

Tallinna Pedagoogikaülikool

Matemaatika- Loodusteaduskond

Informaatika osakond

Proseminaritöö

Martin Ott

## **Elektrooniline tagasisideküsitluste süsteem**

**Juhendajad:** Marek Kusmin, Jaagup Kippar, Katrin Niglas

Tallinn 2002

# Sisukord

<a href="#">Sisukord</a> .....	2
<a href="#">1. Sissejuhatus</a> .....	3
<a href="#">2. Kasutamishend</a> .....	4
<a href="#">2.1. Administreerimine</a> .....	4
<a href="#">2.1.1. Küsimused</a> .....	4
<a href="#">2.1.2. Õppejõud</a> .....	5
<a href="#">2.1.3. Ained</a> .....	5
<a href="#">2.1.4. Küsitlused</a> .....	5
<a href="#">2.2. Küsitlusele vastamine</a> .....	5
<a href="#">3. Programmilahendus</a> .....	6
<a href="#">3.1. Koodi struktuur</a> .....	7
<a href="#">3.1.1. Programmifailid</a> .....	7
<a href="#">3.1.2. Teegifailid</a> .....	7
<a href="#">3.1.3. Klassid</a> .....	8
<a href="#">3.2. Andmebaas</a> .....	8
<a href="#">3.2.1. Diagramm</a> .....	10
<a href="#">4. Kokkuvõte</a> .....	12

## **1. Sissejuhatus**

Õppetöö korraldajail on teha raske töö – koostada igale kursusele sobiv ainekava nii, et selle järgimine end ära tasuks ja samas ka meeldiv oleks. Ühelt poolt on neil tarvis teada hetkel kooli lõpetajalt nõutavaid oskusi, leida sobivad oskajad õppejõud nende alast teavet jagama. Teiselt poolt aga on järjest olulisemaks muutumas õppuri enda arvamus just temal kavas olnud ainete kohta.

Õppejõud, kes ise loengutes pidevalt üliõpilastega kokku puutuvad, kasutavad tihti võimalust aine lõppedes kuulajailt nende arvamust küsida. Selline küsimine toob kindlasti selgust antud aine seisukohast vaadates. Võb-olla, et leiaks õppejõud üht-teist kasulikku ka natuke üldisemast küsimisest, mis annaks paljunäinud õppurile võimaluse väikselt võrrelda iga olnud ainet.

Käesoleva töö eesmärgiks on panna alus elektroonilisele süsteemile, mis võimaldaks korraldada küsitlusi ainete kohta ja koguda nende alusel saadud tulemusi. Küsitlus tehakse grupile üliõpilastele, kes on koos läbinud hulga aineid, et siis neid päritavate küsimuste alusel hinnata.

## **2. Kasutamisjuhend**

Küsitlemisel on kaks poolt. Kõigepealt küsitluste loomise pool e administreerimine, mille käigus pannakse kokku erinevate gruppide poolt teatud aja jooksul kuulatud loengud ja nende kohta esitatavaid küsimused ning määratakse nendele ligi pääsemiseks koodsõna ja parool.

Seejärel küsitlusele vastamise pool, kus vastava grupi koodsõna ja parooli teadja vastab kavas olnud loengute kohta küsitavatele küsimustele.

### **2.1. Administreerimine**

Administreerima asumisel peab kasutaja nagu ikka end kõigepealt süsteemile tutvustama kasutajanime ja parooli abil. Süsteemis on kahte liiki ülesandeid, mis teevad kasutajast *admini* ja/või *asjaajaja*. Admin saab kasutajaid tekitada ja muuta, paroole muuta ning määrata kasutajate rolli süsteemis (neid *adminideks* ja *asjaajajaiks* määrates). Asjaajaja ülesandeks on aga kõik, mis küsitlustesse puutub. Küsitluse tegemiseks peavad süsteemis olema kirjeldatud ained, neid seletavad õppejõud ja kõige lõpuks veel ka küsimused, millele vastata.

#### **2.1.1. Küsimused**

Küsimuse sisestamisel on võimalik valida nelja tüübi vahel:

1. Tekst-vastusega küsimuse korral antakse vastajale võimalus sisestada vastuseks vabas vormis teksti.
2. Hinnangut väljendava suuruse valimiseks kuvatakse valik ühest viieni, millest siis saab valida sobiva.
3. Üksik-valik ja Mitmik-valik tüüpide korral moodustab vastusevariandi kõrvalolevas tekstiväljas leiduv tekstirida (kasutaja poolt reavahetus klahviga ise tekitatud reavahetus). Variantide arv ei ole piiratud. Mitmik-valiku korral võib nendest valikutest valida rohkem kui ühe variandi. Üksik-valiku puhul üle ühe variandi valida ei saa.

### **2.1.2. Õppejõud**

Õppejõu eraldajaks on tema eesnimi ja perekonnanimi. Uue õppejõu sisestamisel määratakse talle indeks, mida kasutajal enamasti tarvis ei lähe ja mis edaspidi mõnele teisele õppejõule viitama hakata ei tohiks. See tähendab ka, et õppejõu nime muutmine tähendab uut nime ka kõigis temaga seotud küsitlustes ja kunagi salvestatud vastustes. Õppejõuga saab siduda aineid, mida ta annab.

### **2.1.3. Ained**

Aine määrab üheselt ainekood, mis peaks üle kogu süsteemi olema unikaalne. Kord juba salvestatud ainel koodi muutmiseks admin süsteem edaspidi mugavat võimalust ei paku. Valesti sisestatud koodiga aine korral tuleks teha uus aine sama nime ja õige koodiga ja juhul kui tolle vale ainega on seotud ka küsitlusi või vastuseid, siis panna nad viitama uuele ainele. Aine juurde tuleks märkida ka õppejõud, kes teda loevad.

### **2.1.4. Küsitlused**

Küsitluse juurde käiva koodi ja parooli abil teatab vastaja süsteemile, et just tema kuulub vastavasse õpperühma, et hinnata tol semestril kavas olnud aineid. Küsitluse kood peaks üheselt määrama ühe küsitluse. Samas on ta vabalt muudetav. Küsitlus on vastamiseks avatud alguskuupäevast kuni lõppkuupäevani ja neid võib samuti alati muuta. Sarnaselt õppejõududele määratakse ka küsitlustele loomisel oma unikaalne tunnus, mida pärast enam ei muudeta ega näidata eriliselt ka kasutajale.

Küsitlusele loengute lisamine tähendab ainete ja nendega seotud õppejõudude seast vastava kursuse ja semestri jaoks valiku tegemist. Juhul kui näiteks loengut ja praktikumi andsid erinevad õppejõud tuleks sama aine lisada küsitlusele lisada teist korda teise õppejõu ja vormiga.

## **2.2. Küsitlusele vastamine**

Tagasiside jagamise sooviga koodsõna ja parooli teadvale lahkele üliõpilasele kuvatakse küllaltki lihtne vorm, kus iga aine kohta reas koht küsimusele vastamiseks. Selleks, et täidetud küsitlus ka vastu võetud saaks tuleb igast valitavast vastusest (olenevalt küsimuse) tüübist vähemalt üks variant valida. Vabas vormis vastatava küsimuse võib ka vastamata jätta. Tekstiväljas leiduva vastuse korral kuulub see aga ka salvestamisele.

### **3. Programmilahendus**

Käesoleva töö aluseks olev rakendus on kirjutatud õhukese kliendi vormis klient-server rakendusena. Kliendi kasutatavaks programmiks on lihtne veebilehitseja (nagu seda on Mozilla, Netscape või Internet Explorer). Serveri poole pealt on vastamas PHP toetusega veebi teenindav programm (Apache näiteks) ja andmebaas (MySQL). Veebilehitsejast kliendi tõttu võib antud lahendust ka veebirakenduseks nimetada.

Sellisel vormi kasuks sundisid otsustama üsnagi mitu head omadust:

- Ta ei nõua kliendilt suurt midagi peale veebisirvija ja töötava arvutivõrgu olemasolu.
- Kogu oluline funktsionaalsus lahendatakse pühas segamatuses serveril lehekülgede serveerimise vaheajal. Olenemata kliendiprogrammi seisundist või kasutaja poolsest üritustest teada saada, mis rakenduse sees toimub.
- Rakenduse käitumine toimub tervenisti serveris paiknevate programmilõikude juhendamisel olles seega täielikult juhitud ühest kohast. Puudub vajadus kliendiprogramme uuendada või installeerida
- Kõik muudatused jõuavad kohe värskena ka andmebaasi.

Samas nagu ikka leidub nende vooruste kõrval ka puudusi:

- Veebilehitsemisvõimelisi programme on palju. Hoolimata standarditest erinevad nad siiski üksteisest üsnagi erineval viisil.
- Puudub võimalus üheselt identifitseerida ühte klienti, ühendust. Veebilehitsejad on üksteisest eristatavad sisuliselt ainult aadressi järgi millelt päring tehti. Sessioonide meelde jätmisel saab ainult loota, et järgmise päringu sessioonivõtme sisestanu on ka tegelikult sama, kes kunagi selle sisse logimisel sai.
- Vorme annab täita mitmel moel – ka pahatahtlikul moel.
- Kogu veatöötlus tuleb realiseerida serveril. Ka valesti täidetud vorm peab läbima pika tee serverini ja sealt jälle tagasi kliendini.

### **3.1. Koodi struktuur**

Kirjutatud rakendus koosneb nagu dünaamilised veebidokumendid ikka enamuses PHP lisandeid sisaldavatest tavalistest HTML failidest. Sellest tulenevalt on nende laiendiks `.php`. Kirjutamisel on arvestatud võimalustega, mida pakub PHP alates versioonist numbriga 4.

Funktsionaalsuselt on kood jaotatud kahte kataloogi: `quiz` (küsitlustele vastamine) ja `quiz/admin` (administreerimis-keskkond) vahel. Failid on samuti kahte gruppi jaotatud – (näiteks) programmifailid ja teegifailid.

#### **3.1.1. Programmifailid**

Programmifailid kujutavad endast reaalselt lehekülge väljastavat ja vormide sisestamise tulemusena erinevaid funktsioone täitvaid programme. Programmifailid sisaldavad ka funktsioone mida antud fail üksi mitmes kohas tarvitab.

Omamoodi erandiks on `blank.php`, mis teeb sisselugemisel kasutatavaks kolm funktsiooni tüüpilise HTML dokumendi loomiseks.

Programmifailid on kirjutatud võimalikult vähe PHP koodi sisaldavana. Kogu koodi osa on suletud PHP serveripoolse skripti märkide vahele ja koodi seest välja trükkimist võimaldatavaid lauseid on võimaluse korral välditud. Sellisel viisil on võimalik neid faile kujundada ka mõnes kavalamas hüperteksti kujundamise programmis (näiteks Dreamweaver).

#### **3.1.2. Teegifailid**

Teegifaile kasutatakse konfiguratsiooniparameetrite seadmiseks ning laiemalt kasutatavate funktsioonide ja klasside defineerimiseks – toiminguteks, mida tingimata enne programmi täitmist on tarvis teha. Enamasti ei tehta neid ka üle ühe korra (sellest tingituna on nende lugemiseks ideaalne `require_once` nimeline käsk). Teegifailidest ei oleks ilus teksti väljastada – HTML päiste (ka nende abil saadetavate küpsiste saatmine) ja teksti väljastamine on käesolevas rakenduses jäetud programmifailide ülesandeks. Küll aga võivad nad kirjeldada funktsioone, mis vastavaid toiminguid teevad, et neid

programmifailides lihtne kasutada oleks. Teegifailide tunnuseks on `.inc` enne faililaiendit.

Teegifailide otstarve on samuti failinimest loetav: klassidefinitsioone sisaldava faili nimi algab prefiksiga `class_`. Näiteks kõikide klasside korraga sisse laadimiseks on fail `classes.inc.php`.

### **3.1.3. Klassid**

Klass kujutab endast reaalselt andmebaasi tabeli(te)s pesitsevat objekti. Klassi juurde käivad veel meetodid objektiga sooritavate ülesannete täitmiseks (näiteks tema baasi salvestamine või seda tüüpi objektide lugemine baasist mingi tunnuse järgi). Sellise ehituse korral on baasis olevate objektide kallal töötamiseks vajalikud SQL laused kõik ühes kohas ja nii toob muudatus andmebaasisüsteemis endaga natuke vähem peavalu kaasa.

Programmifaili enda ülesandeks jääb ainult erinevad objektid HTML dokumendiks kokku siduda. Selline jaotatud kood on ka jupi võrra väiksemalt ühes failis korraga ja selliselt ka kergem lugeda.

## **3.2. Andmebaas**

Andmebaasi on tarvis salvestada objekte ja seoseid. Objektidele tekitatakse loomisel primaarvõtmeks unikaalne tunnus, mis identifitseerib üheselt teatud tüüpi objekti (teatud tabelis). Seoste salvestamisel ei ole tarvis primaarvõtmele oma unikaalset komponenti juurde lisada. Seose primaarvõtme saab täielikult moodustada seoses olevate objektide võtmetest.

Primaarvõti võib näidata ka mitu ühele seost mõne objektiga. Sellisel juhul salvestatakse seotud objekti primaarvõti ühe osana seotava objekti primaarvõtmest. Kokku saadud saab üheselt määrada ühe seotava objekti.

Andmebaasi salvestatakse järgmised objektid:

- Küsimused – tabelis Question. Lisaks salvestatakse valikküsimuste puhul variandid tabelisse QstData.
- Ained – tabelis Subject

- Õppejõud – tabelis Don
- Küsitlused – tabelis Query
- Kasutajad – tabelis User

Salvestatavateks seosteks on:

- Ainete ja Õppejõudude vaheliste seoste tabel Lecture. Tegemist on mitu-mitmele seosega ja omaette tabel kus nii aine kui ka õppejõu tunnus primaarvõtmeks on selle salvestamiseks igati soovitatav.
- Küsitlusega seotud loengute tabel QryLecture – sarnaselt Lecture tabelile seotakse loengud küsitlustega. Lisaks õppejõu ja aine tunnusele on tabelis veel küsitluse tunnus ja seose liik (loeng, praktikum, seminar).
- Küsitlusega seotud küsimuste tarbeks on QryQst nimeline tabel. Küsimustel on veel ka järjekorra määramise võimaldamiseks lisatunnus.

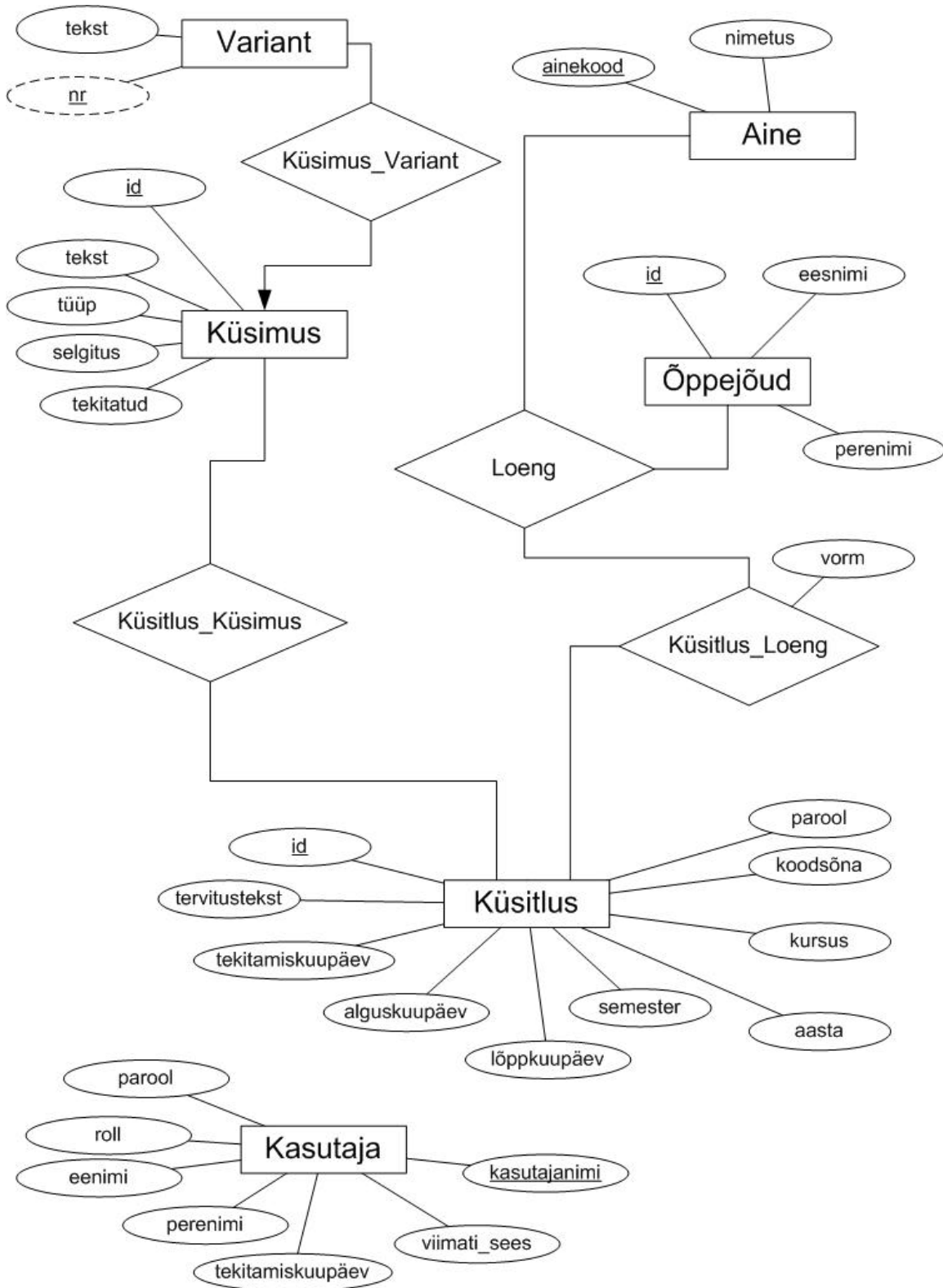
Vastused aga lähevad täitsa omaette tabelisse, kuhu lisaks vastusevariandile salvestatakse sessioonitunnus, küsitluse- , küsimuse- , aine- , õppejõu- tunnused. Tekstvastuse korral salvestatakse kirjutatud vastus omakorda Vastuse tekstide tabelisse, kus ta saab oma numbri, mis omakorda läheb vastusevariantina samasse Vastuste tabelisse. Teksti sisestamisel kui samasugune tekst juba ees on, siis Tekstide tabelisse uut rida uue tunnuse ja samasuguse vastusega tegema ei hakata.

Vastuste lugemisel tuleb vaadata vastusega seotud küsimuse tüüpi. Hinnangut tähendava küsimuse korral on tähendab vastusevarianti hinnet.

Valikküsimuse korral leidub variandi tekst leida küsimuse juurde käivate variantide tabelis.

Vabalt sisestatava tekstiga küsimuse vastuse koodi jaoks peaks vastava koodiga rida sisalduma ka Vastuse tekstide tabelis.

### 3.2.1. Diagramm



Joonis 1 Olem-seos diagramm

Joonisel 1 on toodud objektide ja seoste diagramm. Selguse huvides on vastused jooniselt välja jäetud. Kastid – olemid kujutavad endast andmegruppe, mis iseloomustavad ühte tabelisse salvestatavat objekti. Olemitel on oma primaarne tunnus, mis tabelis määrab üheselt ühe objekti.

Olemid võivad olla teistega seoses. Seoseid märgitakse rombidega, mis on objektidega ühenduses joonte abil. Seose iseloom võib olla üks-ühele, või mitu ühele. Noole kujuline joon märgib ühest seost

## **4. Kokkuvõte**

Käesoleva töö raames on alus pandud elektroonilisele süsteemile, mis võimaldab korraldada küsitlusi ainete kohta ja koguda ka nende alusel saadud tulemusi. Küsitlus on mõeldud grupile üliõpilastele, kes on koos läbinud teatud hulga aineid, et siis nende kohta küsitavatele küsimustele vastata.

Mõiste aluse panek tähendab siinkohal seda, et välja on töötatud andmebaas, mis võimaldaks kõike küsitluste koostamise ja vastamise poole pealt hoida ja sündinud on ka rakendus, mis seda baasi täidab. Tegemist on veel pigem prototüübiga – tulevase programmi funktsionaalse põhjaga, mida annab juba testida ja mis vahest ehk näeb ka natuke kena välja aga mis sellest hoolimata on veel alles toores.

Realisatsioon käesolevast tööst asub serveril, mis lubab endale ühendusi ainult TPÜ kohtvõrgust. Aadressiks on <http://greeny.cs.tpu.ee/~martino/php/quiz/admin/>