



Datavetenskap

Henrik Larsson och Mikael Björkegren

Webbaserat system för dugga

Examensarbete, C-nivå

2006:03

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Henrik Larsson

Mikael Björkegren

Godkänd, 2006-01-18

Handledare: Katarina Asplund

Examinator: Stefan Lindskog

Sammanfattning

Detta är ett 10 poängs examensarbete på C-nivå vid Karlstads Universitet. Målet med examensarbetet var att utveckla ett befintligt system för webbaserade kurstest så att det också kan användas för duggor. Säkerheten skulle förbättras så att studenter blir tvungna att logga in med användarnamn och lösenord när de ska skriva duggor. Resultaten från de studenter som skriver duggor skulle sparas i en databas. Vi hade också som mål att låta studenter testa systemet och att låta dem fylla i en enkät med frågor om vad de tycker om ett webbaserat system för duggor.

Det resultat vi har kommit fram till är ett system som både hanterar webbaserade kurstest och webbaserade duggor. Vi har gjort om systemet så att studenter som skriver en dugga måste fylla i användarnamn och lösenord. Vi lät dessutom studenterna på Datakommunikation 1 under höstterminen 2005 prova systemet. De genomförde en dugga med hjälp av systemet, och sedan lät vi dem fylla i en enkät där vi ställde frågor om vad de tycker om att göra webbaserade duggor. Efter att studenterna hade provat systemet och svarat på enkätfrågorna kom vi fram till att de flesta studenter tyckte att systemet fungerade väl och hade ett bra användargränssnitt. Systemet bör dock utvecklas ytterligare innan det tas i bruk av universitetet.

Online course testing system

Abstract

This is a bachelor's thesis at the University of Karlstad. The goal of this work was to further develop a system for online course testing so that it can be used for small exams. The security was going to be improved so that students have to log on with usernames and passwords when they are going to take exams. The results from the students who take exams were going to be stored in a database. Our goal was also to let students test our system, and to let them fill out a survey with questions about what they think of an online exam system.

The result is a system that manages both online course tests and small online exams. We have redeveloped the system so students who take an exam have to fill in username and password. We let the students on Datacommunications 1, fall 2005 test the system. They took a small exam in the new system, and after we let them fill out a survey in which we asked questions about what they think about writing exams online. After the students had tried the system and answered the survey-questions we concluded that most students thought the system was working well and that it had a good user interface. The system should be further developed though, before it can be used by the university.

Innehållsförteckning

1	Inledning	1
2	Utvecklingsverktyg	3
2.1	HTML	3
2.2	PHP och MySQL	4
2.2.1	PHP	4
2.2.2	MySQL	4
2.2.3	Exempel på en MySQL fråga med hjälp av PHP	5
2.3	JavaScript	6
2.4	Apache	6
3	Beskrivning av det gamla systemet	8
3.1	Systemdesign	9
3.2	Databasdesign	11
3.3	Problem	12
3.3.1	Ofullständigt program	12
3.3.2	Problem med webbläsare	12
3.3.3	Apache, MySql och PHP	13
3.3.4	Databasen	13
4	Resonemang kring det nya systemet	14
4.1	Krav	14
4.2	Fördelar och nackdelar	15
5	Beskrivning av det nya systemet	17
5.1	Exempel på förbättrad kod	17
5.2	Ändringar av gamla PHP-filer	18
5.3	Nya PHP-filer	24
6	Beskrivning av den nya databasen	27
6.1	Allmänt	27
	ER-diagram	28
6.2	Tabeller i databasen	29
6.2.1	Answer	29
6.2.2	Course	29
6.2.3	Question	30

6.2.4	Result.....	30
6.2.5	Student.....	31
6.2.6	Test.....	31
6.2.7	User.....	32
7	Enkätundersökning.....	33
7.1	Frågor.....	33
7.2	Resultat.....	34
7.2.1	Generella frågor om att skriva duggor på nätet.....	34
7.2.2	Frågor om det system du har provat.....	37
8	Diskussion.....	40
8.1	Resultat.....	40
8.2	Vidareutveckling.....	41
8.2.1	Säkerhet mot fusk.....	41
8.2.2	Design.....	41
8.2.3	Kodning.....	42
9	Summering.....	43
	Referenser.....	44
A	Manual.....	45
A.1	Kunskapstest.....	45
A.1.1	index.htm.....	45
A.1.2	student.php.....	46
A.1.3	student1.php.....	47
A.1.4	antal_fragor.php.....	48
A.1.5	test.php.....	49
A.1.6	ratta.php.....	50
A.2	Dugga.....	51
A.2.1	Index.htm.....	51
A.2.2	logginstudent.php.....	52
A.2.3	dugga.php.....	53
A.2.4	ratta.php.....	54
A.3	Administratörer/Lärare.....	55
A.3.1	index.htm.....	55
A.3.2	loggin.php.....	56
A.3.3	usermeny.php.....	57
A.3.4	rubrik.php.....	58
A.3.5	skapa.php.....	59
A.3.6	red_test.php.....	61
A.3.7	red_rubrik.php.....	62
A.3.8	redigera.php.....	63
A.3.9	registrera.php.....	65
A.3.10	listastud.php.....	66
A.3.11	show_result.php.....	67
B	Enkät.....	68

Figurförteckning

FIGUR 3.1-1 FILÖVERSIKT FÖR STUDENTER SOM GÖR KURSTEST [1].	9
FIGUR 3.1-2 FILÖVERSIKT FÖR KURSANSVARIGA [1].	10
FIGUR 3.2-1 ER-DIAGRAM ÖVER DEN GAMLA DATABASEN	11
FIGUR 5.1-1 GAMMAL KOD	17
FIGUR 5.1-2 NY KOD	18
FIGUR 5.2-1 DEN NYA FÖRSTASIDAN	19
FIGUR 5.2-2 NY SIDA FÖR KURSANSVARIGA.	20
FIGUR 5.2-3 NY SIDA FÖR ATT SKAPA TEST	21
FIGUR 5.2-4 NY SIDA FÖR ATT REDIGERA TEST	22
FIGUR 5.2-5 NY SIDA FÖR ATT TA BORT FRÅGOR FRÅN ETT TEST	23
FIGUR 5.3-1 INLOGGNINGSSIDA FÖR STUDENTER	24
FIGUR 5.3-2 EN DUGGA	25
FIGUR 5.3-3 RESULTATLISTA	26
FIGUR 6.1-1 ÖVERSIKT AV DATABASENS TABELLER	27
FIGUR 6.1-2 ENTITYRELATIONS-DIAGRAM ÖVER DATABASEN.	28
FIGUR 6.2-1 TABELLEN ANSWER	29
FIGUR 6.2-2 TABELLEN COURSE	30
FIGUR 6.2-3 TABELLEN QUESTION	30
FIGUR 6.2-4 TABELLEN RESULT	31
FIGUR 6.2-5 TABELLEN STUDENT	31
FIGUR 6.2-6 TABELLEN TEST	32
FIGUR 6.2-7 TABELLEN USER.	32

1 Inledning

Vårt examensarbete går ut på att förbättra säkerheten för ett webbaserat kunskapstest så att man kan använda det till duggor. Det gamla systemet utvecklades av Staffan Nilsson och Mattias Pehrsson [1] för den datavetenskapliga institutionen vid Karlstads universitet. Detta system utvecklades som ett system där man kan testa sina egna kunskaper online genom att man svarar på frågor interaktivt och sedan får de rätta svaren. Det innehåller dock ingen säkerhet och det är denna aspekt som vi har intresserat oss för med denna uppgift. Att kunna göra duggor online tycker vi skulle vara både smidigt och tidsbesparande. Huvudmålet med vårt arbete går därför ut på att skapa ett system som är så pass säkert att en lärare på en kurs kan lita på att duggan verkligen utförs av rätt student.

Det gamla systemet innehåller ingen som helst säkerhet för åtkomst av kurstesten, utan används endast för självinläring. Man kan välja kurs, kunskapstest och sedan göra testet om och om igen utan att systemet varken kontrollerar vem det är som använder systemet eller sparar några resultat. Utan att ta bort det gamla kunskapstestet ska vi utveckla systemet så att lärare också kan skapa duggor för deras studenter. Detta kräver naturligtvis någon form av inloggning av de studenter som ska skriva duggan, så att systemet kan spara deras resultat efter att svaren har fyllts i. Därefter ska systemet enkelt kunna lista tabeller över de studenter som har skrivit duggan och deras resultat. Meningen med uppgiften är att skapa ett alternativ till vanliga duggor i sal, som både är snabbare och enklare.

Förutom utvecklingen av det gamla systemet har vi också som mål att genomföra en utvärdering av det nya systemet. Detta planerar vi att genomföra genom att låta ett antal studenter prova systemet och sedan fylla i en enkät som innehåller de frågor vi vill ha svar på. För vår egen del tycker vi att ett onlinesystem för duggor vore ett bra alternativ, men det är intressant att få veta vad andra tycker också. Vi planerar att ställa frågor både om att skriva duggor online generellt och om vad studenterna tycker om det system vi har utvecklat, vad vi gjorde bra och vad som kunde ha gjort bättre. Undersökningen ska utföras med studenter som läser kursen Datakommunikation 1 under höstterminen 2005. Efter utvärderingen ska vi göra en sammanställning av studenternas åsikter och redovisa detta. På detta sätt kan vi få reda på systemet kan ytterligare utvecklas efter vårt arbete.

Resultatet av vårt arbete är ett vidareutvecklat system som nu kan hantera både duggor och kunskapstest. Systemet för att göra en dugga fungerar så att studenter får hämta ett användarnamn och lösenord hos kurssekreteraren som ger dem tillgång till en dugga. När studenten sedan loggar in med dessa användaruppgifter kommer han eller hon direkt till den rätta duggan som skrivs och sedan sparas i en databas. Kursansvariga och sekreterare kan sedan se efter i databasen hur det gick för studenterna som skrev duggan.

När vi hade vidareutvecklat systemet lät vi studenterna som läste Datakommunikation 1 under höstterminen 2005 prova systemet. De genomförde en riktigt dugga och alla resultat sparades i databasen. Efter duggan delade vi ut en enkät som innehöll frågor angående vårt system och om att göra duggor på nätet i allmänhet. De flesta studenterna var nöjda med systemets användargränssnitt, men det var för oss förvånansvärt många som inte såg något behov av ett webbaserat system för duggor. Genomförandet av duggan med studenterna som läste Datakommunikation 1 och den efterföljande enkäten gjorde det också klart att systemet bör vidareutvecklas ytterligare innan det tas i bruk på universitetet.

Dispositionen av uppsatsen ser ut enligt följande. I kapitel 2 förklarar vi kort om de olika verktyg vi har använt oss av för att utveckla systemet. Kapitel 3 är en beskrivning av hur systemet såg ut innan vi började arbeta med det. Systemdesignen och databasdesignen förklaras. Kapitel 3 innehåller också en del där vi förklarar de problem som vi stötte på när vi skulle börja arbeta med systemet. I kapitel 4 diskuterar vi vilka krav ett system som hanterar duggor bör ha. Vi tar upp olika fördelar och nackdelar med ett sådant system. Kapitel 5 är en beskrivning av det nya systemet vi har utvecklat. Här finns beskrivning på de filer vi har ändrat och de filer vi har skapat. Kapitel 6 är en ingående beskrivning av databasen, med ER-diagram och tabeller. I kapitel 7 går vi igenom den enkät vi delade ut till studenterna på Datakommunikation 1. Svaren redovisas antingen med text eller ett diagram. Kapitel 8 är ett diskussionskapitel. Vi diskuterar angående vad som är bra med systemet och vad som är dåligt. Vi beskriver också idéer till hur systemet kan förbättras, både sådana som vi har tänkt på själva och som vi har fått från handledare och de studenter som testade systemet. I kapitel 9 summerar vi kort de resultat vi har kommit fram till.

2 Utvecklingsverktyg

Under utvecklingen av systemet har vi använt ett antal olika programspråk vilka beskrivs i detta kapitel. Eftersom programmet är onlinebaserat har vi använt HTML-kod för att designa sidorna. HTML förklaras ytterligare i avsnitt 2.1. Information om olika test, kursanvariga, studenter och deras resultat ligger lagrade i databasen `enterprise.cse.kau.se`, vilket har krävt ett skriptspråk och ett språk för att hantera databaser. Till detta ändamål har vi valt PHP-skript och MySQL. PHP-skript och MySQL förklaras i avsnitt 2.2. Apache server beskrivs i avsnitt 2.4. Vi har även använt JavaScript för att hantera vissa pop-fönster och funktioner som kontrollerar vilken information användare skriver i de formulär som visas på skärmen när man kör systemet. JavaScript förklaras i avsnitt 2.3. För att få idéer till vissa kodstycken i HTML har vi för det mesta vänt oss mot "the World Wide Consortium" (W3C) [3] samt den närliggande sidan "W3 Schools" [4]. Denna användbara sida fann vi då vi läste om dynamisk HTML programmering [5].

2.1 HTML

HTML är ett känt sidbeskrivningsspråk för att skriva hypertext på the World Wide Web. HTML står för HyperText Markup Language. Hypertext innehåller länkar så att man kan länka ihop de sidor man skriver. På detta sätt kan sedan besökaren surfa mellan sidorna. HTML är ett universellt språk som kan läsas av alla olika webbläsare. HTML-sidor är textfiler med ändelsen `.html` eller `.htm`.

Syntaxen för HTML är enkel och vare sig man är van vid programmering eller nybörjare kan man snabbt lära sig att skapa enkla hemsidor. För att skapa HTML-text börjar man med att skriva en tag som ser ut på följande sätt: `<HTML>`. Denna tag indikerar att all text man kommer att skriva efteråt ska tolkas som HTML-kod. För att avsluta skriver man `</HTML>`. Sidan behöver också en *body*, d.v.s en kropp där man skriver koden som sedan ska visas på skärmen. För att markera början på en body skriver man taggen `<BODY>`. Också denna tag avslutas med en tag med samma syntax förutom att man lägger till en frontslash - `</BODY>`. På detta sätt skriver man all HTML-syntax, vare sig man skapar ett sidhuvud, en rad eller en kolumn på sidan. Börja med `<det du vill skapa>` och avsluta med `</det du vill avsluta>`.

2.2 PHP och MySQL

2.2.1 PHP

PHP är ett skript-språk som används tillsammans med HTML. Från början betydde PHP ”Personal Home Page Tools”, men när användandet ökade ändrade man namnet och idag betyder PHP ”Hypertext PreProcessor”[6]. PHP är ett språk som används på serversidan och är inbäddat i HTML. Detta innebär att servern på vilken koden körs måste ha stöd för PHP. Exempel på en sådan webbserver är Apache webserver, se avsnitt 2.4. Man kan inte skapa en webbsida enbart med hjälp av PHP, språket är ett hjälpmedel för att utöka HTML.

Med hjälp av PHP kan man skapa dynamiska webbsidor. Detta innebär att användare kan förändra innehållet i den databas som är kopplad till systemet interaktivt. Språket har stöd för flera olika typer av databaser och däribland MySQL, vilket är det databassystem vi har använt för denna uppgift.

Syntaxen i PHP liknar den i C och C++. För att bädda in PHP-kod i HTML skriver man taggen `<?PHP` och för att avsluta skriver man `?>`. Med dessa taggar kan man enkelt växla mellan vanlig HTML-kod och PHP. I PHP finns också inbyggda funktioner för databasanrop så att man ställa frågor till en databas och få ut svar.

2.2.2 MySQL

MySQL är utvecklat av MySQL AB[2]. Det är ett databashanteringssystem som används till att hantera och ställa frågor till databaser. För att använda MySQL krävs en webbserver som har stöd för MySQL[7]. En databas är en ordnad samling data som kräver en databashanterare såsom MySQL för att komma åt, hantera och manipulera data. Detta görs genom att man ställer SQL-frågor till databasen för att komma åt de tabeller, rader och kolumner som finns i databasen.

2.2.3 Exempel på en MySQL fråga med hjälp av PHP

Följande är ett exempel på en SQL-fråga i vårt system. Det använder PHP-kod för att koppla upp sig mot databasen och sedan ställa frågan.

Till att börja med måste man koppla upp sig mot databasen.

```
//Databasens url adress
$host = "enterprise.cse.kau.se";

//Användarnamnet och lösenordet till databasen
$user = "XXXX";
$password = "XXXXXXXXXX";

//Databasens namn
$dbname = "olts";

//Kopplar upp sig mot servern eller avslutar
$link = @mysql_connect($host, $user, $password)
        or die("Error connecting to enterprise: " . mysql_error());

//Väljer rätt databas på servern eller avslutar
$conn = @mysql_select_db($dbname, $link)
        or die("Error connecting to db" . mysql_error());
```

Nu är vi uppkopplade mot databasen (såvida inget fel har inträffat), och därmed är vi redo att ställa frågor till databasen. I detta fall vill vi ha reda på de resultat studenterna har skrivit på en dugga och skriva ut dem till skärmen.

```
/* Hämtar rubriken på testet från föregående sida och lagrar den i en lokal variabel. Rubriken är vald genom att den kursansvarige som använder systemet har valt vilken dugga han eller hon vill se resultaten för. */
```

```
$rubrik = $_POST['VISARUBRIK'];
```

```
/*Hämtar alla rader från tabellen Result i databasen där kolumnen Test överensstämmer med den rubrik man har tagit emot och lagrat i den lokala variabeln $rubrik.*/
```

```

$sql = "SELECT * FROM Result WHERE Result.Test = '$rubrik'";

//Lagrar resultat i en ny variabel $result
$result = mysql_query($sql) or die ("Error performing get results:" . mysql_error());

/*Denna while-loop skriver ut studenternas personnummer och resultat rad för rad med hjälp
av en array i vilken resultaten lagras.*/
while($row = mysql_fetch_array($result))
{
    echo'<tr><td>';
        print( $row["Personnr"] );
    echo'</td><td>';
        print( $row["Result"] );
    echo'</td></tr>';
}

```

2.3 JavaScript

JavaScript är ett skriptspråk som körs på klientsidan istället för på serversidan såsom PHP. Detta medför att användarens webbläsare kan tolka JavaScript. När användaren går in på en hemsida som har JavaScript laddas koden först ner och körs sedan på klientsidan. För att bädda in JavaScript i HTML skriver man taggen `<SCRIPT LANGUAGE = "JavaScript">` och för att avsluta skriver man `</SCRIPT>`.

Liksom i det gamla systemet har vi använt JavaScript till att hantera pop-up rutor och till funktioner som kontrollerar att användaren fyllt i de formulär som visas på skärmen korrekt. Syntaxen är densamma som i Java.

2.4 Apache

Apache är en webbserver som stöder PHP och MySQL. Med en server menas en plats där man kan spara datafiler som också kan nås från andra datorer (klienter) världen över. Eftersom Apacheservrar stöder PHP kan man skapa sidor med ändelsen `.php` på dem. När en klient går in på en sådan sida så kör Apacheservern PHP-koden och omvandlar den till ren HTML kod som klienten sedan kan läsa.

Enterprise på Karlstads universitet är en Apacheserver och det är här vi har lagt upp filerna till vårt system.

3 Beskrivning av det gamla systemet

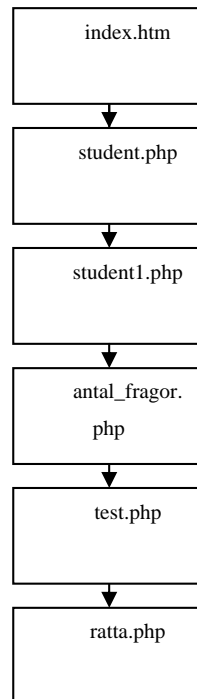
Systemet som utvecklades av Staffan Nilsson och Mattias Pehrsson [1] är ett självkunskapstest som utförs online. En student som använder systemet får välja kurs, kunskapstest och hur många frågor han eller hon vill göra. Testet slumpas fram med frågor och svarsalternativ som hämtas från en databas. Databasen innehåller frågor och svar till kurser samt användarnamn och tillhörande lösenord för kursansvariga. En student som gör testet tilldelas det antal frågor som valts tillsammans med fyra svarsalternativ, varav ett är korrekt. Efter att alla frågor har besvarats och man väljer att rätta får man se vilka frågor man svarat rätt på samt det totala resultatet. Man får dessutom en kommentar till de frågor man svarat fel på ifall den lärare som skapat testet har fyllt i en sådan. Efter att kunskapstestet är gjort kan man välja att gå tillbaka och slumpa fram ett nytt test eller att gå tillbaka till huvudmenyn.

En lärare för en kurs (kursansvarig) eller en administratör kan med hjälp av användarnamn och lösenord få tillgång till fler alternativ än vad en student får. En kursansvarig kan lägga till och ta bort frågor på en kurs, och kan dessutom redigera befintliga frågor. Att ta bort frågor är dock inte implementerat. Administratören har möjligheten att lägga till och redigera kursansvariga. Det är administratören som lägger till lärare för olika kurser och tilldelar användarnamn och lösenord. Administratören kan också lägga till och ta bort kurser. En lärare kan logga in och skapa test med tillhörande frågor för en specifik kurs.

Kurser som läggs till av administratörer och test som skapas av kursansvariga sparas i en databas som ligger på Karlstads Universitet. Servern är `enterprise.cse.kau.se`.

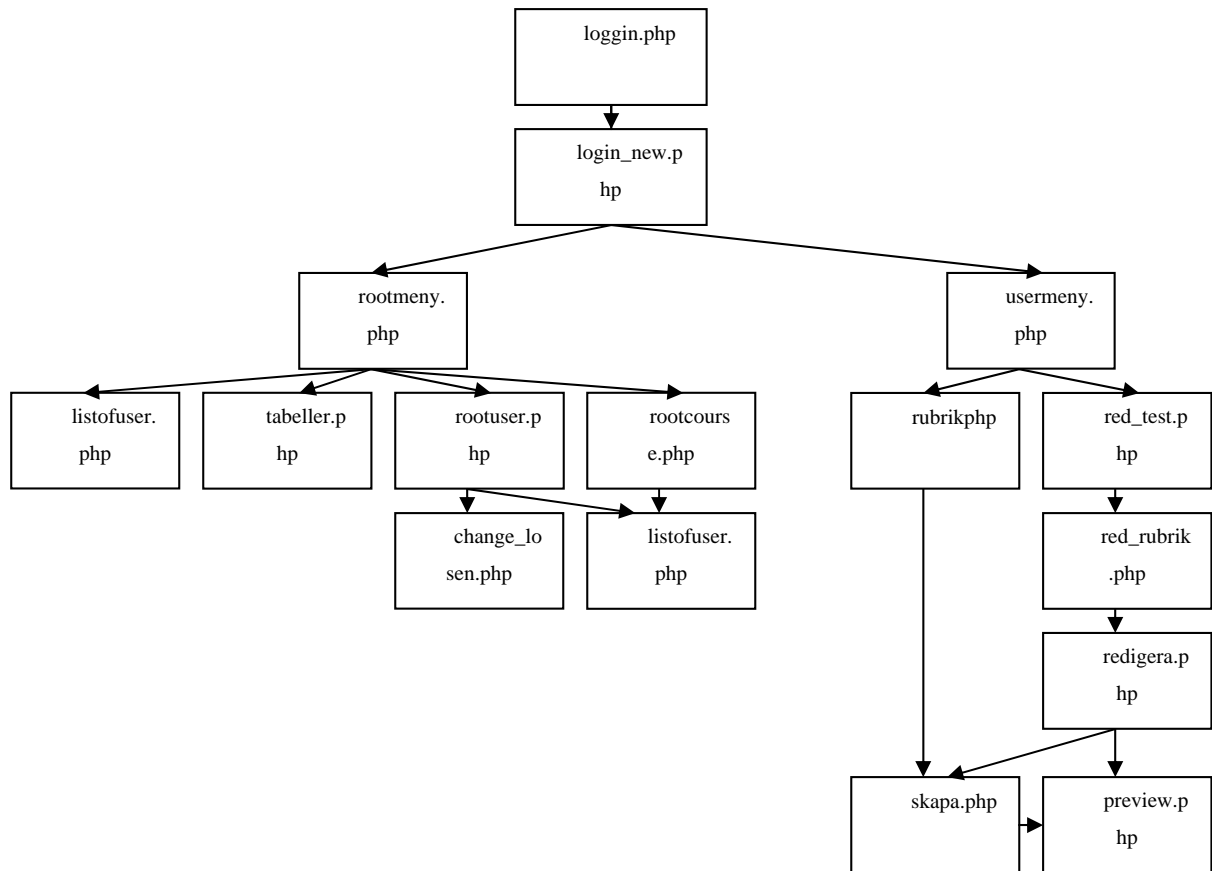
I avsnitt 3.1 beskrivs den gamla systemdesignen. Avsnitt 3.2 beskriver den gamla databasdesignen. I avsnitt 3.3 går vi igenom de problem vi stötte på när vi skulle vidareutveckla systemet.

3.1 Systemdesign



Figur 3.1-1 Filöversikt för studenter som gör kurstest [1].

Ovan visas en bild av de filer som används då en student använder systemet för att göra ett kunskapstest. *index.htm* är välkomstsidan där studenterna går vidare till nästa sida, *student.php*. Här väljs vilken kurs studenten vill göra ett test på, och på nästa sida *student1.php* väljer studenten ett kunskapstest som finns på kursen. Därefter väljs hur många frågor kunskapstestet ska innehålla i *antal_fragor.php* och slutligen får man fram kunskapstestet i *test.php*. Det sista en student gör är att rätta testet med hjälp av *ratta.php*. Efter att ett kunskapstest har utförts kan studenten gå tillbaka och börja om på nytt.

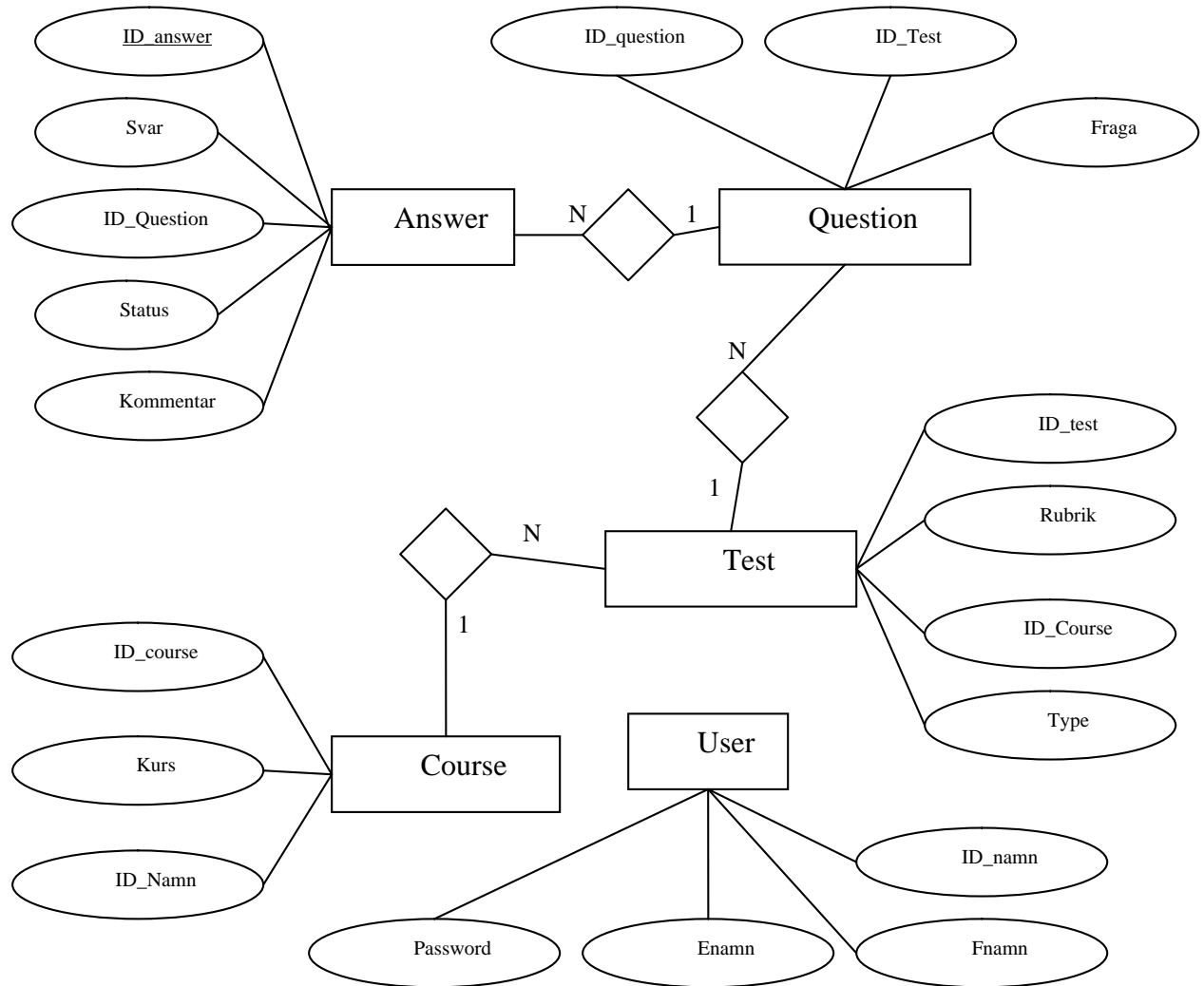


Figur 3.1-2 Filöversikt för kursansvariga [1].

Ovan visas en översikt för kursansvariga och systemansvariga. Från välkomstsidan väljer de att logga in med *loggin.php*. Beroende på vilken tillhörighet man har när man loggar in kommer man att länkas vidare till en sida för systemansvariga (*rootmeny.php*) eller en sida för kursansvariga (*usermeny.php*). Då vi inte har arbetat med att förbättra systemet för de systemansvariga går vi inte närmare in på dessa sidor. Den intresserade läsaren kan finna mer information om dessa filer i den föregående uppsatsen [1]. Som kursansvarig kommer man till en sida där man får valet att skapa ett nytt test eller redigera ett befintligt test. Väljer man att skapa ett nytt test kommer man till sidan *rubrik.php* där man får välja till vilken kurs man vill skapa ett kunskapstest och vilken rubrik testet ska ha. På denna sida kan man också ta bort test. När man har valt rubrik och går vidare kommer man till sidan *skapa.php* där man kan lägga in frågor i kunskapstestet. Väljer man istället att redigera ett test från förstasidan kommer man till sidan *red_test.php* där man får välja kurs, och sedan kommer man till sidan *red_rubrik.php* där man väljer vilket test man vill redigera. Efter det kommer man till sidan *redigera.php* där man kan välja att skapa fler frågor eller att ta bort frågor (även om denna

funktion inte är implementerad). Från *skapa.php* och *redigera.php* kan man också välja att förhandsgranska test med hjälp av *preview.php*.

3.2 Databasdesign



Figur 3.2-1 ER-diagram över den gamla databasen

Ovan är en översiktsbild över hur den gamla designen såg ut över databasen. Vi har återskapat den enligt den gamla uppsatsen [1]. Denna design har vi vidareutvecklat och en mer ingående beskrivning finns i avsnitt 6.1.

3.3 Problem

Under projektets gång har vi stött på många problem, både väntade och oväntade. Från början tog vi oss an uppgiften i tron om att det befintliga systemet var funktionellt, men det visade sig att flera delar antingen saknades eller var ofärdiga. Enligt uppgift från Donald Ross så har systemet blivit förstört på grund av dataintrång. Nedan beskrivs några av de problem vi har stött på.

3.3.1 Ofullständigt program

Vi trodde att onlinetestsystemet var komplett och fungerande men så var inte fallet. Om detta beror på att någon har förstört systemet i efterhand eller om det aldrig har fungerat vet vi inte, men det system vi fick i våra händer fungerade inte. Koden var väldigt rörig och dåligt kommenterad, så felsökningen blev inte någon enkel uppgift. Det krävdes mycket tid för att förstå sig på filerna. Vi upptäckte flera defekter, bland annat var uppkopplingsanropet till servern med `mysql_connect()` felaktigt. Anropet gjordes, men variabeln som uppgifterna sparades i skickades inte vidare när man anropade databasen med för den fortsatta körningen av programmet. Efter hand började vi dock förstå att koden fortfarande var körbar fastän den var bristfällig. Skälet till att systemet inte fungerade över huvud taget låg någon annanstans.

Vidare så upptäcktes fler brister i systemet. Vid felaktiga svar på testfrågorna så ska en kommentar kunna läggas till av den kursansvarige som skapar testet. Denna funktion fungerade inte alls utan man fick ett felmeddelande på skärmen, och så gick det inte heller att gå tillbaka för att försöka igen. Det var också fel på programmet när man som användare skulle rätta sitt test. Trots att man kryssade i rätt svarsalternativ så sade systemet ibland att man hade svarat fel.

3.3.2 Problem med webbläsare

När vi försökte få igång systemet stötte vi på ett problem som skulle komma att ta mycket tid av vårt arbete. Från början provade vi att köra programmet med Internet Explorer och misslyckades om och om igen med att få det att fungera. Därefter gjorde vi ett försök med Mozilla Firefox och fick ett annorlunda resultat. Med Explorer lyckades vi inte ens komma vidare från första indexsidan i programmet, men med Firefox gjorde vi det. Detta fick oss att göra misstaget att tro att Explorer inte läste PHP-skript men att Firefox gjorde det automatiskt. Att den nästföljande sidan i programmet var rörig och felaktig trodde vi berodde på dålig kod. Källkod visades i webbläsaren och man kunde inte välja kurser vilket det stod i

användarmanualen att man skulle kunna. Felaktigt började vi undersöka koden eftersom vi trodde att felet låg här. Orsaken till att Firefox lyckades komma vidare i programmet var dock en annan. Explorer kunde inte läsa PHP-koden eftersom vi inte körde programmet från en server med PHP-stöd, men Firefox skrev istället ut allt den kunde. Detta misstag gjorde att vi trodde koden var felaktigare än vad den var.

3.3.3 Apache, MySql och PHP

För att kunna köra PHP-skripten behövde vi en server med PHP-stöd. Vi valde att installera en Apache 2.x server. Servern fick vi att fungera utan några större komplikationer, men däremot uppstod problem när vi skulle installera PHP-stödet. Vi laddade ner vad vi trodde var den senaste/lämpligaste versionen av PHP vilken var PHP 5.0.5. Det tog mycket tid att få Apacheservern konfigurerad tillsammans med PHP, och när vi till sist hade klarat av detta så fungerade programmet fortfarande inte fullt ut. Servern kunde nu läsa PHP-skripten men låste sig vid mysql kommandona. Skälet var att PHP 5 inte är uppbyggt på samma sätt som PHP 4 och därmed inte kompatibelt. Vi trodde felaktigt att det bara var en senare version och insåg inte att detta inte var fallet förrän vi till sist beslöt oss för att prova med PHP 4 som har inbyggt MySql-stöd.

3.3.4 Databasen

Till en början kom vi inte åt databasen på enterprise.cse.kau.se. Vi hade fått tillgång till användarnamn och lösenord, men ändå lyckades vi inte få kontakt med den. Vi vände oss till Mari Göransson, som kunde hjälpa oss. Problemet var dock inte över för det. När vi väl lyckades få tillgång till databasen enterprise.cse.kau.se där all data för onlinetesten skulle ligga lagrade så fungerade det inte ändå. Vi började misstänka att databasen saknade tabeller helt och hållet. Efter att ha kontrollerat detta med hjälp av MySQL-klienten EngInSite så bekräftades våra misstankar. Vi blev tvungna att bygga om databasen från grunden med hjälp av figurerna i dokumentationen för det gamla systemet. Olyckligtvis var dock denna information sparsam så detta extraarbete har varit tidskrävande.

4 Resonemang kring det nya systemet

Detta kapitel innehåller de resonemang vi gjorde kring systemet innan vi började vidareutveckla det. I avsnitt 4.1 tar vi upp de krav som bör ställas på systemet, såväl pedagogiska som fysiska. I avsnitt 4.2 tar vi upp de fördelar och nackdelar som finns med att ha ett system för webbaserade duggor.

4.1 Krav

Ett system som ska användas till duggor online kräver naturligtvis god säkerhet [8,9]. Det finns flera kravaspekter som måste tillgodoses. Systemet måste skyddas mot intrång, d.v.s. att frågor och svar måste skyddas från att obehöriga kan läsa dem. Detta gäller både databasen och inloggningen till sidorna. En säker inloggning räcker inte ifall det är möjligt att komma åt uppgifterna genom att göra intrång i databasen och läsa informationen därifrån.

En annan kravaspekt är att kursansvariga måste veta vilka studenter som skriver duggan. Detta är ingen enkel uppgift. Hur kan man veta att den som skriver duggan verkligen är den han eller hon utger sig från att vara? Man måste använda sig av olika inloggningsnamn och lösenord för varje student, men detta är ingen garanti för att med säkerhet veta att inget fusk förekommer. Det finns alltid en risk att lösenord och användarnamn sprids mellan studenter. I slutändan är det den kursansvarige som måste avgöra om systemet är tillräckligt säkert för att användas som en del av kursexaminationen.

Man bör också tänka på sekretessen av de resultat studenterna skriver på duggorna. Ska dessa kunna ses av alla eller bara av lärare och sekreterare på kursen? Eftersom systemet utvecklas för duggor måste studenternas personnummer, namn och resultat sparas i databasen. Känner sig studenter tillfreds med detta?

Förutom säkerhetskraven finns det andra krav som måste uppfyllas. Systemet bör vara lätthanterligt och lättförståeligt. Dessutom finns det krav från studenterna som ska skriva duggorna. I det befintliga systemet så slumpas frågor ut eftersom det endast är ett kunskapstest. Är detta godtagbart även för ett system som ska användas vid examination? Eller tycker studenter att detta är orättvist? En fråga man bör ställa sig är också om de frågor

som används vid duggor ska finnas tillgängliga även vid vanliga kunskapstest, eller om det ska vara helt nya frågor. Möjligheten finns naturligtvis också att blanda frågor från kunskapstesten tillsammans med nya frågor för att på så sätt locka studenterna till att använda systemet mer kontinuerligt.

Ytterligare en pedagogisk aspekt är huruvida frågor ska delas in i olika svårighetsgrader. Använder man sig av slumpvis uttagna duggor så finns ju risken att vissa duggor blir lättare än andra. Frågor bör kategoriseras efter svårighet, så att en lärare kan bestämma hur många lätta respektive svåra frågor som ska finnas på en dugga.

4.2 Fördelar och nackdelar

Vilka fördelar respektive nackdelar finns med att göra duggor online jämfört med att skriva dem i sal? Eftersom Internet är tillgängligt från hemmet är det naturligtvis mycket enklare och smidigare att kunna göra det online, men samtidigt medför det problem. Förutom den ständiga möjligheten att ett datasystem kan krångla finns det flera andra problem. Det första är givetvis att risken för fusk ökar när studenterna inte övervakas då de skriver duggan. Fusk kan i detta fall förekomma på olika sätt. En skicklig hackare skulle kunna bryta sig in i systemet och få tillgång till frågorna och svaren i förväg, men man kan också tänka sig mycket enklare former av fusk. Hur ska man till exempel kunna skydda sig mot att studenter har telefonkontakt medan de skriver? Dessutom kan användarnamn och lösenord spridas så att obehöriga kan logga in och skriva en dugga i en annans namn.

Det finns också andra aspekter man måste tänka på. En av dessa är att systemet som vi har valt att utveckla enbart består av flervalsfrågor. Vissa frågor är svåra att ställa ifall studenten inte kan svara med hjälp av löpande text. Är det godtagbart att duggor enbart består av flervalsfrågor? En annan aspekt är att när en student skriver en dugga hemifrån så kan denne naturligtvis ha böcker såväl som Internet till hjälp vid sidan av. Detta är dock ingen skillnad mot vissa tentor där studenten får ta med precis vilket material han eller hon vill. Det ställer dock högre krav på läraren som ställer frågorna. Är de för lätta kan man enkelt skriva in ett ord i en sökmotor på Internet och direkt få svaret, eller slå i registret på en bok.

Nackdelarna är många, men det finns även fördelar. En dugga är visserligen en sorts tentamen, men den är mindre och inte lika betydelsefull. Med tanke på denna aspekt kan man

tänka sig att offra lite säkerhet till förmån för smidighet, enkelhet och snabbhet. Tentamensal behöver inte bokas och inte heller behöver några tentamenövervakare finnas till hands. Alla resultat sparas i en databas från vilken lärare och sekreterare på kursen kan enkelt få ut dessa. Dessutom förekommer fusk även vid tentor i sal. Vill en student fuska sig igenom en kurs så gör det ingen skillnad om han eller hon får göra duggor hemifrån eller på universitetet.

5 Beskrivning av det nya systemet

För att förändra systemet så att det kan hantera duggor och inte bara kunskapstest har vi behövt förändra många av de filer som programmet bestod av. Förutom detta har vi även strukturerat om samtliga filer så att det är lättare att förstå dem. Avsnitt 5.1 innehåller exempel på gammal kod och exempel på hur vi har förbättrat den. I avsnitt 5.2 beskrivs de förändringar vi har gjort i de gamla PHP-filerna. Avsnitt 5.3 innehåller beskrivningar av de nya PHP-filer vi har skapat.

5.1 Exempel på förbättrad kod

Tidigare har koden varit slarvigt indenterad, dåligt kommenterad och onödigt krånglig. Vi har indenterat alla filer, lagt till kommenterar till funktioner och förenklat många utskrifter.

Exempel på en del av en gammal fil:

```
    }
    ??

    <tr>
        <td><br><strong><? if ($antal_ratt == $antal_fragor) echo 'OTROLIGT bra alla rätt<br>';
        else if($antal_ratt > $antal_fel)=echo 'GRATTIS du fick mer än hälften rätt<br>';
        else if( $antal_fel > $antal_ratt)
echo 'Du bör nog försöka igen<br>';?>Du fick <?echo $antal_ratt?> rätt av <?echo $antal_fragor?> frågor</strong></td>
        </td><td><form action="#">
            <td><br><td><input type=button class="button" value="<-- Tillbaka          " onClick="history.go(-1)"
        </form>
    </td>
    <tr>
        <td><input type="button" class="button" value="<-- Huvudmeny" onClick="location='../index.htm'" "></td>
    </td>
</tr>
<tr >
```

Figur 5.1-1 Gammal kod

I figur 5.1-1 kan man se hur illa indenterad den gamla koden var. Det finns ingen struktur för indrag och radbrytningar. Man kan se hur koden växlar mellan PHP-kod och HTML-kod helt utan anledning.

Den förnyade koden i samma fil:

```
<tr>
  <td><br><strong>
    <?PHP
      if ($antal_ratt == $antal_fragor)
        echo 'OTROLIGT bra alla rätt.<br>';
      else if ($antal_ratt >= $antal_fel)
        echo 'GRATTIS du fick hälften eller fler rätt.<br>';
      else
        echo 'Tyvärr, du fick mindre än hälften rätt.<br>';
      echo 'Du fick ', $antal_ratt, ' rätt av ', $antal_fragor, ' frågor.<br>';
      if( $dugga == '1')
      {
        $personnr = $person['Personnr'];
        $sql = "INSERT INTO Result (Personnr, Test, Result)
        values ('$personnr','$rubrik', '$antal_ratt)";
        $save = mysql_query($sql) or die("Error performing save results: " . mysql_error());
      }
    ?>
  </td>
</tr>
```

Figur 5.1-2 Ny kod

Här har vi strukturerat upp koden med ordentlig indentering. Det finns inte heller några onödiga växlingar mellan PHP-kod och HTML-kod som försvårar förståelsen av koden. De nedre raderna har vi lagt till för att hantera personuppgifter för studenter som ska skriva duggor.

5.2 Ändringar av gamla PHP-filer

Här följer en lista på PHP-filer som vi har ändrat i så att de ska fungera med det nya systemet. Se också avsnitt 3.1.

index.html

Förutom den gamla knappen för att göra kunskapstest innehåller nu förstasidan en knapp för att göra duggor också. Väljer man att göra en dugga länkas man vidare till en login-sida för studenter. Denna nya knapp kan ses i figur 5.2-1.



Välkommen till systemet för kunskapstest online samt onlineduggor.
Välj om du vill göra ett kunskapstest eller en dugga.

Kunskapstest

Dugga

Kurstestansvarig: MPESTV

Figur 5.2-1 Den nya förstasidan

login_new.php

Denna sida hanterade tidigare inloggning för root och kursansvariga. Nu hanterar den även inloggning för studenter som ska skriva duggor. Det är en ren php-fil som kontrollerar användarnamn och lösenord i databasen utan att visa någon HTML-kod. Beroende på vilken behörighet användaren har så länkas denne vidare till olika sidor i systemet.

student.php

Filen student.php har nu ett hidden-fält¹ som kontrollerar om det är en dugga eller ett kunskapstest som ska göras. Denna variabel är till för att studenter inte ska kunna komma åt duggor när de vill göra vanliga kunskapstest.

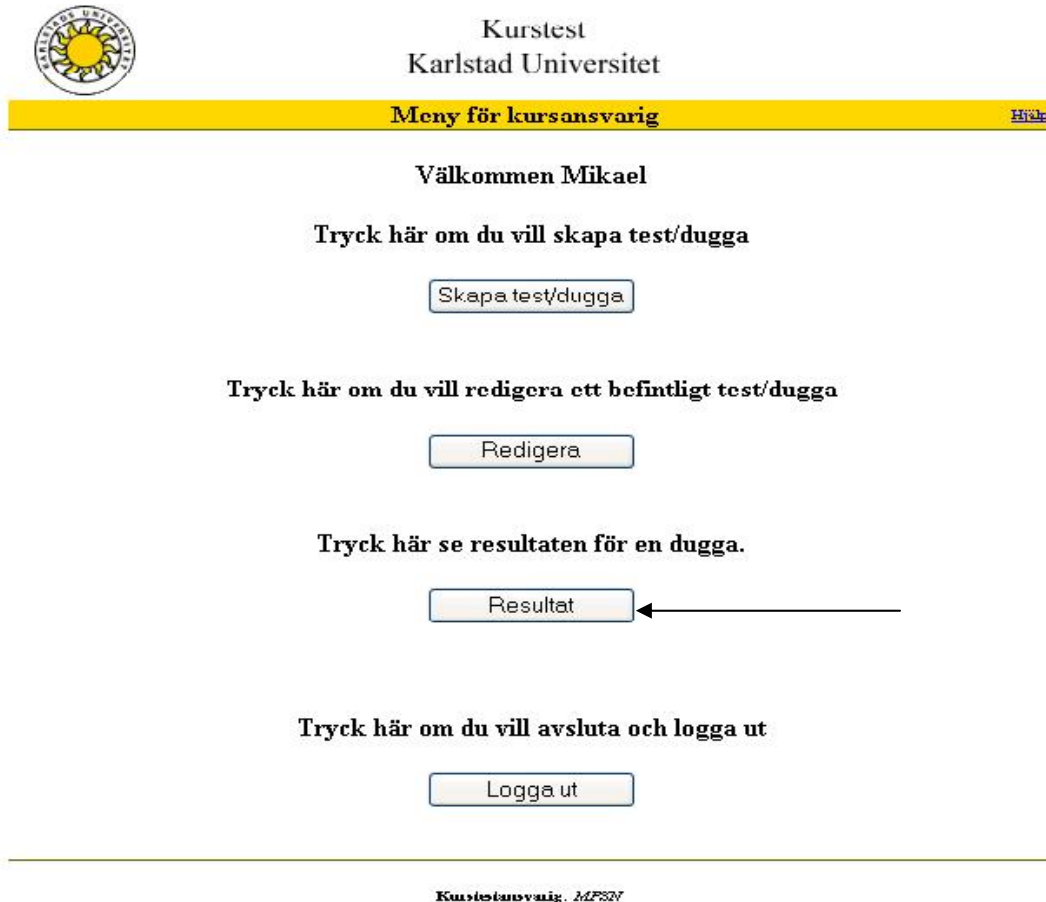
student1.php

Filen student1.php har nu ett hidden-fält som kontrollerar om det är en dugga eller ett kunskapstest som ska göras. Denna variabel är till för att studenter inte ska kunna komma åt duggor när de vill göra vanliga kunskapstest. Indikerar hidden-fältet att testet är en dugga så länkas man vidare till den nya sidan show_result.php som skriver ut studenternas resultat för duggan när man trycker på knappen välj test, se avsnitt 5.2.

¹ Textfält i hmtlkod kan väljas till att vara gömda – "hidden". Med detta menas att användaren av systemet inte kan fylla i något i fältet. Dessa fält används för att skicka information som inte användaren in kan ändra.

usermeny.php

Filen usermeny.php har nu en knapp som länkar till en ny sida där studenternas resultat på en specifik dugga visas, se figur 5.2-2.



Figur 5.2-2 Ny sida för kursansvariga

rubrik.php

Denna sida innehåller nu ett val i form av två radioknappar² där man får fylla i om man vill skapa ett kunskapstest eller en dugga, se figur 5.2-3.

² Radioknappar i HTML visas som runda, ifyllbara knappar. De används i regel för att ställa frågor till användaren av systemet.



Välj vilken kurs du vill skapa test för
och sätt en rubrik

DSA Test: Test: Dugga:
Rubrik: ManTest Skapa

Ta bort test: ManualD Ta bort

Visa alla test

[← Huvudmeny](#)

Kurstestansvarig: MPSEN

Figur 5.2-3 Ny sida för att skapa test

skapa.php

Denna sida markerar nu en etta i fältet Type i tabellen Test i databasen om man har valt att skapa en dugga, och en nolla om man har valt att skapa ett kunskapstest.

redigera.php

Filen redigera.php hämtar nu information från databasen om för den dugga eller det kunskapstest man vill redigera. Är det en dugga så visas en knapp som länkar till den nya sidan registrera studenter för en dugga, se figur 5.2-4.



Redigera fråga: 1

Hur många Mbyte kan du ladda ner på 10 sekunder om du har en 10Mbits uppkoppling

Redigera dina svarsalternativ:

Svar:

- | | | |
|----|------|----------------------------------|
| A: | 100 | <input checked="" type="radio"/> |
| B: | 125 | <input type="radio"/> |
| C: | 10 | <input type="radio"/> |
| D: | 12,5 | <input type="radio"/> |

Kommentar till testpersonen:

Tänk på att dela med

- Uppdatera
- Lägg till fråga
- Ta bort fråga
- Registrera student
- Förhandsgranska

- ← Tillbaka
- ← Huvudmeny

Figur 5.2-4 Ny sida för att redigera test

ratta.php

Filen ratta.php kontrollerar ifall det är en dugga eller ett kunskapstest som ska rättas. Om det är en dugga så sparas studentens personnummer, duggans id och studentens resultat till databasen.

delfraga.php

Denna fil är till för att ta bort frågor från databasen. Den var tidigare inte implementerad utan visade bara en text i fönstret om vilken fråga man ville ta bort. Funktionen är nu implementerad så att frågan också tas bort från databasen, se figur 5.2-5.

Kurstest
Karlstad Universitet

Ta bort fråga ur testet: Manual

Välj vilka frågor som ska tas bort

Fråga 1	Är detta en fråga till manualen	<input type="checkbox"/>
---------	---------------------------------	--------------------------

Ta bort

<- Tillbaka

Kurstestansvarig: MPESV

Figur 5.2-5 Ny sida för att ta bort frågor från ett test

5.3 Nya PHP-filer

Här följer en beskrivning av de nya PHP-filer vi har lagt till i systemet.

loginstudent.php

På denna sida loggar studenter in med användarnamn och lösenord för att göra en viss dugga. Det finns ett fönster med ett avtal som studenten måste godkänna innan inloggning kan ske. Användarnamn och lösenord skickas vidare till filen login_new.php (se avsnitt 5.1) som kontrollerar med databasen ifall studenten är registrerad på en dugga. Är användaruppgifterna korrekta så länkas studenten vidare till en dugga, annars kommer man tillbaka till inloggningssidan. Denna sida visas nedan i figur 5.3-1.

 Nej, jag accepterar inte: '. There are two input fields for 'Användarnamn:' and 'Lösenord:', followed by a 'Logga in' button. At the bottom, there is a link for 'Glömt lösenord?'. The footer contains the text 'Kurstestansvarig: MPSSN'."/>

 Kurstest
Karlstad Universitet

[Testsida] **Inloggning för studenter.** [Hjälp](#)

Använd ditt tilldelade användarnamn och lösenord för att skriva duggan.

Avtal för att göra tentamen vid KAU.

1. Det är ditt personliga användarnamn & lösenord.
2. Du gör tentamen endast med tillåtna hjälpmedel.

Lycka till!!!!

Jag accepterar avtalet: Nej, jag accepterar inte:

Användarnamn:

Lösenord:

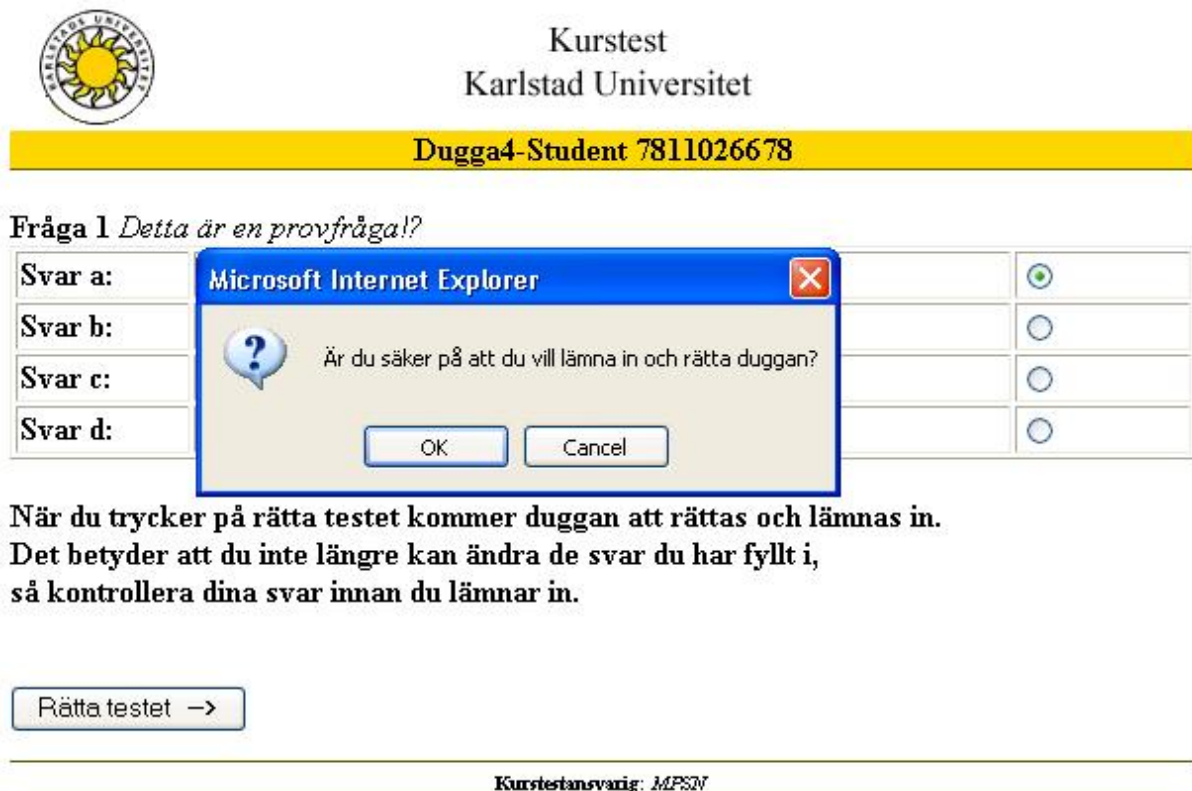
[Glömt lösenord?](#)

Kurstestansvarig: MPSSN

Figur 5.3-1 Inloggningssida för studenter

dugga.php

Sidan dugga.php skriver ut alla frågor och svar till den dugga som matchar till det användarnamn och lösenord som studenten har loggat in med. På slutet av sidan finns en knapp för rättning som länkar till ratta.php, där resultatet sedan sparas. Innan länken utförs får studenten en extra "popup-ruta" som frågar om duggan verkligen är klar att lämnas in, se figur 5.3-2.



Figur 5.3-2 En dugga

listastud.php

Sidan listastud.php skriver ut alla studenters förnamn, efternamn, personnummer, användarnamn och lösenord som är registrerade för den dugga som man har valt att lista studenter för. Denna lista kan skrivas ut av kursansvarig eller sekreterare så att studenterna sedan kan hämta deras användarnamn och lösenord.

registrera.php

På denna sida kan kursansvarig eller sekreterare registrera studenter för en dugga. Studentens förnamn, efternamn och personnummer ska fyllas i och denna information skickas sedan till


sparastudent.php där informationen sparas tillsammans med ett genererat användarnamn och lösenord. Personnumret måste fyllas i med tio siffror utan bindestreck. På sidan finns även en knapp som länkar till listastud.php där man kan se vilka studenter som hittills är registrerade (se ovan).

sparastudent.php

Denna sida innehåller enbart PHP-skript och skriver inte ut något till skärmen. Den tar emot en students förnamn, efternamn och personnummer, och genererar ett användarnamn på åtta tecken och ett lösenord på åtta tecken. Denna information sparas till tabellen Student i databasen. Efter att informationen är sparad återvänder programmet till registrera.php.

show_results.php

Detta är en sida som skriver ut en lista på studenters personnummer och deras resultat för en specifik dugga. Det är en praktisk resultatlista som en sekreterare på kursen kan skriva ut och hänga på anslagstavlan, se figur 5.3-3.



Kurstest
Karlstad Universitet

Resultat ManualD	
Personnummer	Resultat
1234567890	1
1234569870	1

Kurstestansvarig: MPSEN

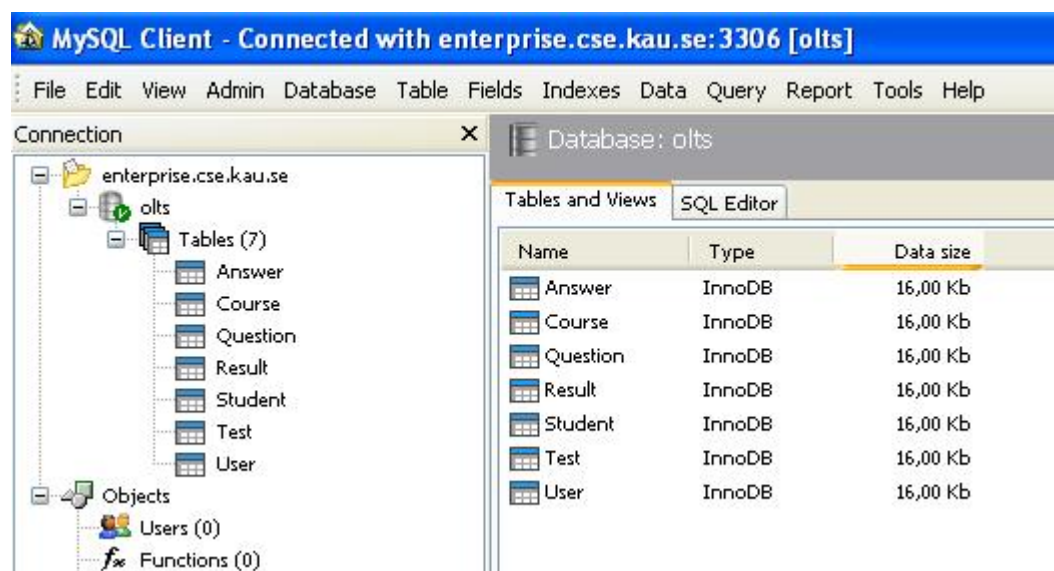
Figur 5.3-3 Resultatlista

6 Beskrivning av den nya databasen

I detta kapitel beskrivs den nya databasen. I avsnitt 6.1 beskrivs databasen översiktligt, medan det i avsnitt 6.2 finns utförligare beskrivningar av databasens tabeller.

6.1 Allmänt

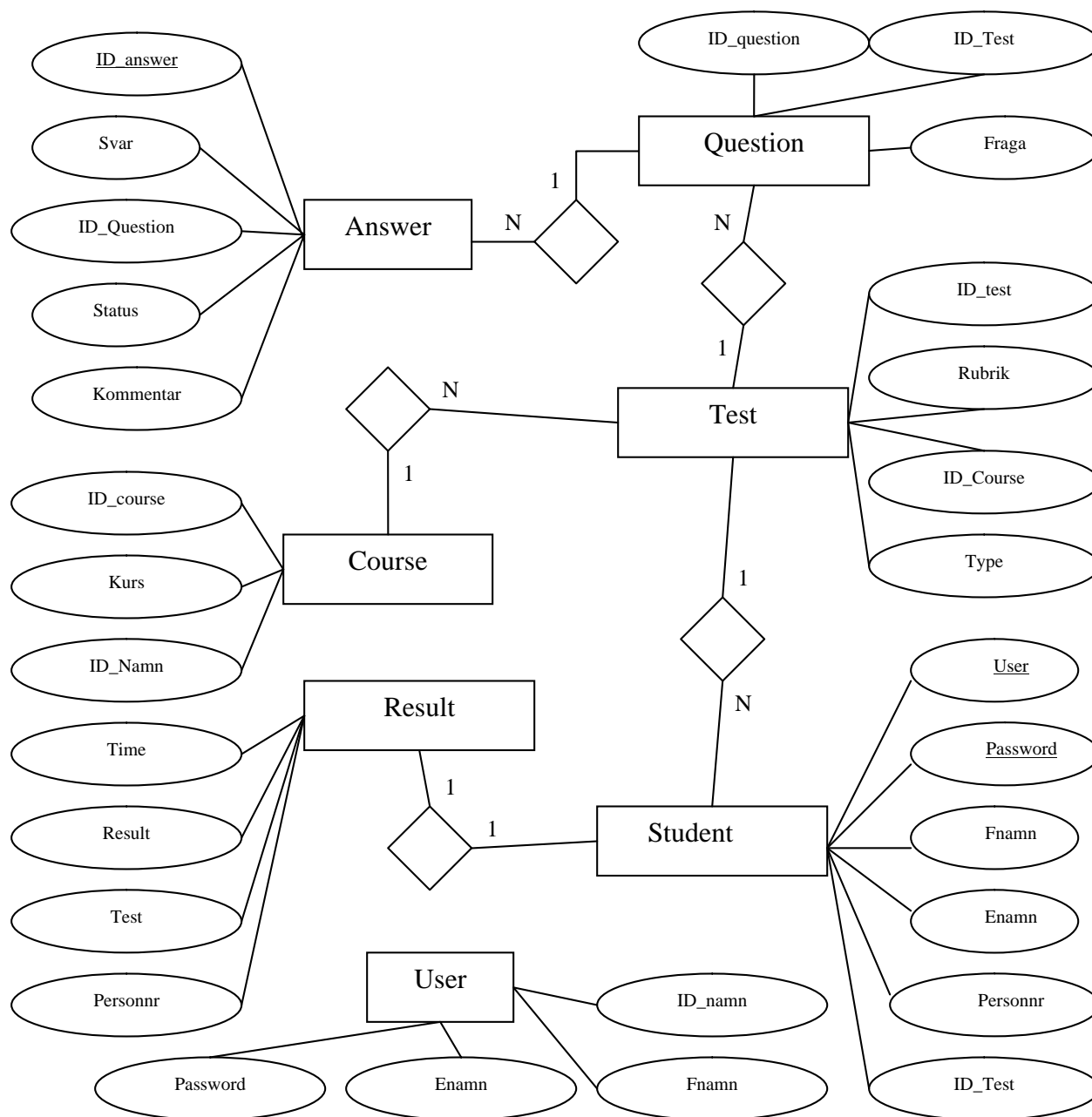
Den nya databasen bygger till viss del på den information som vi hade om databasen till det gamla systemet. Men då vi har byggt ut systemet för att användare ska kunna göra både kunskapstest och duggor, har vi varit tvungna att bygga ut databasen för att kunna använda oss av de nya funktionerna. Vi har lagt till två nya tabeller – *Student* och *Result*. Student är en tabell som lagrar användarinformation om studenter. Result är en tabell där en students resultat för en viss dugga lagras. I figur 6.1-1 nedan syns alla de tabeller som ligger på databasen enterprise.cse.kau.se.



Name	Type	Data size
Answer	InnoDB	16,00 Kb
Course	InnoDB	16,00 Kb
Question	InnoDB	16,00 Kb
Result	InnoDB	16,00 Kb
Student	InnoDB	16,00 Kb
Test	InnoDB	16,00 Kb
User	InnoDB	16,00 Kb

Figur 6.1-1 Översikt av databasens tabeller

ER-diagram



Figur 6.1-2 EntityRelations-diagram över databasen.

Här kan man se hur de olika tabellerna (entiteterna) är kopplade till varandra, samt vilka attribut de innehar. Entiteter visas som rektanglar medan deras attribut utgörs av ovala ringar. De attribut som är primärnycklar är understrukna.

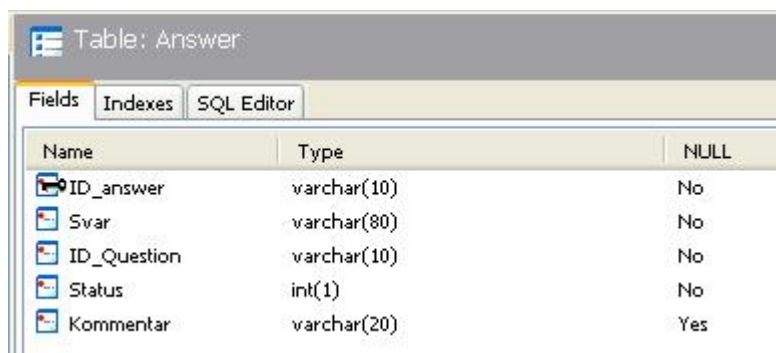
En fråga har flera olika svar, ett test har flera frågor och en kurs kan ha flera test. Vidare har ett test flera studenter, men i detta fall kan inte en student göra flera test eftersom vår design

är gjord så att tabellen student innehåller en specifik student på ett specifikt test. Tabellen User som innehåller kursansvariga är för tillfället inte kopplad tabellen Course. Alla kursansvariga har tillgång till alla kurser.

6.2 Tabeller i databasen

6.2.1 Answer

Answer är tabellen där alla svar till frågor lagras. Den består av fem kolumner. Den första kolumnen är **ID_answer** som är primärnyckel och unik för varje svar. Nästa kolumn är **Svar**, där lagras svaret i ren text. Nästa kolumn är **ID_Question** som är en främmandenyckel för att man ska veta till vilken specifik fråga svaret hör till. **Status** är en kolumn för att visa om detta är rätt eller fel svarsalternativ. I den gamla versionen så markerades rätt svar med "on" och fel svar av NULL (värde saknas). Eftersom man i databaser bör undvika NULL-värden beslöt vi oss för att ändra till 1 för rätt svar och 0 för fel svar. Sista kolumnen är **Kommentar**, här sparas en kommentar som kan visas för en användare av systemet om studenten svarat fel på en fråga. Denna kommentar är inte obligatorisk och kan därför utelämnas. Tabellen Answer visas i figur 6.2-1 nedan.



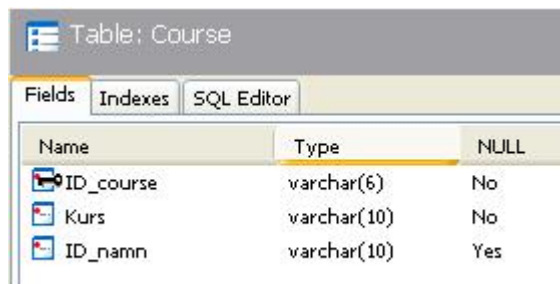
Name	Type	NULL
ID_answer	varchar(10)	No
Svar	varchar(80)	No
ID_Question	varchar(10)	No
Status	int(1)	No
Kommentar	varchar(20)	Yes

Figur 6.2-1 Tabellen Answer

6.2.2 Course

Course är tabellen där kurser som finns registrerade i systemet sparas. Den har tre olika kolumner, men det är bara två som används för tillfället. Första kolumnen är primärnyckeln **ID_course** som är designad efter Karlstads Universitet kursbeteckningar. Den består av fyra bokstäver och två siffror (DAVB01). **Kurs** är namnet på kursen t.ex. DSA. Sista kolumnen är **ID_Namn**. Denna kolumn är inte implementerat i systemet, men den är tänkt att användas för

att vissa lärare i tabellen User ska höra till en viss kurs. Eftersom detta inte är implementerat så kan kolumnen lämnas tom. Tabellen Course visas i figur 6.2-2.

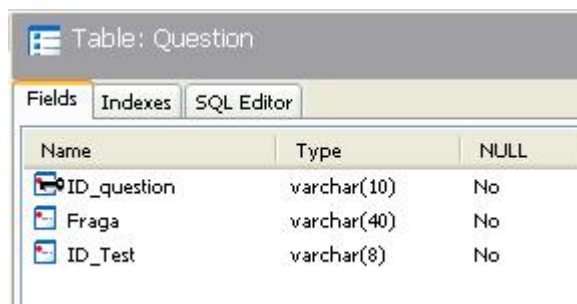


Name	Type	NULL
ID_course	varchar(6)	No
Kurs	varchar(10)	No
ID_namn	varchar(10)	Yes

Figur 6.2-2 Tabellen Course

6.2.3 Question

Question är en tabell som sparar frågor tillhörandes ett visst test. **ID_question** är primärnyckeln i denna tabell. **Fraga** är en fråga som sparats ner som ren text, och **ID_Test** är främmandenyckeln som visar till vilket Test frågan tillhör. Inga fält får vara tomma i denna tabell. Tabellen Question visas i figur 6.2-3 nedan.



Name	Type	NULL
ID_question	varchar(10)	No
Fraga	varchar(40)	No
ID_Test	varchar(8)	No

Figur 6.2-3 Tabellen Question

6.2.4 Result

Tabellen Result är till för att spara en students resultat för en viss dugga. Denna tabell består av en sammansatt nyckel, där Personnr är primärnyckel och Test är sekundärnyckel, då ett Personnr kan komma att sparas flera gånger men endast en gång per dugga. **Personnr** sparas som ett tiosiffrigt tal, detta görs automatiskt när en student rättat sin skrivna dugga. **Test** är den dugga som studenten skrivit och rättat, och **Result** är det antal rätt studenten fick på duggan. Slutligen har vi lagt in en kolumn **Time**. Detta är en tidstämpel som sparar ner datum och tid då duggan lämnades in. Tabellen Result visas i figur 6.2-4.

Name	Type	NULL
Personnr	bigint(10)	No
Test	varchar(10)	No
Result	int(3)	No
Time	timestamp	No

Figur 6.2-4 Tabellen Result

6.2.5 Student

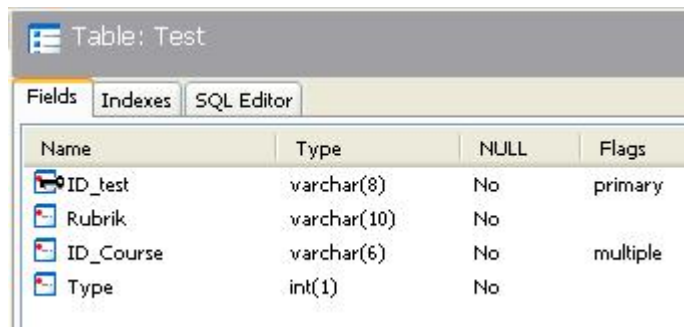
Student är en tabell för att lagra de studenter som är registrerade för en viss dugga. **User** och **Password** är användaruppgifter som genereras till en viss student när denna registreras. **Personnr** är en students tiosiffriga personnummer. **ID_Test** är främmandenyckel och visar vilken dugga som användaruppgifterna tillhör. **Fnamn** är förnamnet på studenten, och **Enamn** är efternamnet på studenten. Den här tabellen används när man vill lista vilka studenter som är registrerade på en viss dugga. Tabellen Student visas i figur 6.2-5 nedan.

Name	Type	NULL	Flags
User	varchar(10)	No	primary
Password	varchar(10)	No	primary
Personnr	bigint(10)	No	
ID_Test	varchar(8)	No	multiple
Fnamn	varchar(20)	No	
Enamn	varchar(20)	No	

Figur 6.2-5 Tabellen Student

6.2.6 Test

Test är en tabell för lagrade skapade test/duggor. **ID_test** är primärnyckeln som identifierar testen/duggorna. **Rubrik** är namnet på ett test. **ID_Course** är främmandenyckel och visar vilken kurs testet tillhör. **Type** är en variabel för att hålla koll på om det är ett vanligt test eller en dugga. 1 indikerar att det är en dugga, och 0 att det är ett kunskapstest. Tabellen Test visas i figur 6.2-6.

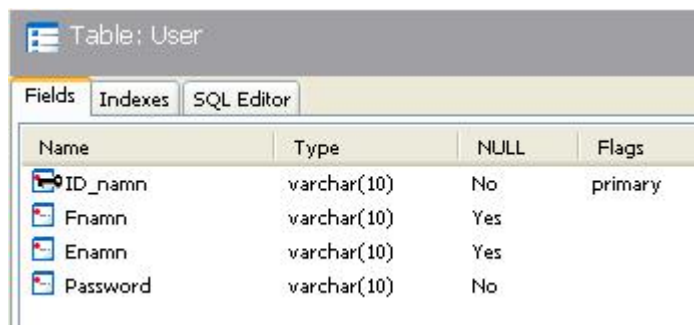


Name	Type	NULL	Flags
ID_test	varchar(8)	No	primary
Rubrik	varchar(10)	No	
ID_Course	varchar(6)	No	multiple
Type	int(1)	No	

Figur 6.2-6 Tabellen Test

6.2.7 User

User är en tabell som lagrar användare av systemet, det vill säga lärare och administratörer. Denna tabell är lite mindre och enklare än Student då man inte behöver lika mycket uppgifter. **ID_namn** är primärnyckel och i denna finns användarnamnet sparad. **Fnamn** är förnamnet för personen, och **Enamn** är efternamnet. **Password** är lösenordet som valts av användaren. Tabellen User visas i figur 6.2-7 nedan.



Name	Type	NULL	Flags
ID_namn	varchar(10)	No	primary
Fnamn	varchar(10)	Yes	
Enamn	varchar(10)	Yes	
Password	varchar(10)	No	

Figur 6.2-7 Tabellen User

7 Enkätundersökning

För att få reda på vad studenter tycker om att göra duggor online har vi genomfört en enkätundersökning för studenter som har läst kursen Datakommunikation 1 under höstterminen 2005 (studenterna genomförde inte duggan hemifrån utan i en datasal på universitetet). Förutom att utveckla systemet tyckte vi att det vore intressant att få reda på om det fanns intresse för onlineduggor hos studenterna. Vår handledare Katarina Asplund var också lärare på datakommunikationskursen, så hon använde vårt system till att genomföra en av duggorna på kursen. Vi skrev en enkät på två sidor med frågor vi ville ha svar på och delade ut dessa i samband med att studenterna skrev duggan. Av de 17 studenter som skrev duggan deltog 14 i undersökningen.

Frågorna delade vi upp i två olika kategorier. Den första kategorin innehöll generella frågor om att skriva duggor online, men den andra kategorin innehöll frågor om vårt specifika system. Enkäten finns i bilaga B.

7.1 Frågor

I den första kategorin av frågor ville vi få svar på vad studenter tycker om att skriva duggor på nätet jämfört med att skriva dem som vanligt i sal, och hur säkert de tycker att detta låter. Enkäten i sin helhet finns i bilaga B. Den första frågan är enkel och precis – ”Skulle du hellre skriva duggor på nätet hemifrån än med papper och penna i sal?” Denna fråga tycker vi är den viktigaste av alla. Vill studenter ha ett system för att skriva duggor online eller inte? Efter denna fråga har vi gett dem två rutor där de kan fylla i de fördelar och nackdelar de kan komma på som en följd av första frågan. Efter det har vi frågat huruvida studenterna är oroliga för att deras resultat tillsammans med personuppgifter kan komma åt av andra, samt ifall de tror att mer fusk skulle förekomma vid onlineduggor jämfört med ”vanliga” duggor. Som sista fråga i den första kategorin har vi frågat ifall studenten har tidigare erfarenheter av att skriva duggor online.

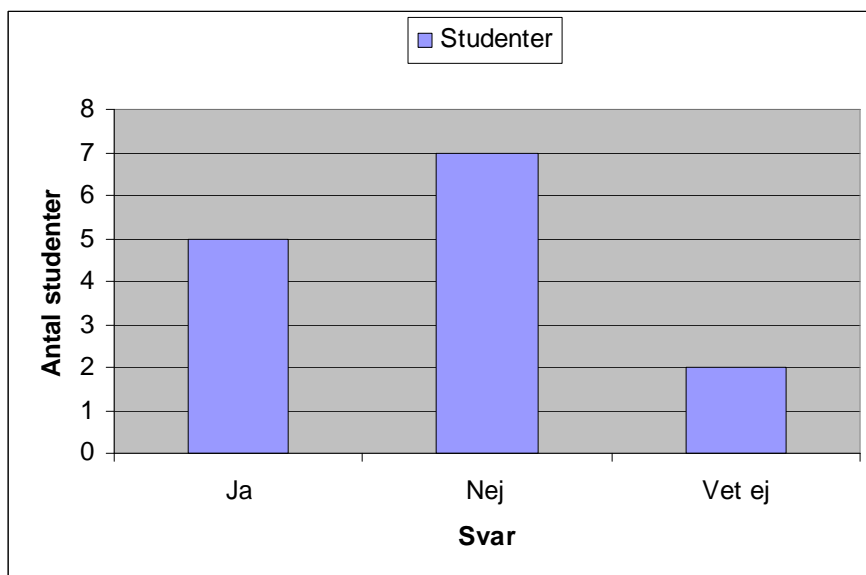
Den andra kategorin innehåller frågor som gäller vårt specifika system. Vi ställer frågor angående vad studenterna tyckte om användargränssnittet, funktionalitet, säkerhet och vilket

övergripande intryck de fick av systemet. Vi har även här gjort två tomma rutor där studenterna kan fylla i de fördelar och nackdelar de har upptäckt. Slutligen har vi också ett antal blanka rader där de som vill kan fylla i kommentarer och förhoppningsvis konstruktiv kritik.

7.2 Resultat

7.2.1 Generella frågor om att skriva duggor på nätet

i) Skulle du hellre skriva duggor på nätet hemifrån än med papper och penna i sal?



ii) Nämn de fördelar respektive nackdelar du kan komma på med att skriva duggor online.

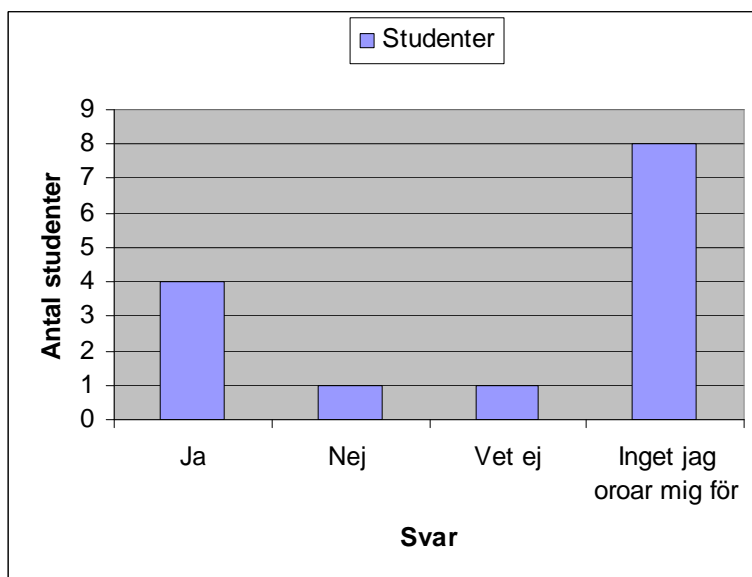
Fördelar:

Många av studenterna tyckte att ett webbaserat system för att skriva duggor är ett snabbt, enkelt och smidigt alternativ till penna och papper. Det var även många som tyckte om att kunna befinna sig i sin hemmiljö när de skriver duggan. En annan fördel som ett antal studenter svarat var att det var bekvämt att få resultatet direkt när man skrivit duggan.

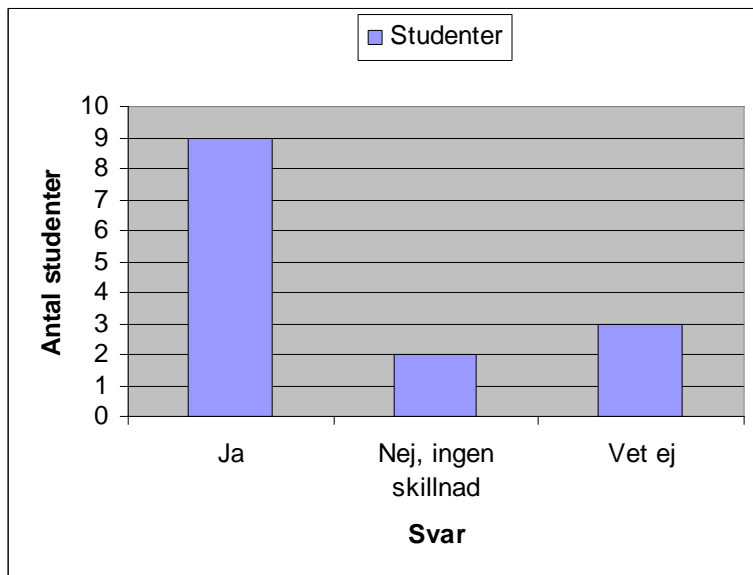
Nackdelar:

Som nackdel hade en stor del av studenterna svarat att de var rädda för att fusk skulle förekomma i större utsträckning. Vissa var även rädda för program- och användarfel så att information skulle försvinna. Det var även en student som svarade att systemet framkallade nervositet. En annan student skrev att det var svårt med flervalsfrågor då det inte går att förklara sina svar.

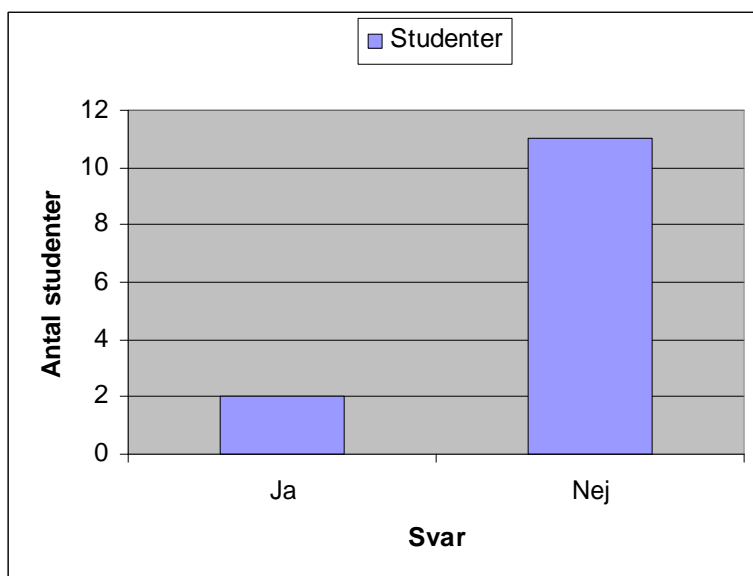
iii) Skulle du känna dig säker med att skriva duggor online, i fråga om sekretess? D.v.s. att ditt resultat och personuppgifter hålls hemliga.



iv) Tror du att mer fusk skulle förekomma om studenter ges möjligheten att skriva duggor hemifrån?



v) Har du några tidigare erfarenheter av att skriva tentamen eller duggor online?



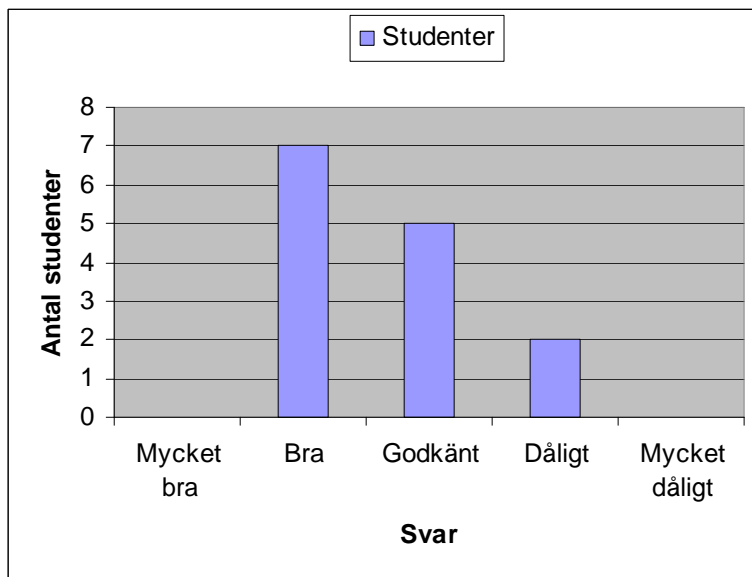
Kommentarer på studenternas svar på generella frågor:

Det var intressant att så många inte skulle vilja skriva duggor hemifrån. På denna fråga förväntade vi oss att fler skulle svara ja. Även om många tyckte att ett system för webbaserade duggor visserligen är enkelt och smidigt, så var de oroliga för att mycket fusk skulle förekomma, och att det skulle kunna uppstå fel i systemet så att information gick förlorad. Av detta drar vi slutsatsen att ett webbaserat system för duggor bör vara så

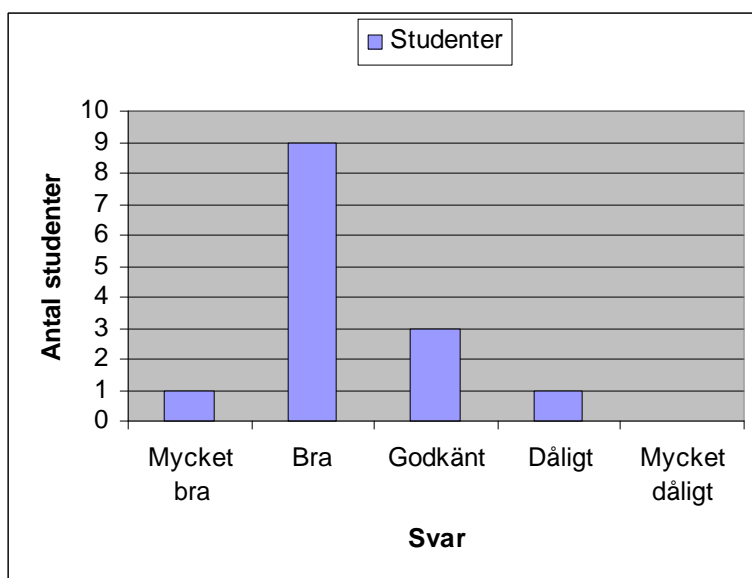
förtroendeingivande som möjligt innan alla kan uppskatta det. Det var dock bara en student som svarade att denne var orolig för att resultat och personuppgifter inte hölls hemliga. Flera studenter svarade också att de inte vill ha duggor som endast innehåller flervalsfrågor. Dessa kommentarer förväntade vi oss, då detta är en uppenbar nackdel med webbaserade system.

7.2.2 Frågor om det system du har provat

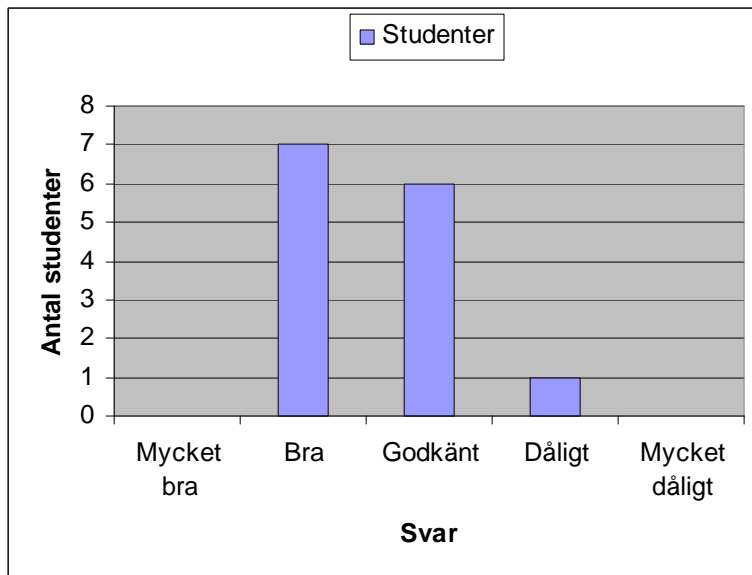
i) Hur var ditt övergripande intryck av systemet?



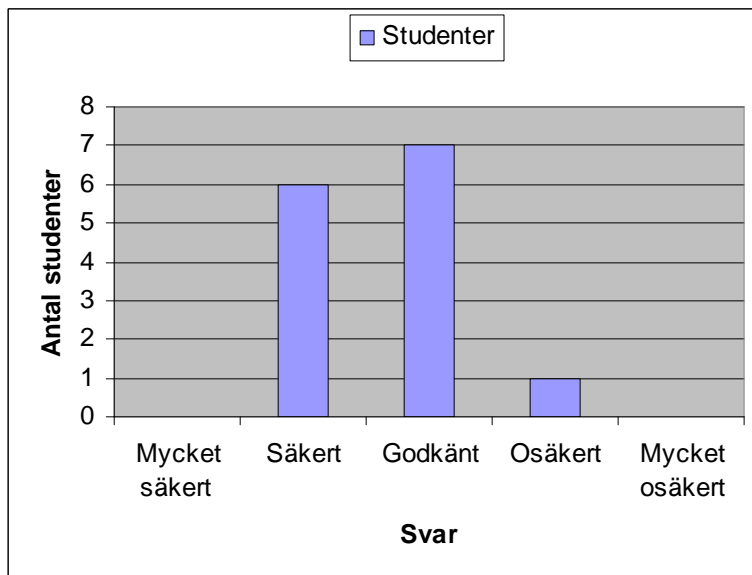
ii) Vad tyckte du om användargränssnittet?



iii) Vad tyckte du om funktionaliteten?



iv) Fick du intrycket av att systemet var säkert?



v) Nämn de fördelar respektive nackdelar du upptäckte i systemet.

Fördelar:

Även här kom kommentarer som snabbt och enkelt. Någon tyckte att designen för frågorna var bra, och att man fick svaret direkt.

Nackdelar:

Här var det många som svarat att de ville ha möjligheten att inte svara på en fråga. Det var även någon som tyckte att systemet skulle generera enklare lösenord. Någon svarade att man inte kan ha en dugga med enbart flervalsfrågor och ville därför ha med textfrågor, samma student tyckte även att det skulle finnas automatisk utskrift vid rättning.

Kommentarer till studenters svar på frågorna angående vårt system:

Det var positivt att så många studenter svarade att de tyckte om systemet. Det var endast en student som tyckte att systemet var genomgående dåligt, vilket vi är väldigt nöjda med. De flesta tyckte att systemet var lätthanterligt. Detta är en viktig egenskap hos ett sådant här system.

8 Diskussion

I detta kapitel för vi en diskussion om olika frågor som rör systemet och dess utformning. Bland annat systemets design, och hur man skulle kunna vidareutveckla det. I avsnitt 8.1 gör vi en sammanfattning av hur det nya systemet fungerar idag.

8.1 Resultat

Meningen med det nya systemet var att vi skulle kunna använda det vid en riktig dugga för en kurs vid Karlstads Universitet. Vi har nu genomfört detta på Katarina Asplunds kurs i Datakommunikation 1. När studenterna hade använt vårt system för att skriva sin dugga så delade vi även ut en enkät som vi lät studenterna fylla i. Resultatet av denna enkät tas upp i kapitel 7 där vi sammanställt alla frågor. Några svar och synpunkter från enkäten tas även upp i avsnitt 8.2 om vidareutveckling. Vårt system är utvecklat så att det kan användas vid duggor, d.v.s. mindre tentor. Först funderade vi på att blanda frågorna, för att på så sätt få ett system där studenter lockas även till att göra kurstesten, men efter att ha överlagt bestämde vi oss för att separera frågorna till kurstesten och duggorna totalt. På detta sätt kan inte studenter göra kurstesten om och om igen tills de har lyckats slumpa fram samtliga frågor och kan dem utantill när de skriver duggan. Vi har även slumpmässig utplacering av frågorna på en dugga, så även om alla som skriver en specifik dugga har samma frågor, så ligger dessa i slumpmässig ordning. Detta för att skapa lite extra säkerhet mot fusk. Om två studenter har telefonkontakt så kan de inte bara läsa upp ”den rätta raden” för varandra eftersom frågorna ligger i slumpmässig ordning. Vill de fortfarande fuska måste de läsa upp frågorna i sin helhet för varandra, vilket är mycket mer tidskrävande. Har man då en tidspress så kan kanske detta minska risken för fusk.

För tillfället kan man bara ha flervalsfrågor med ett korrekt svar. Varje fråga har fyra svarsalternativ där ett är korrekt och övriga är felaktiga. Man kan skriva ett kunskapstest utan att vara registrerad på en kurs. Om man är registrerad för en dugga och gör den online, sparas resultatet ner i databasen. Detta gör det lätt för en lärare att senare lista alla resultat. Vi har skapat en sida där läraren kan kontrollera alla resultat för en dugga, där personnummer och resultat skickas till webbläsaren. Vårt system är uppbyggt så att det finns en huvudansvarig systemadministratör som ansvarar för systemet. Det är dennes ansvar att registrera lärare och

kurser på systemet. Om en lärare glömt lösenordet eller vill ändra sitt lösenord, måste det gå genom systemadministratören. Det är däremot lärarens ansvar att registrera studenter till en viss dugga. Systemet genererar ett användarnamn och ett lösenord till en student i samband med att studenten registreras för en dugga. Detta för att studenten ska ha minimalt tillträde till systemet. De genererade användarnamnen och lösenorden kan sedan studenten komma till sekreteraren och hämta mot uppvisad legitimation.

8.2 Vidareutveckling

I detta kapitel tas några frågor upp om vidareutveckling av systemet. Synpunkterna som tas upp är sådana som vi själva kommit på och sådana som kommit från kritik vi fått från studenterna som använt systemet.

8.2.1 Säkerhet mot fusk

När man talar om säkerhet finns det mycket man kan vidareutveckla. Frågan man måste ställa sig är, hur mycket säkerhet behöver man med tanke på informationen som finns lagrad? En säkerhetsaspekt som vi tänkt på men inte implementerat är att även ha ett tidslås. Detta skulle kunna fungera på så sätt att en student som ska göra en dugga bara kan logga in och använda systemet inom en viss tidsram.

8.2.2 Design

Man skulle kanske vilja sätta en spärr, så att en lärare endast kan komma åt sina egna kurser för modifieringar. Som det är i nuläget kan alla lärare modifiera alla kurser. Detta är redan påbörjat men inte implementerat fullt ut. En annan sak är att man skulle kunna skapa olika många svarsalternativ, istället för att bara ha fyra. Man skulle även kunna utveckla det så att flera svarsalternativ skulle kunna vara korrekta, och att alla dessa måste vara ifyllda för korrekt svar. Om man skulle vilja ha annat än flervalsfrågor kan man implementera det, men då måste man komma ihåg att automatisk rättning inte kan tillämpas och det är en viktig del i ett sådant här system.

En annan idé är att implementera en funktion som automatisk sparar ner hela duggan med frågor och svar i samband med rättning. Man skulle också kunna tänka sig att det också finns en automatisk funktion som skickar den rättade duggan till både studenten och läraren med e-post.

8.2.3 Kodning

Om systemet ska byggas ut ytterligare tror vi att det kan vara bra att använda objektorienterad programmering, för att minska koden och underlätta för återanvändning.

9 Summering

Målet med examensarbetet var att vidareutveckla det webbaserade systemet så att det kan användas till duggor vid Karlstads Universitet. Det system vi har vidareutvecklat hade inte denna funktionalitet. Vi har byggt ut det förra systemet då detta både saknade en databas och inte hade någon säkerhet. Alla nya funktioner och liknande finns beskrivna i kapitel 5. Resultatet av utvecklingen är att systemet nu kan hantera duggor såväl som kunskapstest.

När systemet var klart genomförde studenterna som läst Datorkommunikation 1, en dugga i det nya systemet. Detta gick bra då vi inte hade några tekniska problem. När studenterna använt systemet, lät vi dem fylla i en enkät som hittas i bilaga B. De svar och kommentarer vi fick har vi sedan sammanställt i kapitel 7. Systemet fungerade att göra duggor med, men om man ska ta det i bruk kanske viss vidareutveckling behöver tillämpas på systemet.

Referenser

- [1] Staffan Nilsson och Mattias Pehrsson. Online Självvärderingskunskapstest, Examensarbete, C-nivå, 2004:17, Karlstads Universitet.
- [2] MySQL AB, www.mysql.com, 2006-01-10.
- [3] World Wide Web Consortium, www.w3c.org, 2006-01-10.
- [4] W3Schools Online Web Tutorials, www.w3schools.com, 2006-01-10.
- [5] Danny Goodman. *Dynamic HTML: The Definite Reference*. O`reilly & Associates, Inc, 2nd Edition. 2002.
- [6] Tim Converse and Joyce Park. *PHP 4 Bible*. IDG Books Worldwide, Inc, 2000.
- [7] Alexander Sasha Pachev. *MySQL, enterprise solutions*, Wiley publishing, Inc, 2003.
- [8] William Stallings. *Network Security Essentials, applications and standards*, Prentice Hall, Inc, 2000.
- [9] Rolf Oppliger, *Security Tehnologies for the world wide web*, ARtech House, Inc, 2000.

A Manual

Användarmanual för systemet. Manualen delas in i tre delar: en del för att skriva kunskapstest (student), en del för att skriva en dugga (student) och en del Administratörer/Lärare. För en manual till root (ansvarig för systemet) hänvisar vi till den tidigare skrivna manualen[1].

A.1 Kunskapstest

Kunskapstest är den del som alla personer kan göra utan att vara registrerad på en kurs.

A.1.1 index.htm



Kurstest
Karlstad Universitet

[\[Administratör\]](#) Online testsystem [Hjälp](#)

Välkommen till systemet för kunskapsstest online samt onlineduggor.
Välj om du vill göra ett kunskaps test eller en dugga.

Kunskapstest

Dugga

Kurstestansvarig: MPSTV

Detta är den första sidan man stöter på när man startar systemet. För att göra ett kunskapstest trycker man på knappen *Kunskapstest* för att gå vidare i systemet.

A.1.2 student.php



Kurstest
Karlstad Universitet

Kunskaps test

[Hjälp](#)

Välj kurs!

Datacom 1

Välj ämne

[← Tillbaka](#)

Kurstestansvarig: *MFSN*

Nu har du kommit till denna sida, och då är det dags att välja för vilken kurs du vill göra ett kunskapstest. Börja med att välja kurs i *menyfönstret* och tryck sedan på knappen *Välj ämne* för att gå vidare.

A.1.3 student1.php



Kurstest
Karlstad Universitet

Kunskaps test [Hjälp](#)

Välj test

Manual ▼ Välj test

← Tillbaka

Kurstestansvarig: MFSN

Du har redan valt en kurs, så nu ska du välja ett specifikt test för den kursen. Välj i *menyfönstret* det test du vill göra, och tryck sedan på knappen *Välj test* för att gå vidare.

A.1.4 antal_fragor.php



Kurstest
Karlstad Universitet

Kunskaps test

[Hjälp](#)

Det finns 1 frågor att välja från i detta test.
Det antalet frågor du väljer slungas från det totala antalet



Antal frågor

[← Tillbaka](#)

Kurstestansvarig: *MPSN*

När du valt kurs och test är det dags att välja hur många frågor testet ska innehålla. Du har även här ett *menyfönster* där du kan välja från en fråga till antal frågor testet har. När du gjort ditt val trycker du på knappen *Antal frågor* för att gå vidare.

A.1.5 test.php



Kurstest
Karlstad Universitet

Kunskapstestet: Manual

Fråga 1 *Är detta en fråga till manualen?*

Svar a:	Ja.	<input checked="" type="radio"/>
Svar b:	Nej.	<input type="radio"/>
Svar c:	Aldrig.	<input type="radio"/>
Svar d:	Kanske.	<input type="radio"/>

Rätta testet ->

<- Tillbaka

<- Huvudmeny

Kurstestansvarig: MFSSN

Nu är alla val gjorda och det är dags för dig att skriva testet. Varje fråga har fyra svarsalternativ, välj ett svar genom att markera det svarsalternativ du tror är rätt (endast ett rätt svar/fråga). Upprepa detta tills du fyllt i svar för alla frågor, tryck sedan på knappen *Rätta testet* -> för att gå vidare och se ditt resultat.

A.1.6 ratta.php



Kurstest
Karlstad Universitet

Antal rätt på testet: Manual

Fråga 1 *Är detta en fråga till manualen?*

Svar a:	Ja.	<input checked="" type="radio"/>	Rätt
Svar b:	Nej.	<input type="radio"/>	
Svar c:	Aldrig.	<input type="radio"/>	
Svar d:	Kanske.	<input type="radio"/>	

OTROLIGT bra alla rätt.
Du fick 1 rätt av 1 frågor.

[← Huvudmeny](#)

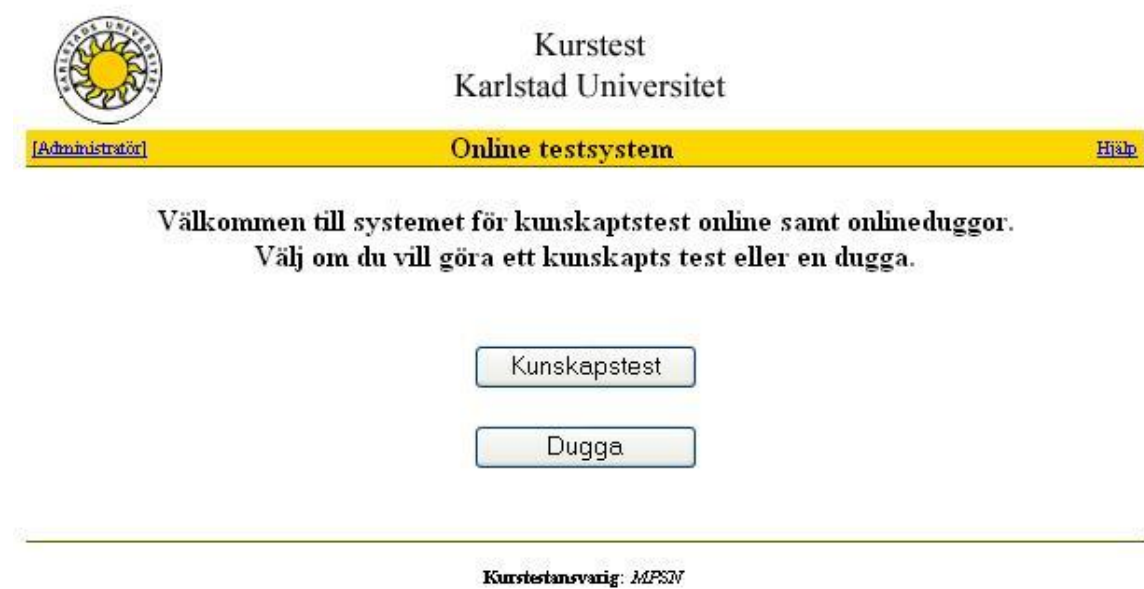
Kurstestansvarig: *MPSN*

När du kommer till denna sida är testet rättat. Alla frågor skrivs ut på sidan så att du kan se vilka svar du fyllt i. Det kommer även upp en *grön text Rätt* om du angett rätt svarsalternativ, eller en *röd text Fel* om du angett fel svarsalternativ.

A.2 Dugga

Dugga är den del där man måste vara registrerad för att kunna använda. En student har blivit tilldelad ett användarnamn och ett lösenord för en speciell dugga.

A.2.1 Index.htm



The screenshot shows the login page for the 'Kurstest Karlstad Universitet' online test system. At the top left is the university's logo, a circular emblem with a sun and the text 'KARLSTAD UNIVERSITET'. To the right of the logo, the text 'Kurstest Karlstad Universitet' is displayed. Below this is a yellow horizontal bar containing the text 'Online testsystem'. On the left side of this bar is a link labeled '[Administratör]' and on the right side is a link labeled 'Hjälp'. Below the yellow bar, the text reads: 'Välkommen till systemet för kunskapstest online samt onlineduggor. Välj om du vill göra ett kunskaps test eller en dugga.' There are two buttons: 'Kunskapstest' and 'Dugga'. At the bottom of the page, there is a horizontal line and the text 'Kurstestansvarig: MP52V'.

Denna sida är samma som den när man skriver ett kunskapstest. För att gå vidare och skriva en dugga så trycker du på knappen *Dugga*.

A.2.2 logginstudent.php



Kurstest
Karlstad Universitet

[\[Testsida\]](#) **Inloggning för studenter.** [Hjälp](#)

Använd ditt tilldelade användarnamn och lösenord för att skriva duggan.

Avtal för att göra tentamen vid KAU.

1. Det är ditt personliga användarnamn & lösenord.
2. Du gör tentamen endast med tillåtna hjälpmedel.

Lycka till!!!!

Jag accepterar avtalet: Nej, jag accepterar inte:

Användarnamn:

Lösenord:

[Glömt lösenord?](#)

Kurstestansvarig: MPSEN

När du kommit till denna sida är det dags att fylla i ditt *användarnamn* och ditt *lösenord*. Det är några saker du behöver tänka på, kontrollera så att du skriver i dina uppgifter korrekt annars kommer inget att hända, du måste även acceptera avtalet. När du gjort detta så trycker du på knappen *Logga in* för att komma vidare.

A.2.3 dugga.php



Kurstest
Karlstad Universitet

ManualD

Fråga 1 Är detta fråga 2?

Svar a:	Ja	<input type="radio"/>
Svar b:	Nej, fråga 1	<input type="radio"/>
Svar c:	Nej, fråga 3	<input checked="" type="radio"/>
Svar d:	<input type="radio"/>

Fråga 2 Är html ett språk?

Svar a:	Ja	<input type="radio"/>
Svar b:	Nej	<input type="radio"/>
Svar c:	Ett programmerings språk	<input checked="" type="radio"/>
Svar d:	<input type="radio"/>

När du trycker på rätta testet kommer duggan att rättas och lämnas in.
Det betyder att du inte längre kan ändra de svar du har fyllt i,
så kontrollera dina svar innan du lämnar in.

Rätta testet ->

Kurstestansvarig: MPEN

Nu är det bara att välja ett svarsalternativ till varje fråga. När du gjort det och är nöjd med dina svar trycker du på knappen *Rätta testet* -> för att gå vidare.

A.2.4 ratta.php



Kurstest
Karlstad Universitet

Antal rätt på testet: ManualD

Fråga 1 Är detta fråga 2?

Svar a:	Ja	<input type="radio"/>	Fel
Svar b:	Nej, fråga 1	<input type="radio"/>	
Svar c:	Nej, fråga 3	<input checked="" type="radio"/>	
Svar d:	<input type="radio"/>	

Fråga 2 Är html ett språk?

Svar a:	Ja	<input type="radio"/>	Rätt
Svar b:	Nej	<input type="radio"/>	
Svar c:	Ett programmerings språk	<input checked="" type="radio"/>	
Svar d:	<input type="radio"/>	

**GRATTIS du fick hälften eller fler rätt.
Du fick 1 rätt av 2 frågor.**

[← Huvudmeny](#)

Kurstestensvarig: *MPSN*

När du ser denna sida är dina svar sparade i databasen och du är klar med duggan. Du kan se till höger om varje fråga för att se om du svarat rätt på just den frågan. Under alla svar kommer en text upp med antal rätt du fick totalt. Det är nu bara att trycka på knappen *← Huvudmeny* för att avsluta och komma tillbaka till huvudmenyn.

A.3 Administratörer/Lärare

Administratörer har många olika saker som han/hon kan göra. Nedan i kapitlet finns det en fullständig redogörelse, för hur systemet kan användas.

A.3.1 index.htm



The screenshot shows the main interface of the 'Kurstest Karlstad Universitet' online test system. At the top left is the university's logo, a circular emblem with a sun and the text 'KARLSTAD UNIVERSITET'. To the right of the logo, the text 'Kurstest Karlstad Universitet' is displayed. Below this is a yellow horizontal bar containing the text 'Online testsystem'. On the left side of this bar is a link labeled '[Administratör]' and on the right side is a link labeled 'Hjälp'. Below the yellow bar, the main text reads: 'Välkommen till systemet för kunskapstest online samt onlineduggor. Välj om du vill göra ett kunskaps test eller en dugga.' There are two buttons: 'Kunskapstest' and 'Dugga'. At the bottom of the page, there is a horizontal line and the text 'Kurstestansvarig: MP52V'.

Startsidan är samma som för Kunskapstest och Dugga, men om du vill logga in i systemet som Administratör/Lärare får du använda länken som finns i rubriken. Tryck på länken [\[Administratör\]](#) för att komma till inloggningssidan.

A.3.2 login.php



Kurstest Karlstad Universitet

[\[Testsida\]](#)

Inloggning för kursansvarig och administratör

[Hjälp](#)

Logga in för att skapa ett kurstest

Användarnamn:

Lösenord:

[Glömt lösenord?](#)

Kurstestansvarig: *MPSON*

Den här sidan är till för administratörer/lärare, och systemansvarig (root) [1]. Som administratör/lärare använder du ditt användarnamn och lösenord som du fått tilldelat av den systemansvarige. Fyll i uppgifterna i fälten och tryck på knappen *Logga in* för att komma vidare.

A.3.3 usermeny.php



Kurstest
Karlstad Universitet

Meny för kursansvarig

[Hjälp](#)

Välkommen Mikael

Tryck här om du vill skapa test/dugga

Skapa test/dugga

Tryck här om du vill redigera ett befintligt test/dugga

Redigera

Tryck här se resultaten för en dugga.

Resultat

Tryck här om du vill avsluta och logga ut

Logga ut

Kurstestansvarig: *MPSEN*

Sidan som nu visas för dig är huvudmenyn, härifrån görs alla val och det är även hit du kommer när du är klar med ett val, och väljer att gå till huvudmenyn. Först kommer det visas vad som händer om du trycker på knappen *Skapa test/dugga*. Vad som händer om du trycker på de andra knapparna kommer senare i kapitlet.

A.3.4 rubrik.php



Kurstest
Karlstad Universitet

Rubrik på test

[Hjälp](#)

Välj vilken kurs du vill skapa test för
och sätt en rubrik

DSA Test: Dugga:

Rubrik: ManTest Skapa

Ta bort test: ManualID Ta bort

Visa alla test

[← Huvudmeny](#)

Kurstestansvarig: *MPSN*

I detta stadium kan du göra lite olika saker. Du kan ta bort ett befintligt test genom att välja namnet på ett befintligt test i menyn och trycka på knappen *Ta bort*. Du kan trycka på knappen *Visa alla test* och få upp en lista över alla befintliga test, dessa kommer att visas i en ny sida med namnet *listofuser.php* där en tabell med samtliga test/duggor i databasen visas. Om du vill skapa ett test, väljer du först vilken kurs du vill skapa ett test för i menyfönstret. Sedan väljer du om det ska vara ett test eller en dugga genom att göra ditt val. Slutligen fyller du i namnet på testet/duggan och trycker på knappen *Skapa* för att gå vidare i systemet.

A.3.5 skapa.php



Fråga 1:

Skriv in din fråga:

Hur många byte kan du ladda ner på 10 sekunder om du har en 10Mbits uppkoppling.

Skriv in dina svarsalternativ:

- A: 100 Mbyte
- B: 125 Mbyte
- C: 10 Mbyte
- D: 12,5 Mbyte

Rätt svar:



Kommentar till testpersonen:

Tänk på att du måste dela med 8.

Spara fråga

Förhandsgranska

← Tillbaka

← Huvudmeny

För att skapa en fråga till testet så fyller du i alla fälten (kommentar är inte obligatorisk). När du skrivit in din fråga behöver du inte använda ? eftersom systemet lägger till ? automatiskt. Fortsätt sedan med att fylla i svarsalternativen och markera vilket svarsalternativ som är det korrekta för frågan. Kommentar är ett frivilligt fält som kommer visas om man svarar fel på

frågan. När du fyllt i en fråga trycker du på knappen *Spara fråga*, så finns nu denna fråga på testet. Om du trycker på knappen *Förhandsgranska* kommer du till en sida ***preview.php*** som visar dina skapade frågor till testet. Om du trycker på knappen <- *Tillbaka* kommer du tillbaka till ***rubrik.php***. Om du istället väljer att trycka på knappen <- *Huvudmeny* kommer du tillbaka till ***usermeny.php*** som är din huvudmeny.

A.3.6 red_test.php

Om du skulle välja att trycka på knappen *Redigera* i huvudmenyn kommer du hit.



Kurstest
Karlstad Universitet

Redigera ett test [Hjälp](#)

Välj i vilken kurs du vill redigera ett test

DSA ▼ Redigera

← Huvudmeny

Kurstestansvarig: MPSSN

Här väljer du för vilken kurs du vill redigera ett test. Välj kurs i menyfönstret och tryck sedan på knappen *Redigera* för att komma vidare.

A.3.7 red_rubrik.php



Kurstest
Karlstad Universitet

Redigera ett test

[Hjälp](#)

Välj vilket test du vill redigera

ManualD

Redigera

<- Tillbaka

<- Huvudmeny

Kurstestansvarig: *MPSN*

Nu är det dags att välja det test du vill redigera. Välj testnamn i menyfönstret och tryck sedan på knappen *Redigera* för att komma vidare.

A.3.8 redigera.php



Redigera fråga: 1

Hur många Mbyte kan du ladda ner på 10 sekunder om du har en 10Mbits uppkoppling

Redigera dina svarsalternativ:

Svar:

A:	100	<input checked="" type="radio"/>
B:	125	<input type="radio"/>
C:	10	<input type="radio"/>
D:	12,5	<input type="radio"/>

Kommentar till testpersonen:

Tänk på att dela med

- Uppdatera
- Lägg till fråga
- Ta bort fråga
- Registrera student
- Förhandsgranska
- <- Tillbaka
- <- Huvudmeny

Kurstestansvarig: MPSSIV

På denna sida har du en mängd olika möjligheter. Välj redigera fråga, svar eller svarsalternativ och tryck sedan på knappen *Uppdatera* så ändras värdena i databasen. Tryck på knappen *Lägg till fråga* och du kan då lägga till en ny fråga till testet från sidan *skapa.php*. Knappen *Ta bort fråga* tar dig till sidan *delfraga.php* där du kan välja att ta bort en eller flera frågor från testet. Knappen *Förhandsgranska* tar dig till sidan *preview.php* som visar hur

testet ser ut för tillfället. Knappen <- *Tillbaka* tar dig ett steg bakåt i systemet och knappen <- *Huvudmeny* tar dig tillbaka till huvudmenyn. Om du trycker på knappen *Registrera student* så kommer du vidare till ***registrera.php*** se nästa sida.

A.3.9 registrera.php



Kurstest
Karlstad Universitet

Registrera studenter för ManualD

Förnamn:	Efternamn:	Personnummer (ÅÅMMDDXXXX):
<input type="text" value="Erik"/>	<input type="text" value="Viking"/>	<input type="text" value="1234567890"/>
<input type="button" value="Spara Student"/>		
<input type="button" value="Visa registrerade"/>		
<input type="button" value="← Till huvudmeny"/>		

Kurstestansvarig: *MPSN*

Här fyller du i studentens namn och personnummer. När du vill spara studenten tryck på knappen *Spara Student*. Efter detta kan du spara flera om du vill. I samband med att du sparar en student genereras ett användarnamn och ett lösenord som sparas tillsammans med de andra uppgifterna i databasen. Tryck på knappen *Visa registrerade* för att se alla studenter som finns registrerade för just denna dugga. Se nästa sida.

A.3.10 listastud.php



Kurstest
Karlstad Universitet

Resultat

Förnamn	Efternamn	Personnummer	Användarnamn	Lösenord
TestS	TestS	1234569870	0JdDL9b7	dr4984K4
Erik	Viking	1234567890	jQd9tdgb	0VXq98WI

Registera mera

← Till huvudmeny

Kurstestansvarig: *MPSN*

Så här ser de genererade användarnamnen och lösenorden ut för en registrerad student. Dessa uppgifter tillhör en viss dugga för att en student ska ha minimala rättigheter. I nästa avsnitt kommer huvudmenyns alternativ Resultat att visas.

A.3.11 show_result.php

När du väljer knappen *Resultat* i huvudmenyn, kommer du först till *student.php* där du väljer kurs. I nästa steg kommer du till *student1.php* där du väljer vilken dugga du vill se resultatet för. När du gjort dessa val kommer du till denna sida.



Kurstest
Karlstad Universitet

Resultat ManualID

Personnummer	Resultat
1234567890	1
1234569870	1

Kurstestansvarig: *MPSN*

B Enkät

Utvärdering av system för onlineduggor

Förklara gärna dina svar under kommentarer på sista sidan.

1. Generella frågor om att skriva duggor på nätet

i) Skulle du hellre skriva duggor på nätet hemifrån än med papper och penna i sal?

Ja

Nej

Vet ej

ii) Nämn de fördelar respektive nackdelar du kan komma på med att skriva duggor online.

Fördelar

Nackdelar

iii) Skulle du känna dig säker med att skriva duggor online, i fråga om sekretess? D.v.s. att ditt resultat och personuppgifter hålls hemliga.

Ja

Nej

Vet ej

Inget jag oroar mig för

iv) Tror du att mer fusk skulle förekomma om studenter ges möjligheten att skriva duggor hemifrån?

Ja

Nej, ingen skillnad

Vet ej

v) Har du några tidigare erfarenheter av att skriva tentamen eller duggor online?

Ja

Nej

2. Frågor om det system du har provat

i) Hur var ditt övergripande intryck av systemet?

Mycket bra

Bra

Godkänt

Dåligt

Mycket dåligt

ii) Vad tyckte du om användargränssnittet?

Mycket bra

Bra

Godkänt

Dåligt

Mycket dåligt

