusetappidest Lääne-Virumaa Kutsekõrgkooli näitel;
- rakendust loova uurimuse kaudu analüüsida olukorda enne ja pärast üliõpilasmooduli loomist/rakendamist;
- õppeinfosüsteemi üliõpilasmooduli loomise/rakendamise kaudu lihtsustada ja kiirendada õppeosakonna tööülesannete täitmist.

Tarkvara arendusetapid on: eeluuring, analüüs, tarkvara disain ja realisatsioon, testimine, juurutamine, hooldus.

Magistritöö esimene peatükk käsitles eeluuringu osa, toimus ideede ja informatsiooni otsimine, leiti vastused küsimustele:

Kuidas toimivad analoogsed süsteemid Võrumaa Kutsehariduskeskuses ja Tartu Kutsehariduskeskuses?

Millised õigusaktid reguleerivad üliõpilasmooduli loomisel tekkida võivaid probleeme?

Teine peatükk keskendus tarkvara arendusetapi teisele punktile – analüüsile. Käesoleva rakendust loova uurimuse kaudu analüüsiti teises peatükis hetkeolukorda, kirjeldati kuidas toimub infovahetus statistikaametiga, millised probleemid ilmnevad kui kogu hindeid ning üliõpilaste andmeid puudutav informatsioon on erinevates paberites ja arvutites, kes on kasutajad ning missugused on kasutajate vajadused ja nõuded. Näiteks: kui enne üliõpilasmooduli loomist pidid kursusejuhatajad esitama semestri lõpus õppeosakonda aruanded, mis kajastaksid üliõpilaste keskmist hinnet ning õppekava täitmise protsenti, siis käesoleva rakenduse abil on õppeosakonnal võimalik esitada kursuste aritmeetilise keskmise ning õppekava täitmise protsendi kohta päring ning printida või salvestada tulemus PDF-failina.

Tarkvaraarenduse teooria on kolmas peatükk. Olemasoleva üliõpilasmooduli tööprotsesside paremaks juhtimiseks sobis mitme-etapiline mudel, kuna rakenduse valmimine ja kasutuselevõtt toimuvad funktsioonide kaupa ning seega on riskid vähendatud, probleemid ilmnevad varakult,

suureneb planeerimise adekvaatsus, kasutaja tekkinud nõudmisi saab arvestada järgmise või uue funktsiooni loomisel.

Tarkvaraarenduse kolmandaks etapiks ning magistritöö neljandaks peatükiks on rakenduse realiseerimine. Neljandas peatükis käsitleti järgmiseid teemasid: klient-server rakenduse üldine tööpraktika, PHP ja MySQL, vormi kohustuslike väljade kontrollimine, andmebaasi ja päringute tegemine, päringud, kasutaja autentimine ning kokkuvõtte tehnilise teostuse kitsaskohtadest.

Viies peatükk keskendus üliõpilasmooduli kasutatavuse testimisele ning kasutajate hinnangutele, testimine on tarkvaraarenduse neljas etapp.

Üliõpilasmooduli juurutamine kui tarkvaraarenduse viies etapp algab õppeaastal 2005/2006 ning esialgseteks andmeteks on tulevaste esimeste kursuste üliõpilaste andmed. Üliõpilasmooduli kasutuselevõtt ning laiendus toimub järk-järgult, aastal 2006/2007 toimub uute üliõpilaste lisamine ning pidevalt uuendatakse rakendust vajaminevate funktsioonidega. Toimub rakenduse hooldusetapp – tarkvaraarenduse viimane ning produktiivne osa.

Kokkuvõtteks võib öelda, et õppeinfosüsteemi üliõpilasmoodul kergendab õppeosakonna tööd tähelepanuväärselt ning seega on magistritööl samuti praktiline väärtus, kuna aitab igapäevast tööd lihtsustada ja vajaminevaid tööülesandeid kiiremini ning tulemusrikkamalt täita. Valminud magistritöö aitas autoril paremini aru saada ning tundma õppida tarkvaraarendusteooriat ja andis ühtlasi väärtusliku praktilise kogemuse tulevikuks.

Üliõpilasmooduli edasiarenduseks on järgmised etapid:

- üliõpilaste ligipääsu tagamine isiklikele hinnetelehtedele, autentimine ning paroolivahetus, võimalusel sidumine IVA sisselogimisandmetega;
- aineõpetajate ning kursusejuhatajate lisamine andmete ning hinnete sisestajate rolli;
- lõpudokumentide printimine;
- üliõpilaste päringu genereerimine vastavalt õppekavale ning üliõpilaste sooritatud tulemustele.

Lääne-Virumaa Kutsekõrgkooli õppeinfosüsteemi üliõpilasmooduli rakendamise eesmärgi ning visiooni võtab töö autor kokku sõnadega: “Et jookseksid andmed, aga mitte inimesed.”

Käesolev magistritöö täitis püstitatud eesmärgi.

Kasutatud kirjandus

- Andmekogude seadus.** RTI 2004, 30, 204, 28.04.2004.
- Avaliku teabe seadus.** RTI, 30.11.2004, 81, 542, 10.12.2004.
- Briggs, J.** (1998); Sissejuhatus tarkvara arendusse. Nõuded ja spetsifikatsioonid.
[http://www.tech.port.ac.uk/staffweb/briggsj/old_units/SOENG/reqs.htm]. 15.04.2005.
- Bulger, B, Greenspan, J.** (2001); MySQL/PHP Database Applications. New York, 622 lk.
- Castagnetto, D, Rawat, H, jt.** (2001); Профессиональное PHP программирование. Санкт-Петербург, 910 lk.
- Eesti Kutseõppeasutuste Kvaliteediauhinna taotlusedokument.** (2004); Lääne-Virumaa Kutsekõrgkool, 59 lk.
- E-teatmik – inglise keelsete info- ja sidetehnoloogia terminite seletav sõnaraamat.**
[<http://www.vallaste.ee/index.htm?Type=UserId&otsing=2920>]. 08.04.2005.
- Haamer, I.** (2001); Ettevõtte infosüsteemid. Audentese Ülikool. [Loengukonspekt].
- Hanson, V, Buldas, A, jt.** (1998); Infosüsteemide turve II Turbetehnoloogia. Tallinn, 371 lk.
- Hanson, V, Tavast, A.** (2000); Arvutikasutaja sõnastik inglise-eesti. Tallinn: AS Kirjastus Ilo, 284 lk.
- Helinurme, A.** (2003); Veebipõhised õppematerjalid
[<http://helinurm.tpt.edu.ee/vabavara/pdf.htm>]. 06.03.2005.
- Hülp, E.** (2004); Infosüsteemide testimisest. A&A. 4/5. lk 123-131.
- Ilitson, T.** (2004); PHP aabits. [<http://bytez.net/php/aabits>]. 18.03.2005.
- Infosüsteemide kolmeastmelise etalonturbe süsteem ISKE Rakendamisjuhend.** (2003); Versioon 1.0.
- Infosüsteemide turvameetmete süsteemi kehtestamine.** RTI, 26.08.2004, 63, 443, 29.08.2004.
- Isikuandmete automatiseeritud töötlemisel isiku kaitse konventsiooni ratifitseerimise seadus.** RTII, 03.01.2001, 1, 3, 01.06.2002.
- Isikuandmete kaitse seadus.** RT 2003, 26, 158, 01.05.2004.
- ISKE Lisad.** (2003); Versioon 1.0.
- Infotehnoloogia. Infoturbe halduse suunised. Osa 1: Infoturbe mõisted ja mudelid.** (1999); EVS-ISO/IEC TR 13335-1. 35 lk.

Infotehnoloogia. Sõnastik Osa 17: Andmebaasid. (1998); EVS-ISO/IEC TR 2382-17. 43 lk.

IT juhtimise käsiraamat. (1999); 1. Infosüsteemi kasutamine. Äripäeva kirjastus.

Kaljula, K. (2004); Tarkvara testimine nõuete formuleerimise, analüüsi ja disaini etapis. Tartu Ülikool, Tarkvara süsteemide õppetool, 66 lk. [Magistritöö].

Kivisild, D. (2002); Andmebaasid. Audentese Ülikool. [Loengukonpekt].

Koteroov, D. (2001); Самоучитель PHP 4. Санкт-Петербург, 556 lk.

Lenstra, A, Wang, X, Weger, B. (2005); Artikkel md5() kollisioonist. [http://www.win.tue.nl/~bdeweger/CollidingCertificates/]. 18.04.2005.

Majak, J. Õppematerjalid – PHP. [http://math.ut.ee/~jmajak/]. 25. jaanuar 2005.

Mikli, T. (1998); Sissejuhatus infosüsteemidesse. Tallinn, 99 lk.

MySQL-manual. Veebipõhine käsiraamat.
[http://dev.mysql.com/doc/mysql/en/password-hashing.html]. 05.04.2005.

Newman, William M., Lamming, Michael G. (1995); Interactive System Design. Cambridge, 468 lk.

Normak, P. (2004); Projektijuhtimine, Tallinna Pedagoogikaülikool. [Loengukonspekt]. 134 lk.

PHP manual. Veebipõhine käsiraamat.
[http://www.php.net/manual/en]. 03. mai 2005.

Põldoja, H. (2003); Veebipõhise õpihaldussüsteemi kasutajaliidese disain. Tallinna Pedagoogikaülikool, Informaatika osakond, 64 lk. [Magistritöö].

Raamatukogusõnastik - Rahvusraamatukogu terminoloogiasektori andmebaas.
[http://oldwww.nlib.ee/termin/termin.php]. 19.04.2005.

Rebane, K. (2003); PHP-MySQL rakendus: kirjalike tööde andmebaas. Audentese Ülikool, Informaatika õppetool, 54 lk. [Bakalaureusetöö].

Roo, R. (2003); Kasutuslikkus ja selle testimine. Esitlus.

Seeba, A. (2001); Unifitseeritud tarkvaraarendamise protsess ja selle rakendamise juhtumianalüüs. Tartu Ülikool, Tarkvarasüsteemide õppetool, 82 lk. [Magistritöö].

Septer, A. (2001); Arvuti ettevõtte asjaajamises. Tallinn, 487 lk.

Sherafat, R. (2005); Tarkvara arenduse seminar. Nõuete dokumentatsioon.
[http://64.233.183.104/search?q=cache:HSMiUjAN6FcJ:www.cas.mcmaster.ca/~sherafr/Univ/Courses/703/files/Requirements%2520Documentation.ppt+IEEE/ANSI+830-1993&hl=et].
21.04.2005.

Sibola, A. (2000); Veebiinfosüsteemid. Tartu Ülikool, Tarkvarasüsteemide õppetool, [Magistritöö]. [<http://kodu.neti.ee/~aulis/Opingud/AulisMag/>]. 25.veebruar 2005

Sinivee, Veiko. ID kaardi kasutamine autentimiseks. [<http://www.id.ee/file.php?id=108>]. 18.04.2005.

Sommerville, I. (2001); Software Engineering. Harlow, England, 693 lk.

Tartu Kutseharisukeskuse koduleht. [<http://www.khk.tartu.ee>]. 19.03.2005.

Tartu Ülikooli õppeinfosüsteemi veebipõhine abiinfo. [http://www.is.ut.ee/ois_help/tudabi_uj.1.1.htm#2]. 08.04.2005.

Tikk, E. Avaliku teabe seaduse rakendamise probleeme. 28. jaanuar 2003. Arvutimaailm. lk 18-21.

Viigipuu, R. Postiajaloo ja filateelia teemaliste artikliviidete andmebaas. (2004); Tallinna Pedagoogikaülikool, Informaatika osakond. 41 lk. [Seminaritöö].

Volere nõuete spetsifikatsioon. [<http://www.systemsguild.com/GuildSite/Robs/Template.html#anchor328466>]. 19.04.2005.

Võrumaa Kutsehariduskeskuse koduleht. [<http://www.vkhk.ee>]. 19.03.2005.

Üliõpilaste kirjalike tööde koostamine ja vormistamine. (2001); Metoodiline juhend. Tartu, 32 lk.

Summary

The theme of the present master's thesis is Software Development Stages on the Basis of the Student's Module of the Study Information System in Lääne-Viru County Higher Vocational School.

The aim of the master's thesis is: to give an overview of software development stages on the basis of Lääne-Viru County Higher Vocational School; to analyse the situation before and after creating the student's module by means of applicable study; to facilitate and accelerate the work of the Study Department by means of creating the student's module of the study information system.

The thesis consists of five chapters and gives an overview of software development stages.

Pre-study – before applying it is necessary to get a clear overview of the problem and possible solutions, to get to know analogical applications, and to study legal acts in force and standards.

Analysis – the present situation, the analysis of the users' necessities and requirements, the structure of the student's module, the parameters of the study information system, and the essential characteristics of the user interface. To direct the work process of the existing student's module more efficiently, the incremental release model was appropriate, as completing and introducing of the application happen in functions and therefore the risks have been reduced, problems appear early, the adequacy of planning increases, the user's arisen requirements can be taken into consideration while creating the next or a new function. Design and realisation – the general principle of the client-server application, the advantages and disadvantages of PHP/MySQL, the form and checking the obligatory fields of the form, the database, the inquiries, the authentication of the user. Testing – testing of usability and evaluation of users. Introducing and maintenance – will start in the academic year of 2005/2006 as well as entering the data of first-year students. The introducing and extension of the student's module will happen gradually.

In conclusion, it can be said that the student's module of the study information system will facilitate the work of the Study Department remarkably, and therefore the present master's thesis also has practical value as it helps to ease daily work and to perform necessary work tasks faster and more efficiently.

The present master's thesis helped the author to understand better and to get to know software development theory, and at the same time gave valuable practical experience for the future.

Lisad

Lisa 1 Intervjuu küsimused

1. Millistest moodulitest koosneb õppeinfosüsteem?
2. Mis keeles on kirjutatud antud rakendus ja miks?
3. Mitu tabelit on andmebaasis ning kuidas on tabelid omavahel seotud?
4. Kuidas õpetajad sisse logivad (kasutajanimi, parool, ID-kaart)?
5. Kuidas parooli saavad, kas saavad parooli muuta?
6. Kuidas on õigused jagatud? Kasutajate grupid?
7. Kas kõigil õpetajatel on ühesugune ligipääs andmebaasile? (Jah, Ei)
8. Mis toimingud on õpetajatel lubatud?
9. Kas on olemas kasutajajuhendid? (Jah, Ei)
10. Missugune muu dokumentatsioon? Struktuurid? Seosed?
11. Kas ja millised probleeme on tekkinud?
12. Kas süsteem on lihtsalt uuendatav?
13. Kas antud süsteem on ühilduv mõne teise süsteemiga?
14. Kas antud süsteemist liiguvad andmed ka kuskile riiklikesse andmebaasidesse?
15. Kas antud süsteem võimaldab lõpudokumente printida? Kuidas toimub lõpudokumentide printimine?
16. Milliseid päringuid on võimalik teostada? Milliseid funktsioone peab süsteem täitma?
17. Kes neid päringu tulemusi vajab?
18. Kas iga kirje sisestamisega/muutmisega/kustutamisega kaasneb ajatempli lisamine?
19. Kuidas on lahendatud turvalisusega seotud probleemid?
20. Kas on oma töötaja, kes tegeleb ainult õppeinfosüsteemiga?
21. Milliseid tehnilised (riistvara- ja tarkvara) lahendused on vajalikud õppeinfosüsteemi tööshoidmiseks?
22. Kui palju kasutajaid on õppeinfosüsteemil?
23. Kas ja kuidas toimub õppeinfosüsteemi andmete varundamine?

Lisa 2



Lisa 3

VKHK õppetöö moodul

- Õppurite register
 - Andmete sisestamine
 - Andmete vaatamine ning muutmine
 - Failid õppuriteregistrisse
 - Tõend, haigekassa tõend
- Tunniplaaniid
 - Õppetöö graafik
 - Õpperühmade tunniplaani
 - Õpetajate tunniplaani
 - Ruumide tunniplaani
 - Tunniplaani tähised
- Õppekavad
 - Kooli õppekavad
 - Ainekaartide täitmine
 - Ainekaartide printimine
 - Lõppenud õppeained
- Õppematerjalid
 - Õppematerjalide lisamine
 - Õppematerjalide vaatamine/otsimine
- Tulemused
 - Võlglaled
 - Koondaruanne
 - Protokollide printimine
 - Tulemuste sisestamine
 - Hinnetelehed
 - Akadeemilised õiendid, Diploma suppliment
 - Lõputööd
 - Lõputunnistus/diplom

- Sessioonid
 - Planeerimine
 - Sessioonide plaan
 - Otsing
- Õppepraktika
 - Praktikakohad
 - Praktikaleping
 - Päeviku täitmine
 - Praktika päevik
- Sisseastumine
 - Andmete sisestamine
 - Andmete vaatamine/muutmine
 - Laekunud avaldused
 - Pingeread
 - Failid õppuriteregistrisse
- Täiendkoolitus
 - Õppekavad
 - Kursuste planeerimine
 - Kursustekalender (vaatamine ja parandamine)
 - Klientide registreerimine
 - Klientide andmed
 - Käskkirjad, tõendid
- Aruanded
 - Põhiõpe õpilaste arv
 - Õpperühmade ja õppijate arv kursuse ja õppekava järgi
 - Kutsealase eelkoolituse õpilaste arv
 - Õpilaste liikumine
 - Sisseastujate statistika
 - Õpetajate koormused
 - Ületundide arvestus

Lisa 4

Andmebaasile mõjuvad ohud (ISKE)

- Personali väljalangemine (haigus, õnnetus, surm, streik, lahkumine)
- Puuduvad, puudulikud või ühildamatud ressursid (GUI + nõrk arvuti, pistikud, liini üür)
- Revisjoniandmete analüüsi puudumine (regulaarsusel on peletav toime!)
- Tarkvara testimis- ja evitusprotseduuride puudumine või puudulikkus (lisaressursid, koolitus)
- Andmebaasi turvamehhanismide puudumine või puudulikkus (paroolid jms)
- Andmebaasihalduri keerukus (ebasobivus, väär installeerimine, halb baasi kontseptsioon)
- Andmebaasi kasutajate vahetumise halb korraldus (baasi "lahtijätmine")
- Andmekandjate puudulik talletus avarii puhuks (värskendused algul eraldi kandjale!)
- Koristajad jm väljastpoolt töötajad (ühenduste lahutamine, koristusvesi, äraviskamine, mängud)
- Väär pääsuõiguste haldus (võimalus kustutada logiandmeid, ebapiisav pääs oma töö tegemiseks)
- Andmebaasisüsteemi hooletu haldus (liigsed õigused, seire puudumine, harv varukopeerimine)
- Andmete kogemata manipuleerimine (oskamatus, teadmatus, liigsed õigused, hooletus)
- Andmebaasi väljalangemine (riistvara tõrke, ründe vm tõttu)
- Pääsu reguleerimisest möödumine ODBC kaudu (ODBC-draiverite väär installeerimine)
- Andmete kadu andmebaasis (kogemata, rünne, rike,...)
- Andmete kadu andmebaasis, salvestusruumi puudumise tõttu
- Andmebaasi tervikluse/konsistentsuse kadu (sünkr-viga; kogemata; rünne)
- IT-süsteemide volitamata kasutamine (sh paroolide muukimisega, nt Interneti uss)
- Kaughoolde portide väärkasutus (kräkkerite ründeobjekt)
- Süstemaatiline paroolide mõistatamine (D. Kleini uuring 1990, 15000 kontot: 24,2%)
- Andmete/tarkvara manipuleerimine andmebaasisüsteemis (pääsuõigustest möödumisel)
- Teenusetõkestus andmebaasisüsteemis (nt: suure päringumahuga, pääsu lukustusega)

Andmebaasile mõjuvaid ohtusid vähendavad turvameetmed (ISKE)

- Volitatud kasutajate ja õiguseprofiilide dokumenteerimine
- IT-süsteemi muutuste dokumenteerimine (uus operatsioonsüsteem, uued parameetrid)
- Juhendite hoidmine käepärast
- Sobiva andmebaasitarkvara valimine (võrrelda sisseehitatud turvameetmeid)
- Andmebaasi installeerimine ja konfigureerimine (SW installeerimine, baasi loomine, konfigureerimine, objektid)
- Andmebaasi turvakontseptsioon
- Andmebaasisüsteemi pääsu reguleerimine
- Andmebaasiinfo pääsu reguleerimine (ainult rakenduste kaudu; objektide kaitse,...)
- Andmebaasi tervikluse tagamine (pääsu reguleerimine, sisestuse kontroll)
- Andmebaasi kasutajate ja kasutajagruppide konfigureerimise reeglid
- Andmebaasisüsteemi logifailide kontroll
- Andmebaasipäringute suunised (maks. täpsus, väljad ilmutatult,...)
- Andmete turvaline teisaldus andmebaasi (eristada: algsed/vanad - regulaarsed)
- Väljaõpe enne programmi tegelikku kasutamist
- Usaldatav süsteemiülem ja ta asetäitja (valida hoolikalt, nõuda õiguste halduslikku kasutamist)

- PC ja serveri paroolkaitse (BIOS-parool; lisariistvara ja -tarkvara)
- Algparoolide muutmine
- Tarbetute andmebaasikontode lukustus ja kustutus
- Andmebaasi seire (regulaarne; killustus jms)
- Andmebaasilinkide kasutamise kitsendamine (lingib ainult baasiülem)
- Regulaarne andmevarundus
- Protseduurid andmebaasi tervikluse kao puhuks
- Andmevarundus andmebaasis (kogu baas harva võimalik, seetõttu poliitika järgi)
- Andmebaasi arhiveerimine (dokumenteerida: andmemudel, ajad, säilitusaeg)
- Andmebaasi taaste

Lisa 6

Õppeinfosüsteemi üliõpilasmooduli andmebaasi andmemudel

Tabel 1. Üliõpilase isikukaart

Välja nimi	Kirjeldus	Täitmis- kohustus	Tüüp	Võtmed
ID	Üliõpilase järjekorra nr	Jah	Auto num.	Võti
Õpinguraamatu number	Üliõpilase õpingurmt nr	Jah	Int	
Eesnimi	Üliõpilase eesnimi	Jah	Text	
Perenimi	Üliõpilase perenimi	Jah	Text	
Isikukood	Üliõpilase isikukood	Jah	Int	
Kodakondsus	Üliõpilase kodakondsus	Jah	Text	
Sugu	Üliõpilase sugu	Jah	Boolean	
Sünniaeg	Üliõpilase sünniaeg	Jah	Int	
Sünnikoht	Üliõpilase sünnikoht	Jah	Text	
Viimati lõpetatud kool	Kool, millele üliõpilane viimati lõpetas	Jah	Text	
Lauatelefon	Üliõpilase kodune telefon (lauatelefon)	Ei	Int	
Mobiil	Üliõpilase mobiil	Ei	Int	
Epost	Üliõpilase eposti aadress	Ei	Text	
Maakond	Maakond, kus üliõpilane on sisse kirjutatud	Jah	Text	
Linn/vald	Linn või vald, kuhu üliõpilane on sisse kirjutatud	Jah	Text	
Tänav/maja/korter/küla	Tänav ja maja/korter või küla, kuhu üliõpilane on sisse kirjutatud	Jah	Text	
Postiindeks	Postiindeks	Jah	Int	
Eriala	Eriala, kuhu üliõpilane soovib sisse astuda	Jah	Text	
Mitmes aasta	Mitmendale kursusele soovib üliõpilane sisse astuda	Jah	Int	
Riigieelarveline/ Riigieelarveväline	Kas soovib astuda riigieelarvelise või riigieelarvevälisele kohale	Jah	Boolean	
Päevane õppevorm/ Kaugõpe	Kas soovib õppida päevases õppevormis või kaugõppes	Jah	Boolean	
Täiskoormus/ Osakoormus	Kas üliõpilane soovib õppida täis- või osakoormusega õppes	Jah	Boolean	
Inglise keel/saksa keel	Kas põhivõõrkeel on olnud inglise keel või saksa keel	Jah	Boolean	

Lisa 6 järg

Välja nimi	Kirjeldus	Täitmis-kohustus	Tüüp	Võtmed
Vene keel	Kas üliõpilane on õppinud vene keelt – jah/ei	Jah	Boolean	
Ühiselamukoht	Kas üliõpilane vajab ühiselamukohta – jah/ei	Jah	Boolean	
Akadeemiline	Kas on akadeemilisel puhkusel – jah/ei	Ei	Boolean	
Lisaandmed	Lisaandmed üliõpilase koha	Ei	Text	

Tabel 2. Õppetulemused

Välja nimi	Kirjeldus	Täitmis-kohustus	Tüüp	Võtmed
ID	Hinde järjekorra number	Jah	Auto num.	Võti
isikukaartID	Isikukaardi järjekorra number	Jah	Int	Võõrvõti
Oppejoud_aineID	Õppejoud_aine tabeli järjekorranumber	Jah	Int	Võõrvõti
Hinne	Hinne	Jah	Int	
Märkus	Märkus	Ei	Text	

Tabel 3. Õppejoud_aine

Välja nimi	Kirjeldus	Täitmis-kohustus	Tüüp	Võtmed
ID	Kirje järjekorra number	Jah	Auto num.	Võti
Oppejou_eesnimi	Õppejõu eesnimi	Jah	Text	
oppejou_perenimi	Õppejõu perenimi	Jah	Text	
aine nimetus	Õppeaine nimetus	Jah	Text	
maht	Õppeaine maht ainepunktides	Jah	Int	
eriala	Eriala, mida üliõpilane õpib	Jah	Text	
mitmes_aasta	Mitmendal kursusel üliõpilane õpib	Jah	Int	
oppevorm	Kas on päevane õppevorm või kaugõpe	Jah	Boolean	
semester	Mitmes semester (aastaga). Nd: sygis2004	Jah	Text	

Sisse logimine

Ava veebisirvija (Internet Explorer, Mozilla vm) ning sisesta aadress:

<http://www.sfinks.pri.ee/mag/05avaleht.php>

Sisesta kasutajanimi: **kasutaja**

Sisesta parool: **salakala**

Üliõpilaste andmete sisestamine

→

Üliõpilaste andmete
sisestamiseks vali raadionupp:
Üliõpilaste andmete
sisestamine
ning kliki nupul Edasi

- ☒ Üliõpilaste andmete sisestamine
- ☐ Üliõpilaste andmete muutmine
- ☐ Protokollid
- ☐ Hinnete lisamine
- ☐ Hinnete muutmine
- ☐ Päringud

Edasi

Avaneb vorm – Üliõpilase isikukaart.

Täida vorm vajaminevate andmetega ning kliki nuppu **Saada**.

Üliõpilaste andmete muutmine

→

Üliõpilaste andmete
muutmiseks vali raadionupp:
Üliõpilaste andmete muutmine
ning kliki nupul Edasi

- ☐ Üliõpilaste andmete sisestamine
- ☒ Üliõpilaste andmete muutmine
- ☐ Protokollid
- ☐ Hinnete lisamine
- ☐ Hinnete muutmine
- ☐ Päringud

Edasi

Avaneb vorm, kuhu on vaja sisestada:

Üliõpilase eesnimi

→

Üliõpilase perekonnanimi

→

Kui on sisestatud üliõpilase,
kelle andmeid muuta tahate,
eesnimi ja perenimi, siis kliki
nupul **Edasi**

Sisestage üliõpilase eesnimi

Sisestage üliõpilase perekonnanimi

Avalehele

Edasi

Avaneb vorm, kus on üliõpilase andmed.

Tee vajaminevad parandused ning kliki nuppu **Muuda**

Hinnete protokollide loomiseks
vali raadionupp **Protokollid**
ja kliki nuppu **Edasi**

- ☐ Üliõpilaste andmete sisestamine
- ☐ Üliõpilaste andmete muutmine
- ☒ **Protokollid**
- ☐ Hinnete lisamine
- ☐ Hinnete muutmine
- ☐ Päringud

Edasi

Avaneb protokollide loomise vorm.

Täida vormi väljad vajaminevate andmetega ja kliki nupul **Edasi**

Avaneb PDF-failina protokoll.

Protokollide salvestamine.

Protokollide salvestamiseks vali: **File→Save Page As**

Otsi koht, kuhu soovid salvestada

Pane failile sobiv nimi

Vajuta Save nuppu



Tagasi Protokollide loomise vormile saab rohelise BACK nuppu abil.

Hinnete lisamine.

Hinnete lisamiseks vali raadionupp **Hinnete lisamine** ja kliki nuppu **Edasi**



- ☐ Üliõpilaste andmete sisestamine
- ☐ Üliõpilaste andmete muutmine
- ☐ Protokollid
- ☒ **Hinnete lisamine**
- ☐ Hinnete muutmine
- ☐ Päringud

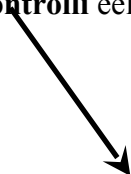
Edasi

Avaneb vorm **Õppetulemuste sisestamine**.

Täida vorm vajaminevate andmetega ning kliki nuppu **Edasi**.

Avaneb **Hinnete lisamise** vorm.

Kontrolli eelnevalt sisestatud andmete õigsust.



Õppejõu eesnimi: Kaarel
Õppejõu perenimi: Karu
Aine nimetus: Keemia
Aine maht: 2,5
Semester: kevad2005
Eriala: raamat
Mitmes aasta: 2,5
Õppevorm: kaugope

Täida vorm vajaminevate **hinnetega** ning lisa vajadusel **märkused**.

Kui andmed on lisatud, vajuta nuppu **Sisesta**.

Tagasi avalehele saab vajutades nuppu **Avalehele**.

Hinnete muutmine

Hinnete muutmiseks vali raadionupp **Hinnete muutmine** ja kliki nuppu **Edasi**

- ☐ Üliõpilaste andmete sisestamine
- ☐ Üliõpilaste andmete muutmine
- ☐ Protokollid
- ☐ Hinnete lisamine
- ☒ **Hinnete muutmine**
- ☐ Päringud

Avaneb vorm

Õppetulemuste muutmine.

Vali kas soovid muuta ühel üliõpilasel anded

või kogu kursusel korraga.

Ühe üliõpilase hinde muutmiseks vali järgnevad väljad

Üliõpilase eesnimi:

Üliõpilase perenimi:

Semester

Sügis 2004 ▾

Aine nimetus:

Kursuse hinnete muutmiseks vali järgnevad väljad

Valige eriala:

Raamatupidamine ▾

Mitmes aasta:

1 ▾

Päevane õppevorm:

☐

Kaugõpe:

☐

Semester

Sügis 2004 ▾

Aine nimetus:

Täida vorm vajaminevate andmetega ning kliki nuppu **Edasi**.

Avaneb **Hinnete muutmise** vorm.

Tee vajaminevad muutused hinnete ja märkuste osas ning vajuta nuppu **Sisesta**.

Tagasi avalehele saab vajutades nuppu Avalehele.

Päringud

Päringute loomiseks vali raadionupp **Päringud** ja kliki nuppu **Edasi**

- ☐ Üliõpilaste andmete sisestamine
- ☐ Üliõpilaste andmete muutmine
- ☐ Protokollid
- ☐ Hinnete lisamine
- ☐ Hinnete muutmine
- ☒ **Päringud**

Hinnete päring

Hinnete päringu tegemiseks vali raadionupp **Hinded (ariteemtiline keskmine)** ja kliki nuppu **Edasi**

- ☒ **Hinded (aritmeetiline keskmine)**
- ☐ Õppekava täitmise protsent
- ☐ Valdade päring (Täiendav toetus)
- ☐ Üliõpilaste arvuga seotud päring

- ☒ **Üliõpilase keskmine hinne**
- ☐ **Kursuse keskmine hinne**

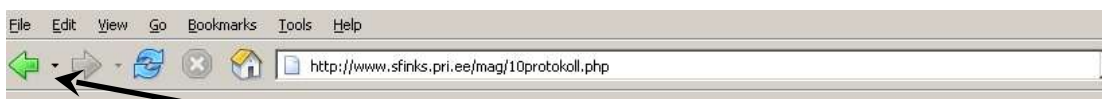
Vali kas soovid esitada päring **üliõpilase keskmise hinne** või **kursuse keskmise hinne** kohta.

Valides **üliõpilase keskmise hinne**, avaneb vorm, kuhu saab sisestada üliõpilase eesnime ja perenime ning on võimalik valida, kas soovite konkreetse semestri aritmeetilist keskmist või kõikide hinnete aritmeetilist keskmist.

Valides **kursuse keskmise hinne**, avaneb vorm, kuhu saab sisestada eriala, mitmenda aasta, õppevormi ning on võimalik valida, kas soovite konkreetse semestri aritmeetilist keskmist või kõikide hinnete aritmeetilist keskmist.

Klikkides nuppu **Edasi**, avaneb päringu tulemus PDF-failina. Päringu salvestamiseks vali:

File→Save Page As→ Otsi koht, kuhu soovid salvestada→ Pane failile sobiv nimi → Vajuta Save nuppu



Tagasi Protokollide loomise vormile saab rohelise BACK nuppu abil.

→
Õppekava täitmise protsendi päringu tegemiseks vali raadionupp **Õppekava täitmise protsent** ja klikki nuppu **Edasi**

- ☐ Hinded (aritmeetiline keskmine)
- ☒ **Õppekava täitmise protsent**
- ☐ Valdade päring (Täiendav toetus)
- ☐ Üliõpilaste arvuga seotud päring

- ☒ **Üliõpilase õppekava täitmise protsent**
- ☐ Kursuse õppekava täitmise protsent

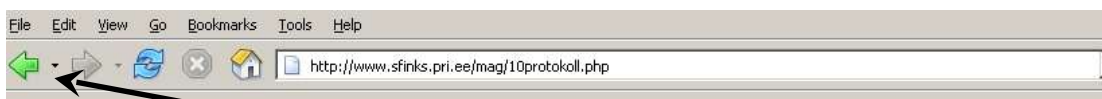
Vali kas soovid esitada päring **üliõpilase õppekava täitmise protsendi** või **kursuse õppekava täitmise** kohta.

Valides **üliõpilase õppekava täitmise protsendi**, avaneb vorm, kuhu saab sisestada üliõpilase eesnime ja perenime ning ainepunktide arvu, mis peab olema täidetud. Lisaks on võimalik valida, kas soovite koos aritmeetilise keskmisega või mitte.

Valides **kursuse õppekava täitmise protsendi**, avaneb vorm, kuhu saab sisestada eriala, mitmenda aasta, õppevormi ning ainepunktide arvu, mis peab olema täidetud. Lisaks on võimalik valida, kas soovite koos aritmeetilise keskmisega või mitte.

Klikkides nuppu **Edasi**, avaneb päringu tulemus PDF-failina. Päringu salvestamiseks vali:

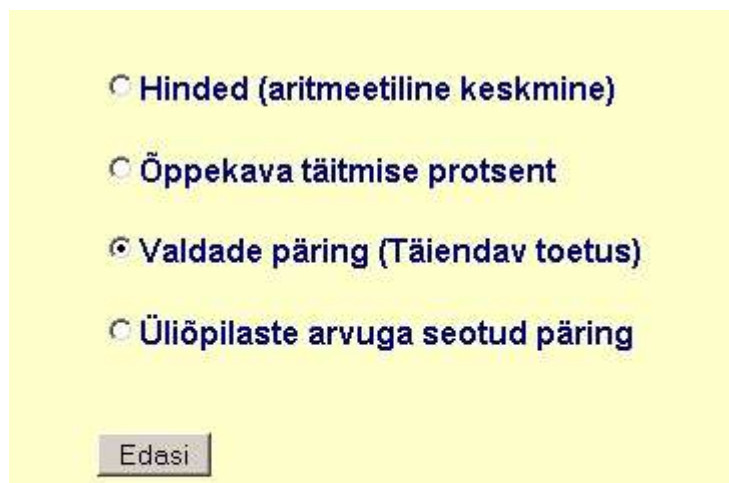
File→Save Page As→ Otsi koht, kuhu soovid salvestada→ Pane failile sobiv nimi → Vajuta Save nuppu



Tagasi Protokollide loomise vormile saab rohelise BACK nuppu abil.

Valdade päring

Selleks, et leida üliõpilaste nimekiri, kes elavad väljaspool Vinni valda ja Vinni vallaga piirnevaid valdu, vali päringute nimekirjast **Valdade päring (Täiendav toetus)** ja kliki nuppu **Edasi**



Järgnevalt vormilt vali **Kursuste kaup**, kui soovid näha konkreetse kursuse nimekirja, kes elavad

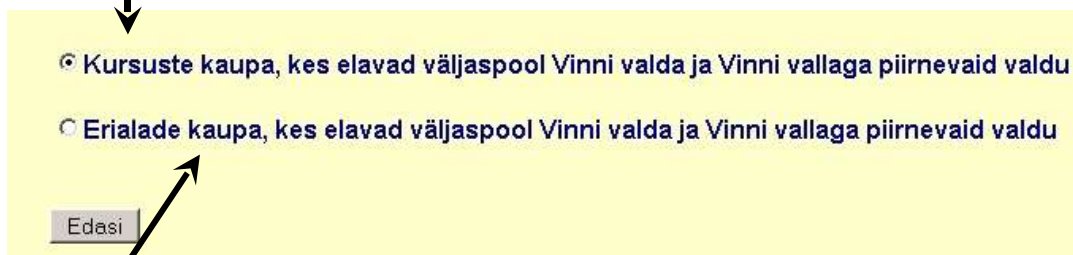
väljaspool

Vinni valda

ja Vinni

vallaga

piirnevaid valdu.



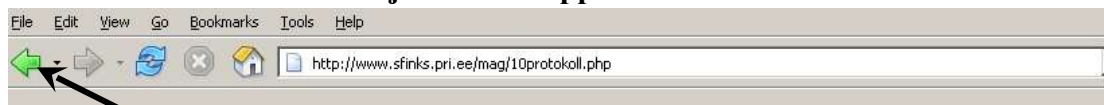
Vali **erialade kaup**, kui soovid näha konkreetse eriala nimekirja, kes elavad väljaspool Vinni valda ja Vinni vallaga piirnevaid valdu.

Klikkide nuppu **Edasi** jõuad vormile, kus on võimalik valida missuguse kursuse või eriala kohta päringute teha soovid.

Klikkides nuppu **Edasi**, avaneb päringu tulemus PDF-failina.

Päringu salvestamiseks vali: **File→Save Page As→ Otsi koht, kuhu soovid salvestada →**

Pane failile sobiv nimi→ Vajuta Save nuppu



Tagasi Protokollide loomise vormile saab rohelise BACK nuppu abil.

Üliõpilaste arvuga seotud päring

Selleks, et leida üliõpilaste arvuga seotud päringu tulemused, vali päringute nimekirjast **Üliõpilaste arvuga seotud päring** ja kliki nuppu **Edasi**



- ☐ Hinded (aritmeetiline keskmine)
- ☐ Õppekava täitmise protsent
- ☐ Valdade päring (Täiendav toetus)
- ☒ Üliõpilaste arvuga seotud päring

Edasi

Avalehele

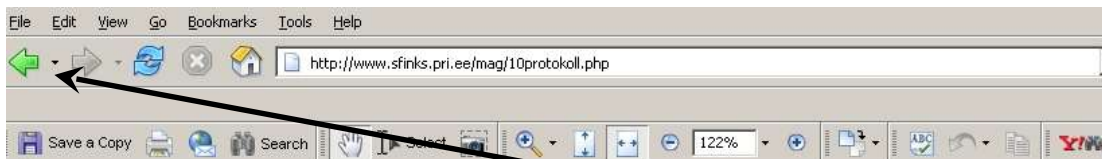
Klikkides nuppu **Edasi**, avaneb päringu tulemus PDF-failina.

Päringu salvestamiseks vali: **File→Save Page As**

Otsi koht, kuhu soovid salvestada

Pane failile sobiv nimi

Vajuta Save nuppu



Tagasi Protokollide loomise vormile saab rohelise BACK nuppu abil.

Lisa 8

Kasutajate hinnangud

Käesoleval üliõpilasmoodulil on suur praktiline tähtsus, mille vajalikkus tulenes igapäevasest õppetöö korraldamisest. Nimetatud rakendus kergendab tähelepanuväärselt õppeosakonna tööd, lihtsustab ja kiirendab tööülesannete täitmist ja aitab kaasa informatsiooni operatiivsemale liikumisele ja kättesaamisele.

Üliõpilasmooduli abil on võimalik leida üliõpilaste õppetulemuste keskmist hinnet ning õppekava täitmise protsenti, vajalikke andmeid õppetoetuse pingerea koostamiseks ja üliõpilasarvuga seotud tulemusi, mis eelkõige huvitavad statistikaametit. Üliõpilastega seotud andmete kiire kättesaadavus on efektiivse töö eelduseks.

Nimetatud rakendus täitis ootused kõrgelt, kuid edasiarendus oleks siiski vajalik.

Üliõpilasmooduli abil saab hästi pidada arvestust üliõpilaste akadeemilise edasijõudmise kohta, mida seni tehti suures osas paberkandjatel. Üliõpilased, kursusejuhendajad, õppeosakond jt saavad teostada vajalike päringuid üliõpilaste andmete kohta.

Üliõpilasmooduli edasiarendamisel peaks olema võimalik:

- lõpudokumentide printimine;
- üliõpilastel näha oma hinnetelehti;
- päring ainete keskmise hinde kohta kursuste lõikes;
- muuta õppekavas olevaid aineid, märkides ära muutmise aasta.

Helle Noorväli

Lääne-Virumaa Kutsekõrgkool

Direktor

Lisa 8 järg

Hinnang õppeinfosüsteemi üliõpilasmoodulile

Vastamaks Lääne-Virumaa Kutsekõrgkooli arengukava tänastele väljakutsetele ja pidevale vajadusele erinevaid üliõpilastega seotud andmeid kiiresti hankida, on õppeinfosüsteemi üliõpilasmooduli olemasolu ja selle rakendusvõimalused elu nõue.

Üliõpilasmooduli järele tekkis vajadus siis, kui Lääne-Virumaa Kutsekõrgkool ja Rakvere Pedagoogikakool ühinesid ning õpilaste arv peaaegu kolmekordistus. Üle tuhande üliõpilasega koolis on õppurite arvestuse käsitsi pidamine tööjõu ilmne raiskamine, aeganõudev ja koormav.

Eve Keerus-Jusupov on väga hästi aru saanud õppeosakonna probleemidest ja vajadustest ning rakendus on igati ootuspärane. Antud rakendus võimaldab lihtsustada mitmeid üliõpilaste arvestusega seotud valdkondi: statistiliste aruannete täitmine, üldine ülevaade üliõpilaskontingendist (arv), isikuandmed (vanus, elukoht, sugu, eelnevalt õpitud võõrkeeled, ühiselamu vajadus jne). Üliõpilasmoodulisse on võimalik sisestada nii andmeid kui päringuid iga üksiku üliõpilase õppekava täitmise kohta, võtta välja ka vastav protsent. Süsteem võimaldab opereerida ka muude andmetega. Rakendus võimaldab koostada üliõpilaste pingerida, mis on vajalik näiteks õppetootuste määramisel; võimaldab üliõpilase järgmisele kursusele üleviimise, täis- või osakoormusega õppele jäämise või hoopis eksmatrikuleerimise küsimuse otsustamisel säästa ressursi (aeg, energia).

Õppeinfosüsteemi üliõpilasmooduli rakendamine hoiab kokku ametnike ja pedagoogide tööaega. Kõnealust õppinfosüsteemi võiks täiustada selliselt, et see võimaldaks õppejõududelksamiprotokollide täitmise otse süsteemi.

Üliõpilasmooduli edasiarendus lihtsustab ka lõpudokumentide vormistust.

Reet Niilus

Lääne-Virumaa Kutsekõrgkool

õppeosakonna juhataja

Liivi Teppe

Lääne-Virumaa Kutsekõrgkool

õppeosakonna sekretär

Lisa 8 järg

Hinnang õppeinfosüsteemi üliõpilasmoodulile

Kindlasti on ÕIS vaja eelkõige õppetulemuste ülevaadete võimalikult kiireks ja lihtsamaks koostamiseks. Rakendus täitis ootused peaaegu 100%-lt. Rakendus annab võimaluse tööaega mõistlikult kasutada, sest jääb ära palju rumalat nõiakäsitööd kõikvõimalike andmete leidmiseks erinevate institutsioonide aruannetele. Puudulikult ei ole minu arvates midagi teostatud, kuid väliskujundust võiks täiendada (näiteks avalehte rõõmsamaks, tihedamaks). Isiklikult olen huvitatud sellest, et oleks võimalik rakenduse abil teha ka kooli tõendeid, mida õpilased vajavad pidevalt (kooliaasta algul massiliselt).

juhiabi Kersti Riivits
Lääne-Virumaa Kutsekõrgkool

Lisa 8 järg

Hinnang õppeinfosüsteemi üliõpilasmoodulile

Antud üliõpilasmoodulit vajas kool selleks, et:

1. Süstematiseerida informatsiooni.
2. Võimaldada üliõpilastele igal hetkel juurdepääs oma õpitulemustele.
3. Muuta informatsioon kättesaadavaks kõigile, kes seda vajavad.
4. Lihtsustada õppejõududel dokumentatsiooni täitmist.
5. Võimaldada andmete analüüsimist.
6. Lihtsustada statistiliste aruannete täitmist.
7. Lihtsustada lõpudokumentide vormistamist.

Rakendus täitis ootused. Tänu sellele on edaspidi võimalik kiiresti välja tuua vajalikud andmed, mida erinevad ametkonnad koolilt nõuavad. Lisaks sellele võimaldab süsteem üliõpilastel olla kursis iseenda hetke õpitulemustega, vähendades läbipaistmatust. Puudusi hetkel välja tuua ei oska. Edaspidi peaks rakendus võimaldama ka elektroonilisi päevkuid.

Eda Vahero

Lääne-Virumaa Kutsekõrgkool

Direktori asetäitja