

mEduWeb

ETT WEBBASERAT UNDERVISNINGSSYSTEM FÖR STUDENTER VID
ODONTOLOGEN, GÖTEBORGS UNIVERSITET

PROJEKTMEDLEMMAR

JENNY CLASSON
FRANS HALLBERG
PETER KRISTENSSON
JOHANNA LINDBERG
TOVE NYTH
JONAS PERSSON
FREDRIK PETTERSSON

HANDLEDARE

OLOF TORGERSSON, INSTITUTIONEN FÖR DATAVETENSKAP, CHALMERS



CHALMERS

GÖTEBORG, DEN 23 MAJ 2002

Innehållet i detta häfte är skyddat enligt Lagen om upphovsrätt, 1960:729, och får inte reproduceras eller spridas i någon form utan medgivande av författarna. Förbudet gäller hela verket såväl som delar av verket och inkluderar lagring i elektroniska och magnetiska media, visning på bildskärm samt bandupptagning.

© mEduWeb: Jenny Classon, Frans Hallberg, Johanna Lindberg, Peter Kristensson, Tove Nyth, Jonas Persson, Fredrik Pettersson

Sammanfattning

Datorer i undervisningen blir allt vanligare. När antalet datorer ökar både i skolorna och hemmen ges nya möjligheter för studenter att på egen hand söka information och kunskap via nya medier såsom internet.

Odontologen vid Göteborgs Universitet har under en tid haft problem med den traditionella undervisningen i Oral medicin. Kurslitteraturen är i vissa fall inaktuell och det är svårt för studenter att få tillgång till relevanta kliniska patientfall.

Projekt mEduWebs uppgift var att ge studenterna tillgång till den databas med patientundersökningar och bilder som under flera år har byggts upp på Odontologen. Informationen i databasen skulle dels presenteras på ett lättillgängligt sätt och dels anpassas till studenternas behov.

Det resulterande systemet är ett webbaserat undervisningsverktyg som låter studenter söka fritt i patientdatabasen och generera övningsuppgifter med patientbilder från verkliga fall. På så sätt kan de skaffa sig mer relevant och användbar kunskap. mEduWeb-systemet kommer att börja användas som ett komplement till den traditionella undervisningen på Odontologen under hösten 2002. Denna rapport beskriver utvecklingen av mEduWeb och innehåller också en beskrivning av det färdiga systemet.

Abstract

It is becoming more common to use computers in education. As the number of computers in schools and homes are increasing, new opportunities arise for students to search for information and knowledge through new media, such as the Internet.

The Faculty of Odontology at the University of Gothenburg have for a time been having problems with the education in Oral medicine. The course literature are in some cases outdated and it is difficult for students to gain access to relevant clinic cases.

Project mEduWeb's assignment was to give the students access to a database with patient examinations and pictures, that has been developed at the Faculty of Odontology. The information from the database was to be presented in an easily accessible way and adapted to the students' needs.

The resulting system is a web-based educational tool that allows students to search in the database and generate exercises with pictures of real patients. In that way they can gain more relevant and useful knowledge. The mEduWeb-system will be used as a complement to the traditional education at the Faculty of Odontology starting in the autumn of 2002. This report describes the development of the mEduWeb-system and also contains a description of the completed system.

Förord

Projekt mEduWeb är till ända. Efter ett år av planering, undersökningar, möten, diskussioner, projektmys, sena kodkvällar och allt annat som hör projektarbeten till har vi till slut ett system som både vi och beställaren kan vara nöjda med.

Vi i projektgruppen vill passa på att tacka de personer som varit till stor hjälp under vårt arbete. Personer som genom sitt stöd och engagemang gjort det möjligt för oss att skapa världens nyaste undervisningsverktyg för oral medicin.

Projekt mEduWeb vill börja med att tacka Olof Torgersson, handledare, bollplank och kritiker. Vi vill även tacka Mats Jontell, professor vid Odontologen i Göteborg, för värdefulla kommentarer och respons under utvecklingsarbetet.

Ett stort tack också till Fredrik Lindahl, examensarbetare vars programvara och kunskap avsevärt underlättat vårt arbete. Projekt mEduWeb tackar studenterna på Odontologen vid Göteborgs Universitet som ställt upp vid flera tillfällen och gett mycket värdefull feedback på vårt arbete.

Vi vill även passa på att rikta ett tack till de organisationer som har försett oss med den programvara som behövts för att göra projektet möjligt. Sun Microsystems för programspråket Java samt JavaServer Pages, Apache Software Foundation för Tomcat - deras implementation av en JSP-server, och PostgreSQL Inc., för databasen postgresQL.

Java, JavaServer Pages, JavaBeans och Servlets är registrerade varumärken och tillhör Sun Microsystems, Inc.

Innehållsförteckning

Sammanfattning.....	I
Abstract.....	II
Förord.....	III
1 Inledning.....	9
1.1 Bakgrund.....	9
1.2 Syfte och motivering.....	9
1.3 Projektgruppen.....	9
1.4 Krav.....	10
1.5 Målsättning.....	10
1.6 Läsanvisning.....	10
2. Analys.....	11
2.1 Användaranalys.....	11
2.1.1 Målgrupp.....	11
2.1.2 Omgivning.....	11
2.2 Uppgiftsanalys.....	12
2.2.1 Befintliga system.....	12
2.2.2 Möjliga förbättringar jämfört med befintliga system.....	12
3. Systemkonstruktion.....	13
3.1 Systemdesign.....	14
3.1.1 Databaserna.....	14
3.1.2 Datalagret.....	14
3.1.3 Datahanteraren.....	14
3.1.4 Dynamiskt webgränssnitt.....	14
3.1.5 Språkobjektet.....	15
3.2 Avgränsningar.....	15
3.3 Verktyg.....	15
3.3.1 Java.....	15
3.3.2 JavaBeans.....	16
3.3.3 JSP.....	16
3.3.4 SQL.....	16
3.3.4.1 PostgreSQL jämfört med MySQL.....	17
3.3.5 XML.....	17
4. Prototypen.....	18
4.1 Syfte.....	18
4.2 Verktyg.....	18
4.3 Testning.....	19
4.3.1 Målsättning.....	19
4.3.2 Testpersoner.....	19
4.3.3 Testmetod.....	19
4.3.4 Genomförande och omständigheter.....	19
4.4 Slutsatser.....	20
5. Enhetskonstruktion och implementering.....	21
5.1 Databasen.....	21
5.2 Datalagret.....	21
5.3 Datahanteraren.....	22
5.4 Språkobjektet.....	23

5.5 JSP-sidorna.....	23
5.5.1 Studenternas websidor.....	23
Sökverktyg.....	24
Övningar.....	24
Diagram.....	24
5.5.2 Administratörens websidor.....	25
Användarhantering.....	26
Statistik.....	26
Övningar.....	26
Kommentarhantering.....	26
Frågehantering.....	27
Kursinformation.....	27
6. Testning och utvärdering.....	28
7. Presentation av mEduWeb.....	30
7.1 Studenternas websidor.....	30
7.1.1 Startside.....	30
7.1.2 Sökverktyg.....	31
7.1.3 Övningar.....	34
7.1.4 Diagram.....	36
7.1.5 Sparat.....	38
7.1.6 Frågor.....	39
7.1.7 Kursinformation.....	39
7.1.8 Inställningar.....	40
7.2 Administratörens websidor.....	41
7.2.1 Startside.....	41
7.2.2 Övningar.....	42
7.2.3 Kommentarer.....	44
7.2.4 Kursinformation.....	45
7.2.5 Frågor.....	46
7.2.6 Användare.....	47
7.2.7 Inställningar.....	48
8. Vidareutveckling.....	50
8.1 Möjliga förbättringar av befintliga funktioner.....	50
8.2 Förslag på utökningar.....	50
9. Slutsatser.....	52
10. Referenser.....	53
Appendix A - Projektplan.....	54
Appendix B - Kravspecifikation.....	55
Appendix C - Studentintervju.....	56
Appendix D - Systemskiss.....	57
Appendix E - Funktionsbeskrivning.....	58
Appendix F - Prototyputvärdering.....	59
Appendix G - Enhetskonstruktion.....	60
Appendix H - Testspecifikation.....	61
Appendix I - Studentutvärdering.....	62
Appendix J - Flödesschema för websidorna.....	63

1. Inledning

Andelen datorer i både skolmiljö och i människors hem har ökat markant de senaste åren. Detta skapar mängder av nya möjligheter för alla tänkbara branscher, även för undervisning. Idag jobbar många studenter självständigt framför sina datorer på tider som de själva väljer. På Odontologen vid Göteborgs universitet bedrivs tandläkarverksamhet parallellt med undervisning av tandläkarstudenter. Men trots att en strid ström av patienter dagligen undersöks på Odontologen har studenterna där inte någon tillgång till dessa patientfall. Dessa fakta låg till grund då MedView, ett samarbetsprojekt mellan Odontologiska fakulteten och institutionen för Datavetenskap på Chalmers, startade projekt mEduWeb våren 2001. mEduWeb skulle vara ett dynamiskt undervisningssystem riktat till studenter i kursen Orala slemhinneförändringar och vara tillgängligt via internet.

1.1 Bakgrund

Våren 2001 startade mEduWeb som ett nytt projekt inom ramen för MedView. MedView är ett samarbetsprojekt som startades 1995 av anställda vid avdelningen för Oral medicin på Odontologen vid Göteborgs Universitet och från sektionen för Datavetenskap på Chalmers. Genom att utnyttja datorer, inte enbart för att lagra information utan främst för att på ett dynamiskt sätt bearbeta och presentera information, syftar MedView till att utveckla nya metoder och verktyg som kan underlätta det diagnostiska arbetet på tandläkarkliniker. Dessa metoder och verktyg kan också underlätta den studentundervisning som också sker på Odontologen. För en mer utförlig information om MedView, se referens 1. mEduWeb skapades med syfte att underlätta och förbättra undervisningen och ska ses som ett komplement till dagens undervisning. Eftersom mEduWeb är ett projekt sprunget ur MedView har en del av de verktyg som tidigare utvecklats inom MedView varit möjliga att återanvända efter vissa modifikationer. Det är främst verktyg och ideer för patientpresentation som har anpassats till mEduWebs format och således återanvänts.

1.2 Syfte och motivering

Avsikten med projekt mEduWeb var att skapa ett dynamiskt undervisningssystem för tandläkarstudenter. Systemet skulle vara lösningen på det problem som upplevts med att nå ut med relevant information och faktiska patientfall till studenterna på Odontologen. Systemet mEduWeb skulle ses som ett komplement till de ordinära studierna och samtidigt vara så pass lättanvänt så att alla studenter, även de med begränsad datorvana, skulle vara kapabla att använda sig av det. Genom att få tillgång till en stor mängd patientdata från varierande fall ur MedViews databas skulle studenterna genom mEduWeb kunna bredda sin kunskap. Att mEduWeb skulle vara webbaserat skulle göra det möjligt för studenterna att använda sig av systemet oavsett om de är i skolan eller hemma. Genom att skapa egna övningar skulle studenter på bästa sätt kunna använda mEduWeb till att testa och utveckla sina kunskaper.

1.3 Projektgruppen

Projekt mEduWeb består av sju studenter från Datateknik på Chalmers Tekniska Högskola. Projektet drivs som ett D3-projekt, vilket innebär att det är en del av den

obligatoriska undervisningen på Datatekniks tredje utbildningsår. Projektet har bedrivits parallellt med de vanliga studierna och det har också givit projektmedlemmarna möjlighet att läsa kurser som behandlar tekniker som sedan kunnat användas inom projektet. Ett stort mervärde med D3-projekten är att studenter från Datateknik ska lära sig att arbeta i grupp och lära sig hur ett professionellt projekt sköts. Att mEduWeb har varit ett D3-projekt har alltså inneburit att mEduWeb drivits som ett professionellt projekt och därmed blivit rikligt dokumenterat. Tre redovisningar har under året hållits för övriga datateknologer, handledare och övriga intresserade och de har också varit en del av de obligatoriska momenten inom D3-projektets ramar.

1.4 Krav

De ursprungliga krav som beställaren hade på mEduWeb-systemet var att patientinformationen skulle göras sökbar och att det skulle vara lättanvänt. Dessutom skulle systemet vara språkoberoende, dvs. alla sidor skulle lätt kunna göras tillgängliga på alla språk. Vidare diskuterades ytterligare krav, främst internt inom projektgruppen, men även i samråd med de blivande användarna och beställaren. Resultatet blev att nya krav som att användare skulle kunna generera övningar och ha en personlig sida bestämdes. Det skulle också vara möjligt att visualisera samband mellan data ur MedView-databasen, samt kunna ställa frågor till kursansvarig. Att systemet skulle vara lättanvänt innebar både att det skulle ha pedagogiska och tydliga gränssnitt mot användarna samtidigt som användarnas varierande vana av datorer inte fick vara ett bekymmer. Det tillkom efterhand även krav på säkerhet och snabbhet inom systemet. Kraven finns listade i Kravspecifikationen, se Appendix B.

1.5 Målsättning

Inom projektgruppen var det huvudsakliga målet hela tiden att inte lämna ifrån sig något vi inte var nöjda med. För att projektgruppen skulle kunna känna sig nöjd, gällde först och främst att uppfylla de mål som hade satts upp i Kravspecifikationen. Att systemet skulle komma att användas till det syfte det var ämnat för ansågs också viktigt. Det skulle inte ersätta traditionell undervisning utan snarare ses som ett komplement, men också vara så pass lättanvänt att det skulle kunna tas i bruk direkt. Slutligen var det även en målsättning att systemet skulle vara utvecklingsbart i framtiden.

1.6 Läsanvisning

De olika kapitlen i denna rapport beskriver utvecklingen av systemet mEduWeb. Kapitel 7, Presentation av mEduWeb, beskriver det färdiga systemet ur en användares perspektiv och kan läsas oberoende av övriga kapitel. Kapitel 5, Enhetskonstruktion och implementering, ger en mer teknisk beskrivning av systemet.

2. Analys

2.1 Användaranalys

För att kunna designa ett system som verkligen kommer att användas är det viktigt att noggrannt analysera de tänkta användarna och deras arbetssituation. mEduWeb's tänkta användare är studenter på Odontologen vid Göteborgs Universitet och i viss mån även den kursansvariga som administrerar systemet. Användaranalysen är uppdelad i två större delar, dels en analys av målgruppen, dvs studenterna och administratören, och dels en analys av den omgivning systemet kommer att användas i, dvs de fysiska förutsättningarna.

2.1.1 Målgrupp

Den primära målgruppen är alltså tandläkarstudenter på Odontologen i Göteborg. Målgruppsanalysen syftar till att klargöra deras förutsättningar för att använda ett webbaserat undervisningssystem som mEduWeb. Det som är mest intressant är studenternas datorvana, deras erfarenhet av internet, och deras allmänna inställning till den här formen av undervisning.

För att skapa en uppfattning om ovanstående frågor ordnades en intervju med nio tandläkarstudenter där de i grupper om tre fick prata fritt om undervisningssystem på internet och berätta om sin erfarenhet av datorer. Studenterna fick också ta ställning till ett antal tänkta funktioner i mEduWeb och bedöma om de tyckte de var viktiga att ha med eller inte. Intervjun visade att studenterna trodde sig ha tillräckligt med datorvana för att kunna hantera ett webbaserat system, men visade viss skepsis angående personlig inloggning och möjligheter att tentera via systemet. Denna skepsis grundade sig till viss del i en rädsla för övervakning, då studenterna var oroliga för att deras betyg skulle kunna påverkas av hur de använde systemet. Överlag var de dock positiva och hade många förslag på hur systemet skulle kunna se ut och användas. Studenterna såg internet som en bra källa för information och ville till exempel att en diagnostisk sökmotor skulle integreras i det färdiga systemet. En sammanfattning av studentintervjun finns i Appendix C.

Kontakten med administratören har varit kontinuerlig under hela utvecklingsarbetet. Under analysfasen ordnades också en intervju där administratörens förväntningar, förutsättningar och arbetsmiljö undersöktes.

2.1.2 Omgivning

Nästan lika viktigt som målgruppens vilja att använda systemet är deras fysiska möjligheter att göra detsamma. Ett webbaserat undervisningssystem som mEduWeb kräver tillgång till datorer och till internet. Upplevelsen och nyttan av systemet kan påverkas kraftigt av prestandan på användarens dator, skärmstorlek och framför allt uppkopplingshastighet mot internet. Under intervjun med studenterna undersöktes därför även hur tillgången till datorer på Odontologen ser ut och en datorsal inspekterades av medlemmar ur projekt mEduWeb. Det visade sig att det fanns flera datorsalar tillgängliga för tandläkarstudenterna, men att datorerna var av något äldre modell och att underhållet av datorsalarna var kraftigt eftersatt, många datorer fungerade inte alls. I den datorsal som inspekterades fanns en blandning av PC- och

Macintosh-datorer, samtliga var av äldre modell och skärmstorleken var typiskt 15 tum. Uppkopplingshastigheten var fullt tillräcklig. Den bristande tillgången på bra och fungerande datorer är något bekymrande, men förhoppningen är att många studenter väljer att använda systemet hemifrån där förutsättningarna troligtvis är bättre.

2.2 Uppgiftsanalys

Projekt mEduWeb hade redan från början relativt fria händer att skapa ett undervisningssystem kopplat till kursen i orala slemhinneförändringar och anpassat till tandläkarstudenter. Beställaren hade en önskan om att göra informationen i MedView-databasen tillgänglig för studenterna, men hur materialet skulle presenteras lämnades öppet. En viktig del i utvecklingen av mEduWeb har därför varit att undersöka vilka önskemål studenterna har om utformandet av systemet, vilken sorts information de önskar och hur informationen ska presenteras för att väcka intresse och locka till återkommande användande. Studenternas åsikter i dessa frågor kom fram under den studentintervju som gjordes för att analysera målgruppen. Det var också mycket viktigt att förstå vilket behov systemet ska fylla, hur dagens undervisning ser ut och om mEduWeb kan tänkas ersätta någon del av den befintliga undervisningen. Uppgiftsanalysen delades in i två delar, befintliga system och möjliga förbättringar jämfört med befintliga system.

2.2.1 Befintliga system

Idag utgörs undervisningen i kursen Orala slemhinneförändringar främst av traditionella föreläsningar. Som kompletterande kursmaterial har en CD-skiva med patientfall och bilder använts. CD-skivans huvudsyfte är att ge studenterna ökad tillgång till kliniska data och därigenom hjälpa dem att lättare känna igen olika symtom och ställa korrekta diagnoser. mEduWeb är tänkt att ersätta denna CD-skiva helt och hållet.

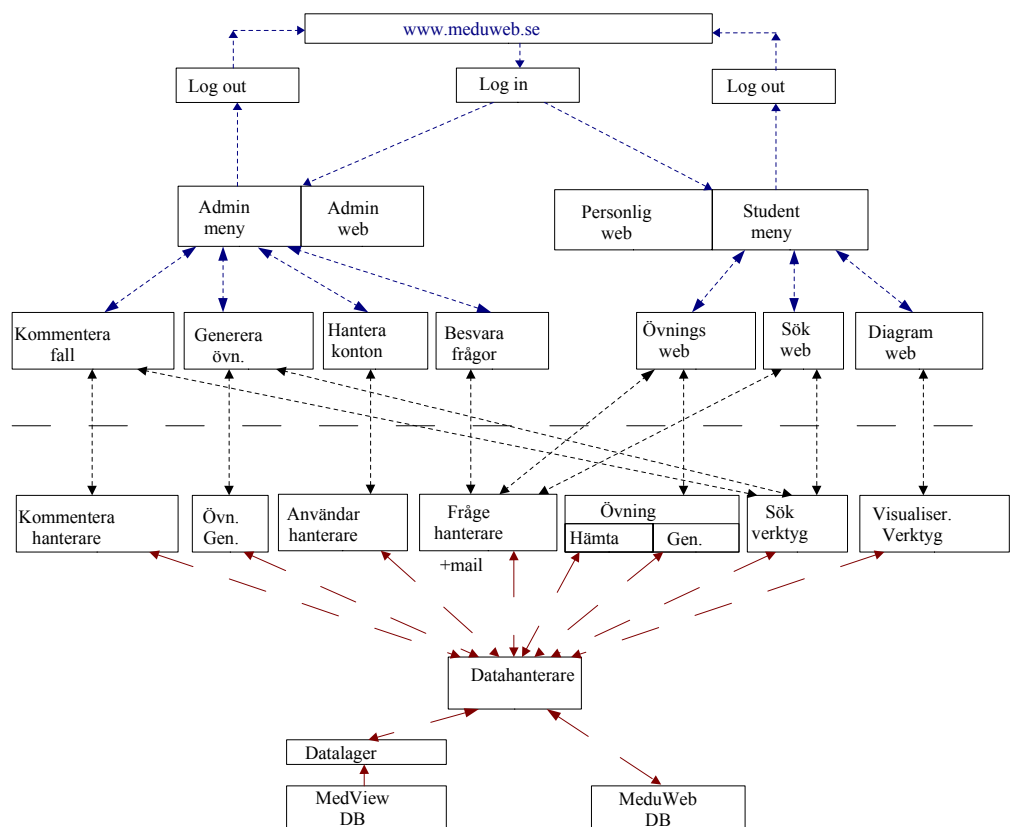
På internet finns ett flertal olika webbsiter som tillhandahåller patientbilder sorterade efter diagnoser eller tester genererade från bild-databaser, se referens 2, 3 och 4. Några av dessa är mycket omfattande och välgjorda, men överlag är de inte anpassade för studenter och är i vissa fall lite för omfattande och oöverskådliga eftersom de behandlar fler områden än just orala slemhinneförändringar.

2.2.2 Möjliga förbättringar jämfört med befintliga system

mEduWeb är tänkt att vara ett komplement till den traditionella salsbunda undervisningen och en ersättning till CD-ROM skivan som tidigare använts. Systemets främsta styrka gentemot CD-ROM skivan är att det är dynamiskt och tillhandahåller aktuell patientdata med information om patienten såväl som bilder från undersökningar. Dessutom ger mEduWeb möjlighet att skapa kursrelaterade övningsuppgifter och blir en informationskanal mellan de kursansvariga och studenterna. Fördelen med mEduWeb gentemot de befintliga system som idag finns på internet är att mEduWeb är anpassat till en speciell kurs och en speciell målgrupp. Den kursansvarige kan själv utforma övningar som passar för kursen och studenterna behöver inte leta och sälla bland överflödiga information.

3. Systemkonstruktion

Utifrån kravspecifikationen och analysen konstruerades ett webgränssnitt till MedView-databasen. För att underlätta konstruktionen delades systemet först upp i tre deluppgifter, hämta data från MedView-databasen, anpassa data och presentera data. En systemskiss upprättades för att sammanfatta systemets uppbyggnad, se bild nedan.



3.1 Systemdesign

De olika deluppgifterna kunde delas upp i mindre enheter för enhetskonstruktion och implementering, se Appendix E. För att hämta data från MedView-databasen konstruerades ett datalager, för anpassning av data designades en databas och en datahanterare och för att presentera data behövdes ett antal dynamiska websidor och ett språkobjekt.

3.1.1 Databaserna

Systemet använder sig av två databaser, dels den befintliga MedView-databasen där alla kliniska fall ligger lagrade, dels den så kallade mEduWeb-databasen. mEduWeb-databasen behövs för att spara all systemdata som inte hämtas från MedView-databasen och för att kunna anpassa data från MedViews databas till mEduWeb utan att redigera i den förra.

3.1.2 Datalagret

Hämtning av data från MedView sköts i första hand av datalagret. Det skapar ett hanterbart gränssnitt mot MedView-databasen och använder sig till stor del av tidigare utvecklade program inom MedView-projektet.

3.1.3 Datahanteraren

Datahanteraren skapar ett gemensamt gränssnitt för de båda databaserna och förmedlar all datatrafik mellan dem och mEduWeb. Genom att kombinera data från MedView-databasen och mEduWeb-databasen anpassas all information till systemet. Datahanteraren plockar alltså fall från MedView-databasen och kompletterar med data från mEduWeb-databasen innan informationen presenteras på websidorna.

3.1.4 Dynamiskt webgränssnitt

Det dynamiska webgränssnitt som ska presentera all data delades även det upp i mindre enheter. Följande delar kunde urskiljas:

- kommentarhantering
- övningsgenerering
- användarhantering
- frågehantering
- student-övningshämtare
- student-övningsgenerering
- sökverktyg
- visualiseringsverktyg

I kommentarhanteringen sköts de ändringar och tillägg som administratören vill göra till ett fall, hämtat från MedViews databas, innan det presenteras för användarna. Med hjälp av övningsgeneratoren kan administratören skapa övningsuppgifter till användarna. Användarhanteringen är den enhet som handhar all information om användarna av systemet, såsom lösenord, rättigheter och statistik. Frågehanteringen möjliggör att användarna kan ställa frågor till administratören, på både fall, visualiseringar och övningar. Till frågehanteringen finns det även en mailfunktion kopplad som ska skicka ett mail till administratören när en ny fråga har kommit in från en användare. Student-övningshämtaren hämtar övningar som administratören har skapat och student-övningsgeneratoren genererar övningsuppgifter automatiskt efter användarens önskemål. Sökverktyget kan användas av både administratören och användarna för att söka upp kliniska fall från MedView-databasen, efter specifika sökattribut. Visualiseringsverktyget ska kunna presentera statistisk data från MedViews databas med hjälp av diagram. Inom MedView används programmet "The Cube", se referens 1, som visualiseringsverktyg, men då det är för komplicerat att anpassa till internet, kommer ett annat enklare verktyg att användas tills vidare.

3.1.5 Språkobjektet

För att uppfylla kravet om språkberoende behövdes ett språkobjekt, det vill säga en typ av databas där all text, som visas på websidorna, ska finnas lagrad på de önskade språk som systemet ska stödja. Metoder för att hämta språkberoende texter ur språkobjektet ska då användas på websidorna istället för själva texten.

3.2 Avgränsningar

Under systemkonstruktionsfasen gjordes några avgränsningar i systemets funktionalitet. mEduWeb ska endast kunna visa tillgänglig information, det vill säga användare av mEduWeb ska inte kunna redigera information som lagts in i MedView-databasen och inte heller kunna lägga till data. Möjligheten att tentera via systemet valdes bort efter intervju med studenterna, men eftersom beställaren ställer sig positiv till denna möjlighet kommer den tas upp vidare under kapitlet om vidareutveckling. Funktionen student-övningsgenerering, som automatiskt skapar övningsuppgifter till användaren, begränsades till två olika typer av övningar, en stegvis övning och en diagnos-bild-övning. Övningstyperna konstruerades efter användarnas önskemål, som kom fram under studentintervjuerna.

3.3 Verktyg

Här nedan följer en kort beskrivning av de verktyg som valdes för att bygga systemet och en motivering till varför de valdes.

3.3.1 Java

Java är ett modernt programmeringsspråk som utvecklats av Sun Microsystems, se referens 5, vars strategi är att hålla Java-specifikationerna öppna. Detta gör att det finns många Java-bibliotek att tillgå och mycket arbete besparas projektgruppen. Java karakteriseras främst av att det är plattformsoberoende, det vill säga Java-program kan användas på alla datortyper. Dessutom är det relativt enkelt att skriva bra program med Java och hanteringen av säkerhet och kommunikation över datornät är mycket bra. Dessa egenskaper i kombination med valet av dynamiskt webverktyg, se punkten 3.3.3 JSP, bidrog till valet av Java som huvudsakligt programspråk till datalagret och datahanteraren.

3.3.2 JavaBeans

JavaBeans, se referens 6, är en standard för Javakod. Standarden säger att Java-klasser som är JavaBeans-kompatibla skall göras enligt ett visst mönster, som garanterar att klasserna kan användas i andra områden än vad de var tänkta för från början. Bl.a. krävs av en JavaBean att den skall kunna omvandlas till en sträng, för att kunna sparas till disk. Ett antal andra krav gör också att JavaBeans kan användas som komponenter vid användning av utvecklingsverktyg för Java.

3.3.3 JSP

För att välja ett verktyg för skapandet av det dynamiska webgränssnittet skapades en självstudiekurs för att utvärdera tillgängliga tekniker för att bygga dynamiska

websidor. De tekniker som utvärderades var CGI, DHTML, Java Applets, ASP, ASP.NET, PHP och JSP. Teknikernas fördelar, nackdelar och relevans för projektet undersöktes med följande resultat:

- CGI faller på sin dåliga prestanda, samt en för hög komplexitet, se referens 7.
- ASP är för dyrt och ASP.NET kommer inte ut förrän tidigast 2002. ASP.Net kostar också det pengar, se referens 8.
- DHTML är otillräckligt, men skulle kunna användas i vissa layout-syften, se referens 9.
- Java Applets är intressant att använda till visualisering av till exempel grafer, se referens 10.
- PHP och JSP är båda högintressanta på grund av deras snabbhet och goda databaskopplingar. Begränsningar i tillgängligheten kan relativt lätt överbryggas. JSP har fördelen att vara utbrett och lättlärt och har dessutom en större användargrupp, se referens 11 och 12.

Tekniken som valdes var JSP, JavaServer Pages, som använder Java-kod för att generera det dynamiska innehållet. HTML-koden kombineras med Java-kod och när sidan anropas översätts JSP-filen till en Servlet, som är en Java-klass som kompileras och körs på servern. Den genererade HTML-koden skickas sedan till webläsaren. De största fördelarna med JSP är tillgången till alla Java klasser, utbudet av gratis webbservrar som kan användas på flera olika plattformar och att det är ett snabbt och effektivt språk vid användning av databaser. Apache Tomcat valdes till webserver, dels för att den är gratis och stöds av Apache, som är en etablerad mjukvarugrupp, dels för att den stödjer de nyaste versionerna av JSP och Servlets.

3.3.4 SQL

SQL, Structured Query Language, valdes som databasspråk för att det definierar metoder för att skapa och manipulera relationsdatabaser på alla stora plattformar. Det är enkelt, snabbt och det finns många olika varianter som är gratis att använda.

3.3.4.1 PostgreSQL jämfört med MySQL

MySQL, se referens 13, är en variant av SQL som är gratis och som användes i början av projektet. Efter ett tag upptäcktes brister som inte kunde accepteras. Bland annat var det önskvärt med funktionella beroenden som ser till att mEduWeb-databasen inte innehåller överflödiga data. Detta är viktigt för att bibehålla en hanterbar storlek och därmed upprätthålla effektiviteten. Tillgängliga versioner av MySQL stödjer inte ännu dessa typer av integritetskontroller och nya alternativ fick undersökas. PostgreSQL, se referens 14, valdes som ersättare då den uppfyller kraven som ställs på mEduWeb-databasen samtidigt som prestandan inte minskar nämnvärt i jämförelse med MySQL.

3.3.5 XML

XML, Extensible Markup Language, se referens 15, valdes som verktyg för att uppnå språkoberoende, som ska lagra alla språkberoende texter som ska visas på websidorna. XML är ett språk för att beskriva strukturen i ett dokument. XML lagrar

data i en textfil och är uppbyggd av ett antal element. En XML-fil måste bestå av rotelement och start- och sluttaggar, i övrigt finns det inga krav på utseendet. Fördelen med XML jämfört med t.ex. en databas av typen SQL, i det här fallet, är att all data skrivs in i en vanlig textfil. Detta innebär att ingen särskild programvara behövs för att kunna ändra och lägga till data i XML-filen. För att få tillgång till den data som finns lagrad i XML-filen finns en mängd olika verktyg redan tillgängliga. DOM, Document Object Model, är ett API som tillhandahåller ett standardiserat, plattformsoberoende gränssnitt mot XML-filen. Ett DOM objekt brukar beskrivas i formen av ett träd. När en XML-fil läses in byggs en trädstruktur upp, där rotelementet hamnar överst i strukturen och övriga element representeras som noder i trädet. Det finns ett antal metoder i DOM som sedan kan användas för att manipulera och hämta data ur trädet.

4. Prototypen

Efter systemkonstruktionen fanns det ett behov av att testa den design som tagits fram. Av denna anledning skapades en prototyp av mEduWeb som testades och utvärderades mot användarna.

mEduWeb prototypen visar hela kontaktytan mot administratören och studenterna, helt utan den bakomliggande funktionaliteten med databaskopplingar och JSP. Prototypen består av följande delar:

- Inloggningssida med två användarnamn - admin och student
- En administratörswebb med fem större delar:
 - Startside
 - Kommentera fall
 - Övningar
 - Användare
 - Frågor och Kursinfo
- En studentdel som består av:
 - Personlig startside,
 - Fallfinnaren
 - Visualiseringsverktyget
 - Övningar
 - Kursmaterial och länkar

Den data som finns inlagd i prototypen är två stycken patientfall, tre olika övningar, och två visualiseringar. Studenterna kan göra en enkel sökning, en avancerad sökning och två visualiseringar. Administratören har färdiga exempel på hur en övning skapas och ett fall kommenteras. Dessutom finns ett antal frågor och användargrupper inlagda.

4.1 Syfte

Syftet med prototypen var att förtydliga och konkretisera de ideer och tankar som framkommit under systemkonstruktionen. För att utvärdera vissa delar, som till exempel de förslag på utformningen av övningar som fanns, var prototypen helt nödvändig. Prototypen begränsades av praktiska skäl till att endast visa kontaktytan för systemet, dvs utseendet på websidorna. Prototypen gav också beställaren en föraning om hur systemet skulle komma att se ut och en möjlighet till värdefull feedback från studenterna på Odontologen.

4.2 Verktyg

För att prototypen skulle likna det färdiga systemet så mycket som möjligt skrevs den i HTML-kod. Det gav också möjligheten att lägga upp prototypen på internet och

tillät studenterna att börja bekanta sig med ett undervisningssystem på nätet. JSP användes för att implementera inloggningen, och Javascript används för att leda användarna rätt väg genom sökverktyget, i övrigt är alla sidor statiska. Prototypen gjordes tillgänglig genom den webserver på Datavetenskap på Chalmers som satts upp för projekt mEduWebs räkning.

4.3 Testning

4.3.1 Målsättning

Det var tre olika aspekter som skulle testas:

- Interaktionen -sökningen, generera övningar mm.
- Presentationen -startsidan, fallpresentationen mm.
- Innehållet -övningstyperna, möjligheten att hämta kursmaterial på nätet mm.

Interaktionen ansågs vara det viktigaste, prototypen gav en möjlighet att få studenternas datorvana bekräftad och visade hur pass bra de kunde navigera på websidan. Presentationen väcker givetvis reaktioner, och även om prototypen var långt ifrån grafiskt fulländad var studenternas åsikt om delarnas tydlighet och relativa placeringar intressant. Genom testningen söktes också respons på innehållet, och då speciellt på övningarna. Var övningarna designade på ett bra sätt? Visades relevant information om patienterna?

4.3.2 Testpersoner

Administratörsdelen av mEduWeb prototypen testades av beställaren. Studentdelen testades av fem tandläkarstudenter som alla tidigare läst kursen i orala slemhinneförändringar och varav tre tidigare varit med och tyckt till om mEduWeb i samband med användaranalysen.

4.3.3 Testmetod

Eftersom det främst var interaktionen som skulle testas sattes några uppgifter ihop till studenterna. Testpersonerna fick tillsammans med en projektmedlem sitta vid datorn och utföra vissa moment som att till exempel skapa en övning och söka upp en patient med diagnosen cancer. Studenterna ombads berätta hur de tänkte för att hitta rätt på sidan och säga till om det var något som kändes konstigt. Efter varje genomfört moment ställdes direkta frågor om identifierade problemområden.

Beställaren hade redan tidigare fått tillgång till prototypen och därför var samma testmetod med "tänka-högt protokoll" omöjlig för administratörsdelen. Istället fick han under överinseende av två projektmedlemmar gå från sida till sida och berätta vad han hade för tankar om just den sidan. Beställaren fick framföra sina åsikter även om studentdelen.

4.3.4 Genomförande och omständigheter

Testningen genomfördes vid två olika tillfällen. Testet med beställaren utfördes på hans eget kontor med två representanter från projektgruppen närvarande. Testet med

studenterna genomfördes i en datorsal på Odontologen med en projektmedlem för varje testperson. Testningen lades på en tid efter tandläkarstudenternas sista lektion och de fick sitta med prototypen så lång tid som behövdes. Under testet låste sig ett par datorer och studenterna tvingades flytta runt lite, men detta påverkade sannolikt inte testresultatet. Resultatet i form av studenternas svar och kommentarer finns i Appendix F

4.4 Slutsatser

Vilka resultat gav då prototyptestningen och hur tas kritik och förslag till förbättringar till vara? Vissa saker är lättare att anpassa systemet efter medan andra kanske är svårare att hitta en snabb lösning till, men ändå viktiga att tänka på i det fortsatta arbetet. Några konkreta förändringar listas här:

- Bilder i övningarna ska göras större, det är viktigt att de är tydliga om de ska vara till någon hjälp.
- Meddelanden och frågor ska kunna hantera inbakade länkar.
- Övningarna kommer att förändras så att svaret eller lösningen blir mer avskild från resterande information.
- Startsidan behöver förändras och göras mer överskådlig.
- Problemet med kategorisering av värden måste lösas. Denna funktion uppskattas mycket av studenterna.
- Informationstext och förklarande exempel är viktigt. Detta måste prioriteras.
- Sökningen kommer att få en undermeny: Enkel sökning, Avancerad sökning och Sökning på patientkod.
- Flödet i sökning och visualisering behöver beskrivas tydligare. Studenterna behöver få veta vad som kommer att hända när de matar in parametrar.
- Användningen av Javascript för formulärhantering måste vara konsekvent genom sökningen och visualiseringen.

Eftersom prototypen inte utvecklades vidare efter denna utvärdering syns dessa förändringar främst i enhetskonstruktionen och i det färdiga systemet. Utöver de ovan listade punkterna gav testerna en känsla för hur användare uppfattar och tar sig an systemet som inte låter sig sammanfattas i ord men som ändå var väldigt värdefull. Sammanfattningsvis fyllde prototypen sitt syfte väl och gav värdefull respons på systemkonstruktionens tankar och uppslag.

5.Enhetskonstruktion och implementering

Enhetskonstruktionen och implementeringen var två skilda faser i utvecklingen av mEduWeb. Men eftersom de båda kan delas upp och beskrivas utgående från de olika systemdelarna behandlas de här i samma kapitel. Enhetskonstruktionen av mEduWeb var ett mycket omfattande arbete där varje systemdel och till och med varje enskild websida utformades och kopplades till övriga delar. Målet var att alla de specifika tekniska lösningarna skulle utarbetas och alla samband tydligt klargöras redan före implementeringen. Ett konstruktionsdokument utarbetades och granskades noggrant innan implementeringen tog vid, se Appendix G. På så sätt kunde varje enskild systemdel sedan implementeras separat och arbetet under implementeringsfasen kunde ägnas huvudsakligen åt de rent programmeringstekniska svårigheterna.

5.1 Databasen

För att lagra all systemdata, dvs all information om användare, övningar, frågor och annat som inte hämtas från MedView-databasen, används en PostgreSQL-databas. Databasmotorn körs på samma server som resten av mEduWeb och följer med installationspaketet. Databasen innehåller 22 tabeller för användare, grupper och profiler, språk, kommentarer, övningar, frågor, meddelanden, länkar, kursmaterial, sparade fall och visualiseringar samt statistik för användare och system. En fullständig beskrivning av tabellstrukturen finns i Appendix G.

Mellan tabellerna, eller relationerna, finns ett stort antal referenser och främmande nycklar som ska säkerställa att ingen felaktig data kan matas in och att ingen överflödiga data som inte längre är aktuell ligger kvar i databasen och tar upp utrymme. Till exempel så får det inte finnas flera användare med samma inloggningsnamn och en användares alla sparade fall och frågor raderas från databasen i samma stund som användaren tas bort från systemet.

Databasen designades utifrån de krav på lagringsmöjligheter och funktionella beroenden som ställdes av övriga systemdelar under enhetskonstruktionen och den modifierades vid flera tillfällen under arbetets gång.

5.2 Datalagret

Datalagret skapades för att på ett smidigt sätt kunna ta del av den data som finns i MedViews ursprungliga patientdatabas. Då någon del av mEduWeb vill läsa från MedViews databas anropar de, via Datahanteraren (se 5.3), någon av de funktioner som finns i Datalagret.

Datalagret konstruerades och implementerades som en Java-klass med olika metoder för att läsa önskad data från MedView-databasen. Då denna databas består av textfiler i en katalogstruktur skulle vanlig sökning och läsning i dessa filer vara relativt långsam. För att öka snabbheten i systemet valdes en lösning där Datalagret vid systemets uppstart läser in samtliga filer till en så kallad hash-tabell, då uppslag i denna går betydligt fortare. Hash-tabellen uppdateras varje dygn.

Patientpresentation hanteras också av Datalagret. För att underlätta användes här stora delar av det arbete som tidigare gjorts inom MedView för just patientpresentation. Vad Datalagret gör är helt enkelt att anpassa tidigare programvara till mEduWebs

webbaserade format. För att metoderna i Datahanteraren skulle bli lättillgängliga under implementeringen av websidorna följdes JavaDoc-standarden för dokumentation och dokumentationen gjordes tillgänglig via projekt mEduWebs hemsida.

5.3 Datahanteraren

Datahanteraren är på flera sätt en mycket viktig del av mEduWeb-systemet. Dess uppgift är att vara ett gränssnitt mellan de två databaserna, mEduWebs egen och MedView-databasen, och websidorna. Datahanteraren förmedlar all trafik som gäller hämtning eller lagring av information.

Datahanteraren konstruerades och implementerades som ett Java-paket med separata Java-klasser för de olika uppgiftstyperna. De klasser som implementerades var följande:

- Administration: Innehåller metoder för användarhantering och administration av systemet.
- Case: Metoder för att hämta patientinformation från MedView-databasen.
- Category: Metoder för att administrera kategorisering av värden.
- Comment: Metoder för fallkommentarer.
- Datahandler: Metoder för att administrera kopplingarna till mEduWebs databas.
- Exercise: Metoder för att skapa och presentera kursövningar.
- Lang: Ger tillgång till språkobjektet beskrivet i 5.4.
- LangDoc: Läser in en XML-fil till språkobjektet.
- Links: Metoder för att skapa länkar till kursinformationssidan.
- Login: Metoder för inloggning i mEduWeb-systemet.
- Mailer: Metoder för att skicka epost.
- Material: Metoder för att spara material på kursinformationssidan.
- Messages: Metoder för att spara och hämta administratörens meddelanden.
- Questionhandler: Metoder för att hantera frågor till administratören.
- Statistic: Metoder för att föra statistik över användandet av mEduWeb.

Enhetskonstruktionen gjordes genom att konstruktörerna av de olika websidorna lämnade in en begäran om vilken data som behövde hämtas eller sparas i databaserna, varefter lämplig metod designades för att lösa uppgiften. I vissa fall när flera websidor behövde hämta likartad information kunde en mer generell metod konstrueras för att passa dem alla. En klass skapades särskilt för att administrera kopplingarna mot mEduWebs SQL-databas medan samtliga förfrågningar om data från MedView-databasen skickas vidare från klassen Case till Datalagret.

Även för Datahanteraren följdes JavaDoc-standarden för dokumentation.

5.4 Språkobjektet

Språkobjektet är namnet på de delar av mEduWeb-systemet som möjliggör ett språkoberoende, dvs möjligheten att göra websidorna tillgängliga på flera olika språk. Vid leveransen fanns språken svenska och engelska implementerade men med relativt liten arbetsinsats kan fler språk läggas till.

Språkobjektet är implementerat som en XML-fil och två Java-klasser i Datahanteraren. I XML-filen lagras alla de strängar som ska visas på websidorna i en struktur baserad på sidindelningen med de tillgängliga språken som innersta element. Den första Java-klassen, LangDoc, har som uppgift att vid start eller omstart av systemet läsa in XML-filen till ett object i Java, en så kallad Document Object Model eller DOM. DOM har en trädstruktur och det är den andra Java-klassens uppgift att ta emot förfrågningar om strängar från websidorna, leta fram dem ur trädet och returnera den funna strängen.

Alla de websidor som visar strängar importerar datahanterarpaketet och därmed också språk-klasserna och använder en metod därifrån för att hämta strängen på rätt språk.

5.5 JSP-sidorna

Liksom övriga delar av mEduWeb enhetskonstruerades också det webaserade gränssnittet, i form av JSP-sidor, noggrant innan implementeringen. För varje sida antecknades vad som skulle visas, vilka interaktioner med databaserna som skulle behövas och hur flödet mellan olika sidor skulle se ut. Som nämnts tidigare, under beskrivningen av Datahanteraren, skapades de olika metoderna i Datahanteraren utefter enhetskonstruktionen av JSP-sidorna. Eftersom ett av kraven på systemet var att det skulle vara tydligt och lättanvänt, då användarnas datavana är begränsad, var det viktigt att tänka på att göra JSP-sidorna så pedagogiska och enhetliga som möjligt. För att skapa enhetlighet konstruerades ett stylesheet som gör att bland annat färger, typsnitt och rubriker ser likadana ut på alla sidor. En tydlig meny implementerades och inkluderas överst på varje sida för att underlätta navigeringen inom systemet. Varje sida har dessutom blivit noga genomtänkt under enhetskonstruktionsfasen för att den ska få ett naturligt och logiskt flöde mellan knappar, text och bilder. Systemkonstruktionen av websidorna ledde till en uppdelning i en studentweb och en administratörsweb, där en del sidor är gemensamma medan andra är specifika för studenterna eller administratören. De gemensamma sidorna, såsom sökverktyget och diagrammen, konstruerades det givetvis bara en version av och beskrivningen av dessa sidor återfinns här under studenternas websidor.

5.5.1 Studenternas websidor

Studenterna har tillgång till flera sidor som på olika sätt presenterar data om patientfall från de båda databaserna. Enhetskonstruktionen och implementeringen av dessa, för systemet mest nödvändiga, sidor beskrivs mer i detalj här nedan. De sidor som stödjer systemet genom att bland annat presentera information från administratören och behandla den data som är individuell för studenten, beskrivs inte närmare under den här punkten. För en beskrivning av de sidorna, se kapitel 7.

Sökverktyg

Sökverktyget är gemensamt för både studenterna och administratören. Det finns tre olika sätt att söka på: enkel sökning, avancerad sökning och patientkodssökning. I den enkla sökningen kombineras alla parametrar som valts automatiskt med det booleska uttrycket 'AND'. I den avancerade sökningen får användaren själv gå in och redigera sökuttrycket med hjälp av booleska uttryck, som 'AND', 'OR' och 'NOT'. För att kunna använda patientkodssökningen krävs en patientkod. Alla typer av sökningar använder sig av metoder ur klasserna Case och Category i Datahanteraren. Sökuttrycket skapas genom att ett attribut, som finns i MedView-databasen, väljs ur en lista. Då ett attribut väljs används en metod i klassen Case för att hämta alla de värden som är associerade med attributet. Värdena visas i en annan lista och genom att kombinera olika attributvärden med varandra går det att skapa mer eller mindre komplicerade sökuttryck. Vissa attribut kan innehålla sökbara kategorier. För att omvandla en sådan kategori till sökbara värden används en metod ur klassen Category. Det färdiga sökuttrycket alternativt patientkoden skickas sedan till en klass i Datahanteraren vilken förmedlar data mellan databaserna och den sida där sökresultatet ska visas.

Övningar

Övningsverktyget är uppdelat i två delar, en övningshämtare, som hämtar kursövningar gjorda av administratören, och en övningsgenerator, som automatiskt genererar övningar till studenterna. De kursövningar som finns tillgängliga listas genom ett anrop till en metod ur klassen Exercise, som hämtar titeln och id-numret på de kursövningar som är markerade som aktuella i mEduWeb-databasen. När en kursövning väljs skickas övningens id-nummer till en metod, återigen i klassen Exercise, vilken hämtar all data om övningen. Datan skickas till en ny sida som presenterar den som en övning. Det finns två olika sorters genererade övningar som kallas stegvisa och diagnos-till-bild övningar. Den stegvisa övningen består av ett antal flikar under vilka vald patients anamnes, status, bilder, daganteckningar och diagnos ska visas. När en diagnoskategori väljs ur en kategorilista, som skapas med hjälp av en metod i klassen Category, genereras ett valt antal patientkoder ur vald kategori med en metod ur klassen Case. När användaren väljer en av dessa patientkoder skickas den, tillsammans med den mall och den översättare som ska användas, till en metod i klassen Case. Valet av mall och översättare görs beroende på vilken flik som har blivit vald. Har användaren t.ex. valt att titta på patientens anamnes, skickas namnet på den mall som genererar anamnesen. Metoden returnerar passande mall ifylld med den valda patientens data. En diagnos-till-bild övning går ut på att välja den bildgrupp av tre som passar ihop med den diagnos som valts. På samma sätt som vid skapandet av en stegvis övning väljer användaren först en diagnoskategori ur en lista. Istället för en lista med patientkoder genereras en lista med diagnoser, som ingår i den valda kategorin. När användaren väljer en diagnos ur listan skickas den till en metod i Case och bilder från en patient med denna diagnos returneras. Dessutom genereras slumpvis, inom vald kategori, två andra patientfall med tillhörande bilder. Endast ett av de tre bildgruppsalternativen motsvarar den valda diagnosen.

Diagram

För att kunna skapa diagram och visa i mEduWeb har KavaChart, se referens 16,

köpts in. KavaChart är ett Servlet-bibliotek som genererar länkade bilder utifrån värden som ges som inparametrar.

Diagramdelen består av fyra olika JSP-sidor, en JavaBean som håller information om den önskade visualiseringen samt Servlets och JavaBeans från KavaChart-biblioteket. Den första sidan i visualiseringsverktyget använder sig av funktioner i Datahanteraren för att hämta de attribut och tillhörande värden som för tillfället finns i MedView-databasen. Först listas alla attribut i två rullningslistor. När ett attribut väljs laddas sidan om, attributet sparas i en JavaBean och värden tillhörande det aktuella attributet listas i en ruta där de önskade värdena i diagrammet kan väljas. När användaren väljer att gå vidare till nästa steg sparas alla valda attribut och värden och användaren vidarebefordras till sidan där begränsningar väljs. Liksom på föregående sida listas här alla befintliga attribut i en meny. När ett attribut väljs, laddas sidan om och tillhörande värden listas med hjälp av funktioner i Dataanteraren. När användaren väljer att lägga till en begränsning sparas denna i samma JavaBean som tidigare.

På nästa sida hämtas all data från den JavaBean som använts. Här används sökverktygets funktioner för att skapa indata till de JavaBeans som genererar diagrammen. Sökning görs för varje värde på attribut ett i kombination med eventuella värden på attribut två. Antalet träffar för varje sökning sparas, liksom vilka värden som använts i sökningen och patientkoderna för de fall som matchar sökningen. Patientkoderna sparas även i samma JavaBean som tidigare information, för att senare användas för att lista fall.

Beroende på om ett eller två attribut används i visualiseringen initieras en JavaBean, från KavaChart, som skall generera lämplig typ av diagram (endast ett attribut ger ett pajdiagram, två attribut ger ett stapeldiagram) med hjälp av de ingående klasserna. I denna JavaBean sätts indata i form av den data som genererades från sökningarna. Länkar skapas och skickas med för att kunna mappa diagrammets sektorer/staplar mot motsvarande patientfall. Värdena till de valda attributen skickas med för att användas på diagrammets x-axel. En bild genereras utifrån den JavaBean som skapats. Bildens sektorer/staplar är länkade till en sida som listar de fall som de representerar. För att åstadkomma detta skickas ett ordningstal med som används för att plocka ut rätt lista av fall från vår informations-JavaBean. Fallen i denna lista är länkade till samma sida som används för att visa fall i till exempel sökverktyget.

På sidan som genererar och visar diagrammet finns även funktioner för att ställa frågor på och spara det aktuella diagrammet. Dessa funktioner använder sig av metoder i Datahanterarens klasser. Frågor sparas i mEduWeb-databasen samt skickas som email till kursadministratören, som också kan besvara frågan från systemet. Sparade diagram läggs i en speciell katalog och en URL sparas i mEduWeb-databasen. Båda funktionerna genomförs genom att sidan laddas om då användaren klickar på 'Skicka' respektive 'Spara'.

5.5.2 Administratörens websidor

Administratören använder sina sidor för att kommunicera med studenterna och för att administrera systemet och kursen. Förutom sökverktyget och diagrammen använder sig administratören av ytterligare några verktyg, nämligen kommentarhantering, användarhantering, övningsskapning och frågehantering.

Användarhantering

Användarhanteringen används av administratören för att hantera konton, profiler och rättigheter för studenterna. Varje student har ett användarnamn och tillhör en viss profil och varje profil är associerad med ett antal rättigheter, som bestämmer vilka åtgärder en användare får utföra i systemet. Administratören har alla rättigheter från början, sedan kan administratören skapa nya profiler, ta bort och ändra profiler, skapa användare, ta bort användare och byta profil för en användare. All data om en användare, en profil och en grupp lagras i mEduWeb-databasen och metoder ur klassen Administration sköter interaktionen mellan användarhanteringssidorna och databasen. För att skapa nya användare väljer administratören endast hur många nya användare som ska skapas, vilken profil de ska ha, det vill säga vilka rättigheter, och vilken grupp de ska tillhöra. Metoden som skapar nya användare genererar sedan valt antal slumpmässiga lösenord associerade med ett användarnamn och sparar dessa i mEduWeb-databasen. Om bara en användare ska genereras är användarnamnet detsamma som gruppnamnet, annars består det av gruppnamnet följt av ett löpnummer.

Statistik

I mEduWeb-databasen sparas information om användandet av systemet, som kan presenteras som statistik för administratören. Metoder i klassen Statistic används för att lagra tider och inloggningar i systemet. Statistiken kan visas för hela systemet, totalt eller genomsnittligt, och för en enskild grupp av användare. En kolumn i statistiktabellen i mEduWeb-databasen räknas upp varje gång en användare loggar in. Total tid från inloggning till utloggning registreras genom att en tidsstämpel sätts när användaren loggar in och tidsskillnaden mellan tidsstämpeln och tiden för utloggning lagras i databasen.

Övningar

Ett av verktygen som administratören kan använda sig av är övningsgeneratoren. Med hjälp av den kan kursövningar till studenterna skapas på ett enkelt sätt. Övningssidan består av tre fönster. I det vänstra finns ett formulär där administratören fyller i titel, övningstext, svar och så vidare. I det mellersta visas de bilder som ska associeras med övningen och i det högra är sökverktyget inkluderat. Om administratören vill kan sökverktyget användas till att söka upp ett eller flera fall som övningen kan baseras på. Exempelvis så kan texten från ett fall kopieras in i övningstextrutan och bilder från ett eller flera fall inkluderas i övningen. Precis som i kommentarhanteringen flyttas en bild till det mittersta fönstret genom att klicka på den. Det finns även möjlighet att redigera bilder med hjälp av ritverktyget. Datan i formuläret i det vänstra fönstret skickas till en metod i klassen Exercise. Metoden sparar all information om uppgiften i mEduWeb-databasen. Samtidigt som övningen sparas skapas också ett unikt övnings-id som lagras tillsammans med övningen i databasen. De olika svaren ihop med en motivering till varför svaret är rätt alternativt fel sparas i en separat tabell. Det är med hjälp av övnings-id:t som studenter senare kan hämta en kursövning och även registrera i en annan tabell att en övning har blivit gjord.

Kommentarhantering

Administratören har möjlighet att göra kommentarer i fall och editera bilder som visas för studenterna. Kommentarhanteringssidan består av tre fönster, ett där kommentaren skrivs in, ett där eventuella bilder kan redigeras och ett där sökverktyget inkluderas. Administratören använder sökverktyget för att söka upp ett fall som ska kommenteras. En metod i klassen Comments tar först reda på om det finns någon tidigare kommentar på fallet, finns det inte det så visas en tom textruta där administratören kan skriva in en kommentar. Om det finns en tidigare kommentar så skrivs den ut i textrutan och administratören kan då välja att ändra i den gamla kommentaren eller lägga till en ny under den gamla i textrutan. När administratören trycker på knappen för att spara kommentaren skickas patientkoden och texten som inparametrar till en metod som lagrar kommentaren i mEduWeb-databasen. Bilder som hör till det uppsökta fallet kan flyttas till det mellersta fönstret med ett musklick. Till bilderna finns ett ritverktyg kopplat med vilket administratören kan göra tillägg till bilden, såsom linjer och cirklar. Ritverktyget är en Java Applet som konstruerades för att passa till önskemålen för mEduWeb. När den redigerade bilden ska sparas används en metod som sparar bilden på servern och lagrar adressen till bilden, tillsammans med patientkoden, i mEduWeb-databasen. Kommentarer och redigerade bilder visas tillsammans med patientfallets text och oredigerade bilder. Metoden som ska presentera fallet tittar alltid först i mEduWeb-databasens kommentartabell om det finns några kommentarer och/eller redigerade bilder associerade med den aktuella patientkoden.

Frågehantering

Studenterna har möjlighet att ställa frågor till administratören. Det går att ställa frågor på fall, övningar och visualiseringar och det går även att ställa allmänna frågor. På varje sida som visar ett fall, en övning eller ett diagram finns det även en frågeruta där studenten kan skriva en fråga. Frågan och användarens id sparas tillsammans med patientkoden, övnings-id eller sökvägen till diagrambilden med hjälp av en metod i klassen Questionhandler. Dessutom används klassen Mailer för att skicka ett email till administratören om att en ny fråga har inkommit. På administratörens frågehanteringssida listas besvarade och obesvarade frågor separat av en metod i Questionhandler. Metoden tittar i den kolumn i frågetabellen där svaret lagras för att avgöra om frågan har blivit besvarad eller inte. För att besvara en fråga klickar administratören på frågan som är länkad till det fall, den övning eller det diagram som frågan är associerad till genom sitt id-nummer eller sökväg. På dessa sidor finns det en svarsuta där administratören kan skriva in ett svar på frågan och detta lagras av en metod i Questionhandler-klassen i frågetabellen i databasen.

Kursinformation

Administratören kan lägga upp material, såsom föreläsnings-OH och länkar och skicka meddelanden till systemets användare. Materialet läggs upp på servern med hjälp av metoder i klassen Material. För att ladda upp materialet från en lokal disk till servern används en färdig JavaBean som heter mySmartUpload, se referens 17. En beskrivning av materialet tillsammans med sökvägen till det lagras i en tabell i mEduWeb-databasen. På samma sätt lagras länkar som administratören lägger upp, men här används istället en metod i klassen Links. För att skicka ett meddelande

skriver administratören in det i en textruta. Meddelandetexten skickas till databasen med hjälp av en metod ur klassen Messages och lagras tillsammans med en tidsstämpel.

6. Testning och utvärdering

Inför implementeringsfasen utformades en testspecifikation som bl. a. behandlade olika testförfarande för att kontrollera att de krav som var ställda på systemet uppfylldes. Arbetet med testning har varit en kontinuerlig process och har skett parallellt med enhetskonstruktion och implementering. Samtidigt som projektgruppen utvärderat och testat systemet efterhand som det växt fram har värdefull hjälp fått från handledare och beställare. Tips och ideér från utomstående har också gett inspiration.

I projektets slutfas genomfördes, utgående från testspecifikationen, en sista enkätundersökning på studenter från Odontologen. Studenterna fick ett antal uppgifter att lösa och sedan möjligheten att på egen hand undersöka och finna brister i systemet. Sedan fick de svara på frågor om systemet, se appendix H. Ett misstag med denna utvärdering, som framgick i efterhand, var att studenterna som utförde testerna redan hade en relativt klar bild av vad de kunde göra med mEduWeb eftersom de var samma studenter som tidigare hade varit med och utvärderat prototypen. Trots detta framkom en del givande information.

Det var överlag väldigt positiv kritik och inga större fel upptäcktes. Samtliga testpersoner ansåg att de skulle använt systemet som studiekomplement om det funnits tillgängligt när de läste kursen. Det framkom också önskemål om att använda sig av systemet även som färdigutbildad tandläkare. Övrigt som framkom vid denna utvärdering kan sammanfattas som följer.

- Det var ingen av testpersonerna som hade några problem med att logga in och samtliga fann startsidan både överskådlig och lättnavigerad. Här är det viktigt att ha i åtanke att testpersonerna redan hade en bild av systemet sedan tidigare erfarenheter. Det framkom också önskemål om att få en genomgång av systemet och dess möjligheter innan det börjar användas som komplement till studierna.
- Trots att det i vissa fall saknades en del patientinformation som borde varit med, upplevde de flesta testpersonerna att den information som presenterades var bra och i de flesta fall fullt tillräcklig. Att det saknades information berodde på att det vid utvärderingstillfället enbart fanns en begränsad mängd patientinformation tillgänglig.
- Övningsdelen upplevdes också som positiv. Det var enkelt att skapa egna övningar, men vid en typ av övning krävs att kursadministratören i förväg skapat kategorier för olika diagnoser. Det fanns vid teststillfället fyra olika kategorier och här upplevde testpersonerna att de ville ha fler kategorier att testa sig på. Detta är något som kommer att föras vidare till den person som i framtiden ska administrera systemet.
- Visualiseringsverktyget fick blandad kritik. Vissa ansåg sig inte ha några problem att skapa diagram, medan andra inte lyckades alls. Samma sak gällde för den avancerade patientsökningen.
- De hjälptexter som skapats för att underlätta vid användningsproblem uppfattades som positiva. De som använde hjälptexterna fann de informativa och lättförståliga. Ett problem upptäcktes med att man inte kunde bläddra i det fönster där

hjälptexterna presenterades, vilket ledde till att all information inte kunde nå ut.

Resultatet av utvärderingen var i stort sett det förväntade. Det uppfattades som lättanvänt och lättöverskådligt även om diagramdelen och den avancerade sökningen kan förbättras ytterligare. Vidare utvärdering av systemet kommer att ske i samband med att det tas i bruk på Odontologen under hösten 2002.

7. Presentation av mEduWeb

Projekt mEduWebs arbete har resulterat i ett färdigt undervisningssystem som kommer att börja användas på Odontologen vid Göteborgs Universitet under höstterminen 2002. Användarna av mEduWeb är i första hand tandläkarstudenter, som ska använda systemet som ett komplement till sin undervisning, och i andra hand kursansvarig, som administrerar kursens och mEduWebs innehåll. mEduWeb består utåt sett av ett webbaserat gränssnitt som är uppdelat i två huvuddelar, studenternas websidor och administratörens websidor. För att lättare kunna förstå flödet mellan systemets olika sidor finns flödesscheman upprättade i Appendix J. Varje student har sin personliga sida som denne loggar in på med hjälp av ett användarnamn och lösenord. Likaså använder sig administratören av ett användarnamn och lösenord för att logga in på administratörssidan.



mEduWebs loginsida

7.1 Studenternas websidor

Efter att ha loggat in har studenten tillgång till ett flertal websidor som kan utnyttjas för att söka bland patientfall ur MedViews databas, göra övningar och kommunicera med kursadministratören. Överst på varje sida finns en meny med vilken det lätt går att navigera bland de olika sidorna. Nedan följer en presentation av studenternas sidor.

7.1.1 Start sida

Den första sidan som visas efter att en student har loggat in på mEduWeb är startsidan. Förutom en kort presentation av mEduWeb visas också aktuell information, både individuell och allmän, på sidan. Sidan är uppdelad i tabellform,

med tydliga rubriker, för att den ska vara lättöverskådlig. Det som visas på startsidan är Senaste meddelandet, Nya övningar, Senaste material, Sparade patientfall, Sparade diagram och eventuella besvarade frågor. Det finns dessutom möjlighet att från startsidan ställa en allmän fråga till kursadministratören. Under rubriken Senaste meddelandet visas den nyaste upplysningen som administratören har skickat ut till studenterna. Meddelandet är länkat och leder till kursinformationssidan, där fler meddelanden visas. Under Nya övningar listas alla de aktuella kursövningar som administratören har skapat och som studenten ännu inte har gjort. För att utföra en av dessa övningar klickar studenten på övnings titeln, som är länkad till övningsuppgiften. Det senaste upplagda materialet, såsom t.ex. Föreläsnings-OH, kommer studenten åt genom att klicka på länken under Senaste material. Mer eventuellt material ligger på kursinformationssidan som nås via menyn. Sparade patientfall och diagram fungerar som ett slags bokmärke till de fall och diagram som studenten själv har sparat undan. Genom att klicka på en titel kan studenten återgå till ett sparat patientfall utan att behöva söka upp det igen eller titta på ett diagram utan att återskapa det. Studenten kan ställa en fråga till administratören genom att skriva frågan i frågerutan och sedan trycka på 'Skicka'. Frågan skickas då till administratörens startsida och som ett email. När studentens fråga har blivit besvarad av administratören dyker den upp under Besvarade frågor. Här visas även de besvarade frågor som studenten har ställt på ett fall, en övning eller en visualisering. För att hålla startsidan så enkel och tydlig som möjligt visas inte de eventuella rubriker som inte har någon information under sig.



Studenternas startsida

7.1.2 Sökverktyg

På dessa sidor har studenterna och administratören möjlighet att söka upp patienter

enligt olika kriterier. Sökningen är uppdelad i tre steg, först väljs värden att söka på, sedan ges en lista med de patientfall som matchade sökningen, och sist visas en presentation av en patient. Steg ett finns i tre olika varianter; enkel sökning, avancerad sökning och patientkodssökning.

Enkel sökning: Här väljer användarna olika kombinationer av attribut och värden. Först väljs ett attribut i den övre vänstra rullningslistan, och sedan väljs ett värde för det attributet bland de som visas nedanför. Denna kombination av attribut och värde visas sedan i rutan till höger. Processen upprepas tills användaren valt ut de önskade kombinationerna. Om en kombination i den högra rutan markeras och knappen 'Ta bort' trycks ner så tas den kombinationen bort ur sökningen. Det finns även en knapp som rensar alla valda attribut och värden. Genom att klicka på 'Sök'-knappen kommer användaren vidare till resultatsidan.

Enkel sökning

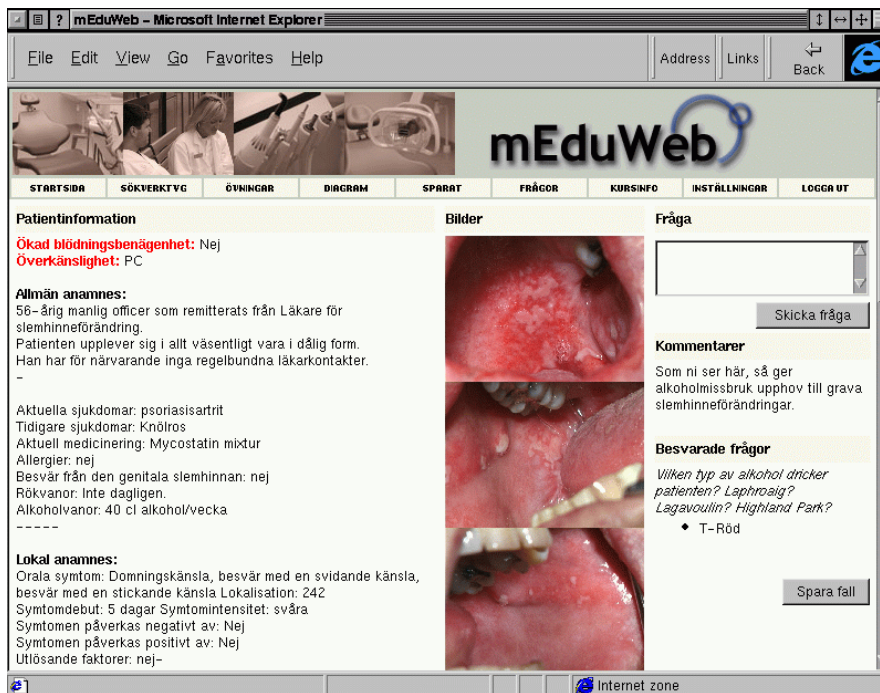
Avancerad sökning: Här kan användarna sätta ihop mer avancerade sökuttryck. De grundläggande stegen i sökningen är likadana som för enkel sökning; först väljs ett attribut och sedan väljs ett värde för det attributet. Efter att användaren valt ut de olika kombinationer av attribut och värden som önskas finns dessa listade i den högra rutan. Ur denna ruta komponeras sedan ett sökuttryck genom att de olika attributen, samt noll eller flera av de olika operatorerna som finns, flyttas till sökuttrycksrutan i önskad ordning. Önskas en sökning som ger t.ex. patienter som är ogifta, födda i Sverige samt är icke-rökare, skapas först dessa tre olika kombinationer av attribut och värden. Sedan markeras de kombinationer som valts och varvas med operatoren 'OCH'. Viktigt att notera är att det måste finnas en operator mellan varje kombination av attribut och värde. Det resulterande sökuttrycket visas i sökuttrycksrutan längst ner. Knappen 'Sök' leder vidare till resultatsidan.

Avancerad sökning

Patientkodssökning: Här finns möjligheten att söka på en patientkod, sidan innehåller endast ett inmatningsfält samt en knapp 'Sök'. Om en felaktig patientkod matas in visas ingen patient, istället visas ett felmeddelande. Matas en korrekt patientkod in visas denna patient direkt på en ny sida, utan resultatsida emellan, eftersom varje patientkod är unik.

Steg två är en resultatsida. Denna sida visar en lista med korta patientpresentationer. Längst upp på sidan visas hur många träffar sökningen genererade. Maximalt tio träffar i taget visas och möjlighet finns att stega fram och tillbaka för att visa nästa tio respektive föregående tio träffar. Exempel på en kort beskrivning kan vara: "45-årig kvinnlig cykelreparatör som remitterats för smärta. Patienten är av libanesiskt ursprung."

Steg tre är patientpresentationen. På denna sida visas information angående en patient. Den vänstra delen av sidan tas upp av patientinformation, som består av fakta kring patienten, orsak till besök, symtom, övriga sjukdomar som kan vara relaterade till besöket, samt ett antal olika egenskaper och värden hos patienten. I mitten av sidan visas de bilder som är tagna på patienten, samt eventuella vävnadsprov, så kallade histologiska snitt. Till höger visas två funktioner för studenterna; fråga och spara. Här kan studenter ställa frågor till kursadministratören angående det fall som visas just nu. När en fråga skrivs in och knappen 'Skicka fråga' trycks ner skickas frågan till kursadministratören. Svar på frågan kommer sedan visas på studentens startside och frågesida. Studenten kan också spara fallet för att kunna titta på det senare, detta görs med hjälp av knappen 'Spara'. När knappen trycks ner kommer det upp ett nytt fönster där en beskrivning på patienten matas in. Den sparade patienten finns sedan tillgänglig på studentens startside samt på studentens Sparat-sida.



Patientpresentation

7.1.3 Övningar

Att studenter har möjlighet att göra övningar är en viktig del av mEduWeb-systemet. Övningarna är indelade i tre delar: Kursövningar, Stegvisa övningar och Diagnos- till- bild övningar. Kursövningar skapas av kursadministratören, medan Stegvisa övningar och Diagnos-till-bild övningar skapas av studenterna. Genom att studenterna utför övningar som kursadministratören skapar, kan de träna sig på kursens centrala delar. De övningar studenter själva skapar ger färdighet i att ställa diagnos.

När Övningar väljs i menyn visas en sida där studenterna har möjlighet att välja vilken sorts övning de vill göra. Till vänster på sidan finns en kort beskrivning av syftet med de olika övningarna, samt en meny där övningstyp kan väljas. Sidans största del upptas av inledningssidan till den övningstyp som valts.

Kursövningarnas startsida består av en lista med de kursövningar som administratören skapat och gjort tillgängliga. Val av övning sker genom ett klick på länken med dess titel. Övningen kommer då att visas och kan sedan utföras. Läs mer om kursövningarnas utformning under 7.2.2.



Kursövningar

Stegvisa övningar ger studenterna träning i att ställa diagnos. I inledningsfasen till de stegvisa övningarna väljer studenterna en diagnoskategori att öva på och anger hur många övningar de vill skapa. Varje övning motsvarar ett slumpvis utvalt patientfall. När kategori och övningsantal har valts visas de utvalda patientfallen i en lista till vänster på sidan.



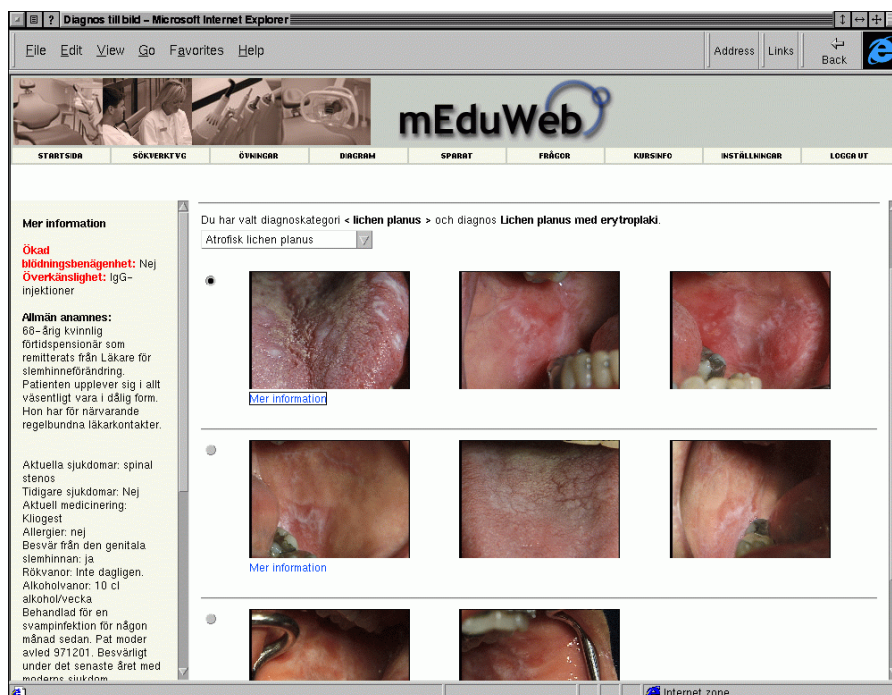
Stegvis övning

En övning väljs genom länken med motsvarande patientkod. Då kommer ett antal flikar att visas på sidans högra del. Det finns fyra informationsflikar, Anamnes, Status, Bilder och Daganteckning, samt en svarsflik, Diagnos. Tanken är att studenten ska klicka sig fram, en flik i taget, och därmed stegvis få tillgång till mer och mer information om patientens tillstånd. Slutligen ska studenten försöka bestämma den aktuella diagnosen innan han eller hon kontrollerar sitt svar under diagnosfliken.

För att välja en ny övning används de klickbara patientkoderna i listan. Det går också att komma tillbaka till inledningsfasen genom att klicka på länken 'Ny stegvis övning'.

Även diagnos-till-bild övningar låter studenterna öva på att ställa diagnos. Övningen går ut på att avgöra vilka bilder som hör till en viss diagnos. I inledningsfasen väljer studenten vilken diagnoskategori han eller hon vill öva på. Efter det så väljs vilken diagnos i diagnoskategorin som ska vara den korrekta. Tre rader med bilder kommer då att visas. Varje bildrad tillhör ett specifikt patientfall. En av patienterna har den korrekta diagnosen, de andra två har en diagnos som finns inom den valda diagnoskategorin. Genom att klicka på länken Mer information som finns under varje bildrad blir mer information om patienten tillgänglig. Informationen visas till vänster på sidan.

En patient väljs med radioknappen till vänster om patientens bildrad. För att svara klickar studenten på 'Svara'-knappen längst ner på sidan. Resultatet, rätt eller fel, kommer att visas till vänster på sidan.



Diagnos-till-bild övning

7.1.4 Diagram

Denna del av mEduWeb är till för att presentera data från MedView-databasen

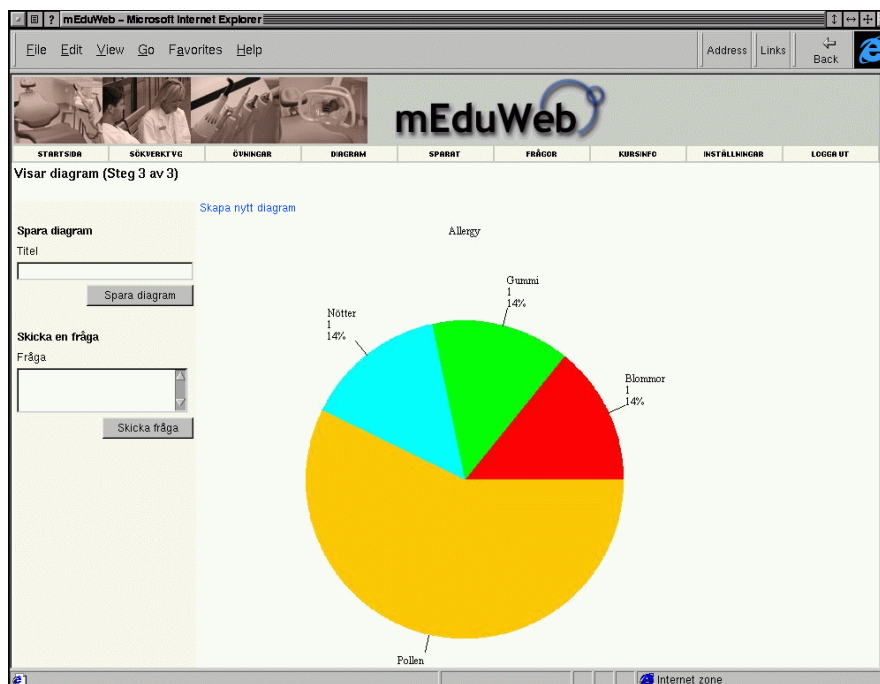
visuellt. Den ger möjlighet att skapa paj- och stapeldiagram för att åskådliggöra eventuella samband och kopplingar mellan de attribut som tas med i diagrammet. Till exempel så kan användaren välja att visualisera förhållandet mellan ålder och diagnos för att på så sätt undersöka om en viss sjukdom är åldersrelaterad eller inte.

Diagramdelen av mEduWeb är densamma för studenter som för administratörer och är indelad i tre steg som här beskrivs i tur och ordning.

Det första steget är inmatning av attribut och värden. Det innebär att användaren här väljer vilka data han eller hon vill ta med i sitt diagram. Det är möjligt att välja ett eller två attribut med tillhörande värden. När endast ett attribut väljs kommer resultatet visas i form av ett pajdiagram, om båda attributen väljs presenteras resultatet istället i ett stapeldiagram. Inmatningen går till ungefär som i sökningen, när ett attribut väljs ur listan visas alla möjliga värden för detta attribut i den nedre listan och ett eller flera värden kan markeras. Det speciella värdet 'ALLA' kan väljas för att låta attributets samtliga värden ingå i diagrammet. Ett exempel på en inmatning kan vara att det ena attributet väljs till 'Born' med värden 'Sverige' och 'England' och det andra till 'Diag-tent' med värdet 'Lichen Planus'. Det resulterande diagrammet skulle då avslöja eventuella skillnader i förekomsten av den tentativa diagnosen Lichen Planus mellan patienter födda i Sverige och patienter födda i England.

Diagram, steg 1

Begränsningar av vad som ska tas med i diagrammet kan göras i steg två. Om användaren till exempel bara är intresserad av att jämföra förekomsten av Lichen Planus bland svenska och engelska kvinnor kan han eller hon här välja att lägga till den begränsningen. För att begränsa patientunderlaget och diagrammet väljs önskad kombination av attribut och värde ur listorna. Det är möjligt att göra flera begränsningar om så önskas.



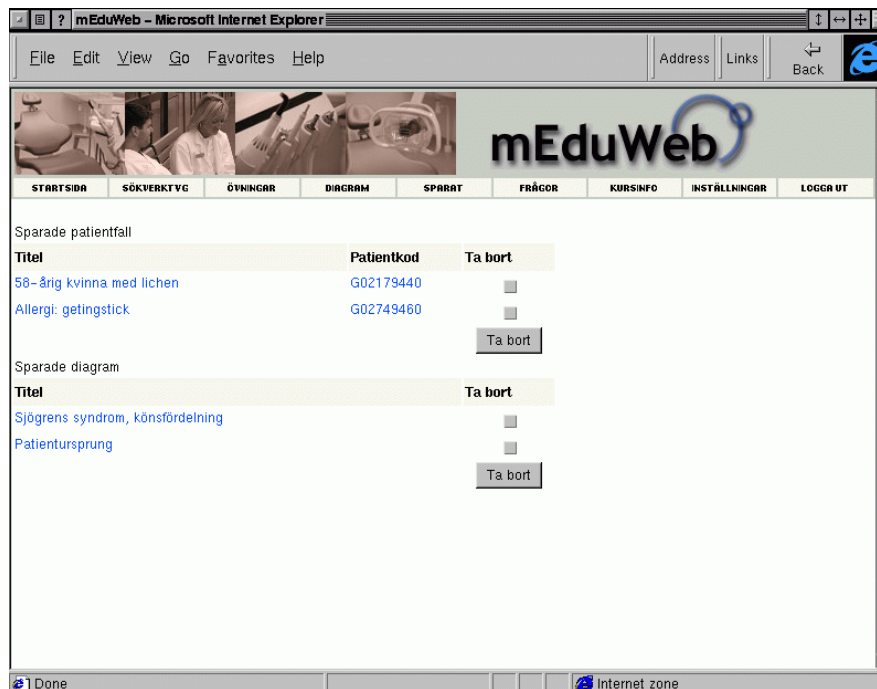
Diagram, steg 3

I steg tre visas det genererade diagrammet. En lista över de patienter som ligger till grund för jämförelsen visas om användaren klickar på ett segment eller en stapel i diagrammet. På denna sida ges också möjlighet att spara det genererade diagrammet och att ställa en fråga till kursadministratören. Diagrammet kommer då att visas tillsammans med frågan när den ska besvaras.

7.1.5 Sparat

På denna sida kan studenterna se de fall och diagram de tidigare sparade. Överst på sidan ligger listan av sparade fall med de beskrivningar studenten skrev in när fallet sparades. För varje fall visas även den patientkod som är relaterad till fallet. Både patientkoden och den egna benämningen på fallet är länkat till patientpresentationssidan.

Tidigare sparade fall kan tas bort genom att de fall som ska tas bort markeras och den övre 'Ta bort'-knappen sedan trycks ner. Nederst på sidan visas de diagram som studenten sparade. Även här visas den egna beskrivningen av diagrammet som också är länkat till själva diagrafbilden. Sparade diagram kan tas bort genom den nedre 'Ta bort'-knappen.



Sparade patientfall och diagram

7.1.6 Frågor

Frågedelen är uppdelad i fyra sidor och ger studenten en översikt över vilka frågor som ställts till kursadministratören och vilka svar som givits. Det finns fyra olika typer av frågor i systemet; allmänna frågor, frågor på fall, frågor på övningar samt frågor på diagram. Här syns även de eventuella svar administratören gett, samt länkar till de fall, övningar eller diagram frågan ställts på. Längst till vänster på varje frågesida finns en frågemeny som leder till sidorna för de olika typerna av frågor. Den första sidan som laddas är sidan med allmänna frågor till kursadministratören.

Allmänna frågor är sådana frågor som studenten ställt från sin startsida, dessa kan vara av vilken karaktär som helst och rör kanske ett tentamensdatum eller en föreläsning. På sidan visas en tabell som innehåller de frågor som ställts, samt de svar administratören gett. Om frågan inte är besvarad visas en text av typen "Din fråga har ännu inte blivit besvarad". Möjlighet finns att ta bort allmänna frågor, först markeras kryssrutan invid frågan och sedan trycks knappen 'Ta bort' ner. Frågan försvinner då ur systemet för både studenten och administratören.

Frågor på patientfall visas på liknande sätt som allmänna frågor. Här har dock studenterna ingen möjlighet att ta bort ställda frågor, detta är något som endast administratören kan göra. För varje fråga finns möjlighet att se det aktuella fallet genom att patientkoden, som är länkad till presentationen för just den patienten, visas längst till höger.

Frågor på övningar är nästan identisk med frågor på patientfall. Den enda skillnaden är att det inte finns någon länkad patientkod utan istället ett övnings-id som är länkat till den övning frågan är ställd på.

Diagramfrågor har nästan samma upplägg som frågor på patientfall och övningar,

med skillnaden att länken som finns till diagrammet gör att ett nytt fönster med det aktuella diagrammet öppnas.

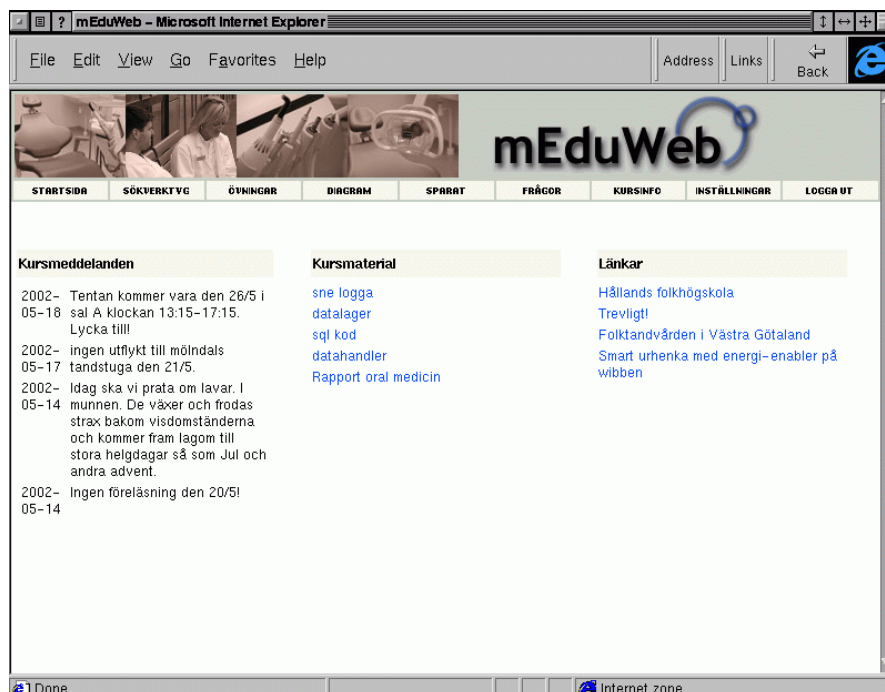
7.1.7 Kursinformation

Under sidan Kursinfo finns all information om kursen som kursadministratören har valt att lägga upp till studenterna. Då sidan innehåller förhållandevis mycket information är den tydligt uppdelad i tre kolumner med huvudrubrikerna Kursmeddelanden, Kursmaterial och Länkar.

Den första kolumnen, Kursmeddelanden, innehåller alla meddelanden som kursadministratören skickat till studenterna och även vilket datum meddelandet skickades. Meddelandena är sorterade utifrån datumet då de skickades och det senaste meddelandet visas överst i kolumnen för att studenterna på ett enkelt sätt ska kunna läsa den information som för tillfället är mest aktuell.

Under rubriken Kursmaterial finns en lista med dokument såsom föreläsnings-OH och rapporter som administratören vill att studenterna skall ha tillgång till. Varje dokument i listan är länkat och har en förklarande titel. Beroende på vilket format dokumentet har kan studenten ladda ner dokumentet till sin dator och sedan skriva ut det, alternativt kan materialet visas direkt i webbläsaren och skrivas ut därifrån.

Den sista kolumnen, Länkar, innehåller de länkar som administratören anser vara relevanta för kursen och som studenterna bör få tillgång till. Istället för att länken visar adressen till den webbplats den leder till så består länken av en kort förklarande text som beskriver webbplatsen. Den förklarande texten skrivs av administratören.



Kursinformation

7.1.8 Inställningar

Inställningssidan är till för att studenten skall kunna ändra sitt lösenord till systemet, samt även byta språk. Överst på sidan finns tre textfält för att ändra lösenord. I det första fältet fylls det nuvarande lösenordet i. De två andra fälten är till för det nya lösenordet, som alltså fylls i två gånger för att förhindra felskrivningar som resulterar i felaktiga lösenord. När samtliga fält är ifyllda genomförs ändringarna genom att 'Ändra'-knappen trycks ner.

Nederst på sidan finns två länkar, 'Svenska' och 'English'. När studenten klickar på en av dessa länkar ändras språkställningarna och alla sidor visas i fortsättningen med det valda språket.

7.2 Administratörens websidor

När administratören loggar in får denne tillgång till websidor som är specifika för administratören, men han eller hon har även möjlighet att utnyttja de sidor som studenterna använder sig av, dvs. sökverktyget och diagrammen. Eftersom dessa sidor redan har blivit beskrivna under punkten 7.1 kommer de inte att behandlas under denna punkt.

7.2.1 Start sida

Administratörens start sida innehåller, liksom studenternas start sida, en kort introduktion till mEduWeb och ett antal rubriker innehållande aktuell information. Det som skiljer sig är att introduktionen beskriver vad administratören kan göra på sidorna och att den aktuella informationen som visas är anpassad till vad administratören är intresserad av.



Administratörens start sida

De olika rubriker som finns på administratörens startsida är Senaste meddelandet, Nya frågor, Total loggad tid i systemet och Aktuella övningar. Precis som på studenternas startsida visar Senaste meddelandet den nyaste upplysningen som administratören har skickat ut. Under Nya frågor listas de frågor som har skickats av studenterna sen administratören senast loggade in. För att besvara en fråga klickar administratören på frågan och kommer då till den övning, det fall eller det diagram som frågan syftar till. Om frågan är av typen allmän, det vill säga om den inte syftar till ett fall, en övning eller ett diagram, leder länken till frågehanteringssidan där frågan kan besvaras. Den totalt loggade tiden i systemet visar hur mycket tid som systemet totalt har använts av både administratörer och studenter. För att läsa annan statistik om systemet kan administratören gå in på Användare via menyn. På startsidan visas dessutom de kursövningar som är aktuella. Övningarna försvinner från startsidan när de har blivit inaktuella eller borttagna av administratören, till skillnad från studenternas startsida, där en övning försvinner när studenten har gjort den. För att titta på hur en kursövning ser ut eller för att göra ändringar i densamma klickar administratören på övningstiteln.

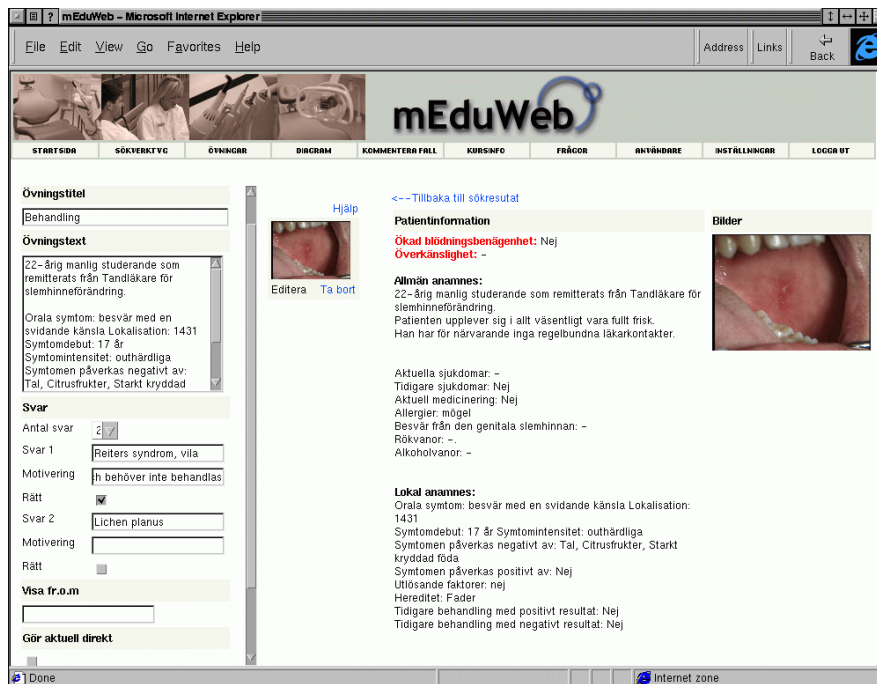
7.2.2 Övningar

Dessa sidor används när administratören skapar kursövningar till studenterna. En övning kan innehålla tre olika saker; text, bilder samt svarsalternativ med motiveringar. För varje övning väljer administratören ut ett godtyckligt antal bilder, skriver en frågetext samt gör 0 till 5 olika svarsalternativ. Det första som syns på övningssidorna är två listor. Den ena listan visar de övningar som är aktuella för studenterna just nu, och den andra visar de övningar som inte är aktuella. Övningarna i listorna är länkade och när en övning väljs laddas sidan med övningsskaparen med just den övningen, där möjlighet finns att ändra i övningen. Det finns även en knapp som öppnar en blank övningsskapare, för att kunna skapa nya övningar.

Övningsskaparen består av tre delar. En vänsterdel där information om övningen matas in, en mittdel där de bilder som hör till övningen visas, samt en högerdel som består av ett sökverktyg för att leta upp patienter och bilder.

Fallsökning och övningsskapning

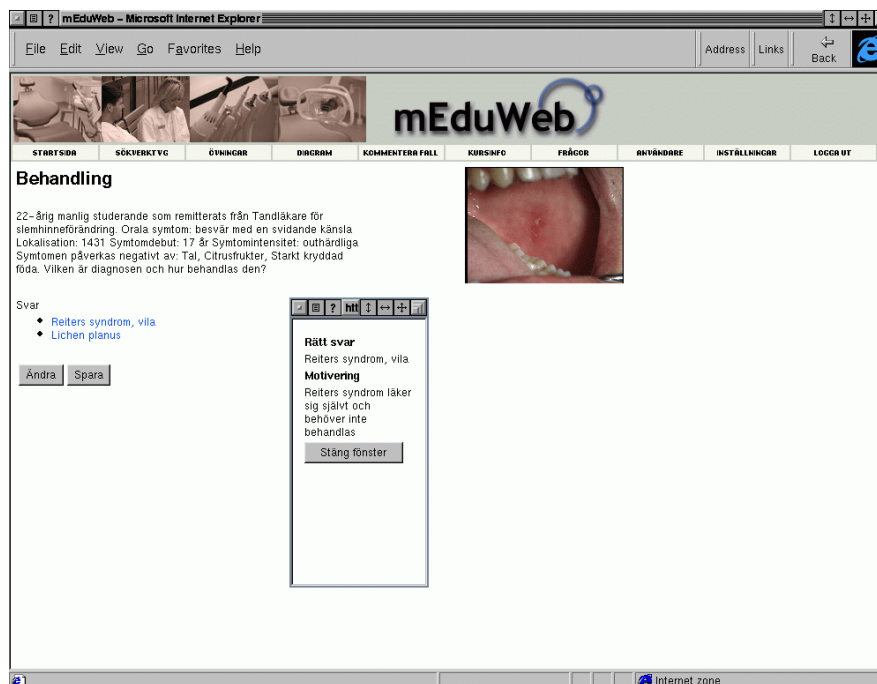
Vänsterdelen består av ett antal olika inmatningsfält. Överst ligger ett textfält där en titel för övningen matas in. Under titelfältet ligger ett textfält där själva övningstexten skrivs in, dvs. den uppgift som studenterna skall lösa. Därefter kommer avdelningen för svarsalternativ, här väljs hur många svarsalternativ som ska finnas på övningen, mellan 0 och 5 stycken. Beroende på hur många svarsalternativ som valts visas olika många inmatningsfält nedanför. För varje svarsalternativ finns två textfält och en kryssruta. Det första textfältet är själva svaret, här skrivs ett svarsalternativ till övningen in. Det andra fältet är en motivering till varför svaret är rätt eller fel. Kryssrutan är till för att markera om svaret är rätt eller inte. Avdelningen under svaren är till för att skriva in ett datum då övningen blir tillgänglig för studenterna, detta för att administratören skall kunna skapa övningar i förväg. Under detta datumfält finns en kryssruta som gör övningen tillgänglig för studenterna direkt. Längst ner i vänsterdelen finns tre knappar. En knapp 'Börja om' rensar alla fälten och låter administratören börja om med en ny övning. Knappen 'Ta bort' tar bort den aktuella övningen ur systemet. Knappen 'Förhandsgranska' leder till förhandsgranskningsidan, som beskrivs nedan.



Övningsskapning

Mittdelen är till för de bilder administratören valt att använda i övningen. Här visas varje bild i förminskad storlek och när en bild klickas på öppnas ett nytt fönster med bilden i originalstorlek. Till varje bild finns två alternativ; 'Editera' och 'Ta bort'. 'Ta bort' alternativet gör att bilden tas bort från den aktuella övningen. 'Editera' gör att ett nytt fönster öppnas. Detta fönster innehåller ett enkelt ritverktyg som ger möjlighet att rita cirklar, rektanglar, dra linjer samt skriva text i bilden. I ritverktyget finns en knapp 'Spara' som sparar den redigerade bilden.

Högerdelen fungerar precis som det vanliga sökverktyget, se punkt 7.1.2, med undantaget att om en bild klickas på så förstoras inte den bilden, utan infogas bland de redigerade bilderna i mittdelen.



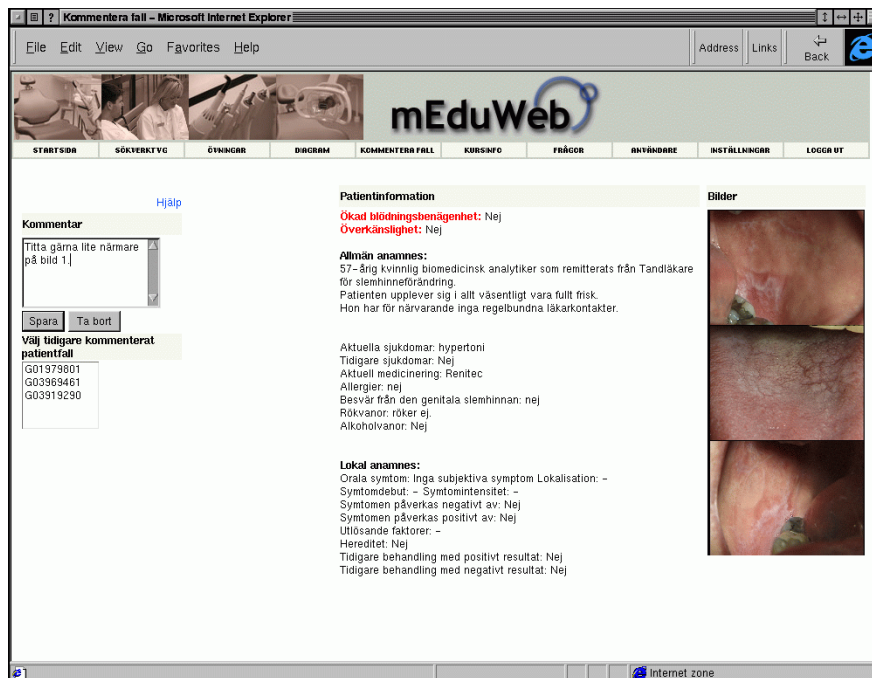
Förhandsgranskning

Den sista delen i övningskaparen är förhandsgranskningen, här får administratören se en sida som är nästan exakt likadan som den studenterna ser när de gör övningen, se 7.1.3. Det som skiljer är att administratören inte har möjlighet att ställa en fråga på övningen. Dessutom finns det två extra knappar på sidan. Den ena knappen, 'Ändra', leder tillbaka till övningsskaparen om administratören vill göra ändringar i övningen. Den andra knappen, 'Spara', sparar undan övningen och skickar tillbaka administratören till sidan som listar de övningar som finns i systemet.

7.2.3 Kommentarer

Administratören har möjlighet att lägga till kommentarer för att förtydliga vissa saker om en patient. Möjligheten finns även att göra markeringar i bilder för att t.ex. ringa in ett särskilt intressant område. Sidan består av tre delar, en vänsterdel för själva kommentaren, en mittdel där de redigerade bilderna finns, samt en högerdel som innehåller ett sökverktyg för att leta upp fall.

Vänsterdelen består av två huvudsektioner, en övre del där kommentaren till fallet skrivs in, samt två knappar. Knappen 'Spara' gör att kommentaren, samt eventuella redigerade bilder, sparas länkat till fallet så att kommentaren och bilderna kommer upp nästa gång någon tittar på just den patienten. Knappen 'Ta bort' gör att kommentaren tas bort för det aktuella fallet, även eventuella redigerade bilder tas bort. Den nedre sektionen består av en lista med patientkoder för de patienter administratören tidigare kommenterat. Om en patientkod i listan markeras laddas sidan om och kommentaren på patienten visas till vänster, eventuella bilder visas i mitten och patientpresentationen visas till höger.



Kommentering av patientfall

Mittdelen fungerar på liknande sätt som i övningsskaparen, se 7.2.2. Den enda skillnaden är att de bilder som redigerats och sparats kommer att ersätta originalbilderna när patientfallet visas nästa gång.

Högerdelen fungerar precis som det vanliga sökverktyget, se punkt 7.1.2, med undantaget att en bild inte förstöras då användaren klickar på den. Bilden infogas istället bland de redigerade bilderna i mittdelen.

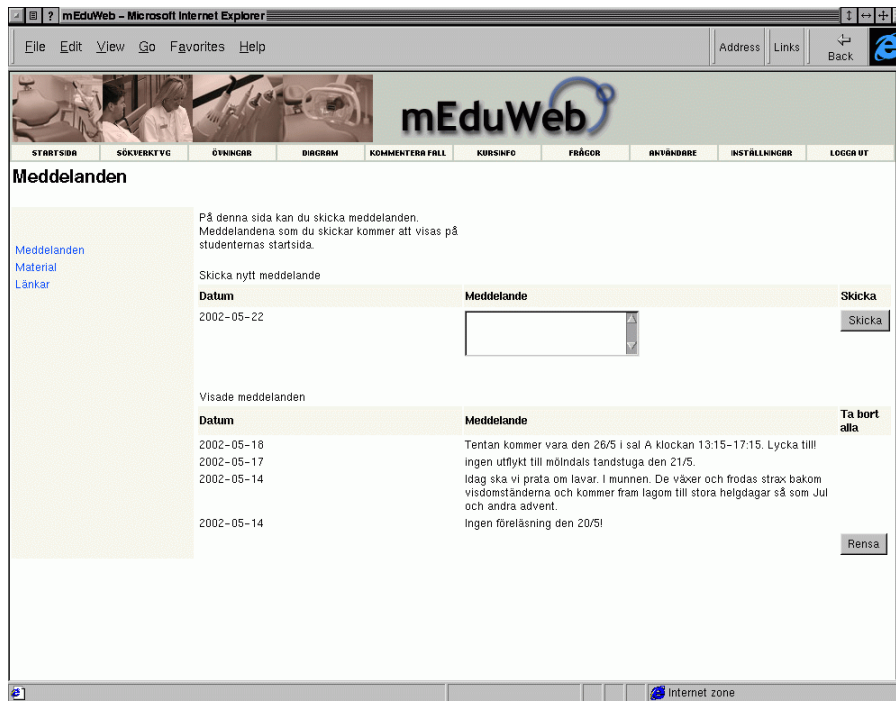
7.2.4 Kursinformation

Administratörens Kursinfo är uppdelad i tre sidor och innehåller en separat meny med länkar till de tre kursinformationssidorna Meddelanden, Material och Länkar. Från dessa sidor styr administratören den information studenterna skall få tillgång till.

Från början visas sidan Meddelanden och under denna sida kan administratören skriva korta meddelanden till studenterna om kommande föreläsningar, inställda lektioner osv. Meddelandet kopplas direkt till dagens datum och visas på studenternas kursinformationssida så fort det skickats från administratören. Det senaste meddelandet visas även på studenternas startsida. På meddelandesidan finns även en lista med alla tidigare meddelanden sorterade i datumordning med det senaste överst. När kursadministratören tar bort gamla meddelanden försvinner de även från studenternas sida. För att inga meddelanden som kan vara relevanta för studenterna ska kunna tas bort, har administratören endast tillgång till en knapp för att rensa alla meddelanden för att användas innan en ny kurs med nya studenter ska starta.

På sidan Kursmaterial kan administratören lägga upp dokument såsom föreläsnings-OH och rapporter som kan vara relevanta för studenterna. Administratören kan söka bland dokumenten på sin egen dator med hjälp av en dialogruta och sedan skriva en beskrivande text till materialet som kommer att visas både för studenterna och för

administratören själv. På sidan finns även en lista med allt material som tidigare lagts upp sorterat i datumordning med det senaste materialet sist och administratören har möjlighet att ta bort material som inte längre är aktuellt. Det senaste upplagda materialet visas på studenternas startside.



Kursinformation, meddelanden

På sidan Länkar kan administratören lägga upp nya kursrelaterade länkar som visas på studenternas kursinfosida. URL-adressen till webbplatsen skrivs in tillsammans med en beskrivning för att göra länken så tydlig som möjligt för studenterna. På sidan har administratören även en lista på alla de länkar som visas för studenterna, och de länkar som inte längre är intressanta kan enkelt tas bort.

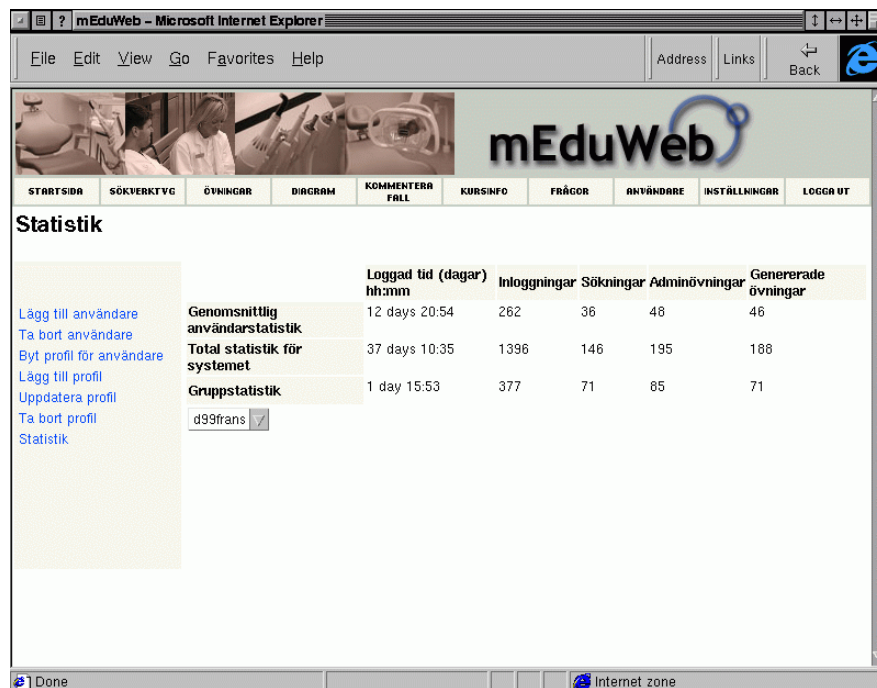
7.2.5 Frågor

På sidan Frågor visas frågor som studenterna har ställt. Studenterna kan ställa fyra olika typer av frågor, allmänna frågor, frågor på patientfall, frågor på övningar och frågor på diagram. Alla frågor sorteras in under någon av frågesidans två delar, besvarade frågor och obesvarade frågor.

Varje frågas titel är klickbar och när en fråga väljs så återskapas situationen där frågan uppstod. Om administratören klickar på en obesvarad fråga kommer den att visas tillsammans med en svarsruta där han eller hon kan besvara frågan. Svaret på frågan kommer att visas tillsammans med frågan på studenternas del av mEduWeb-systemet. Administratören kan även gå in och titta på besvarade frågor. Frågan visas då tillsammans med sitt svar. Det finns också möjlighet att ta bort frågor, dock inte av typen allmän eftersom de tas bort av studenterna själva. Frågan tas då bort från det patientfall, den övning eller det diagram där frågan först ställdes.

7.2.6 Användare

På sidan Användare finns det möjlighet att administrera användare och profiler. Det finns även möjlighet att titta på statistik över användandet av mEduWeb-systemet. Till vänster på sidan finns en meny med följande alternativ: Lägg till användare, Ta bort användare, Byt profil för användare, Lägg till profil, Uppdatera profil, Ta bort profil, Statistik.



Användare, statistik

Användare läggs till i grupp. Varje användargrupp innehåller ett obegränsat antal användare och har ett specifikt namn, t.ex. ht-02. För att lägga till en ny grupp användare anges gruppens namn, antal användare, profil och språk. Efter att administratören har tryckt på 'Skapa'-knappen kommer det valda antalet användare att skapas. De får var sitt användarnamn och lösenord.

Administratören kan välja att ta bort en grupp av användare eller en enskild användare. Grupper av användare tas bort med hjälp av gruppnamnet. Om administratören väljer att ta bort en grupp av användare kan han eller hon sedan välja om alla användare i gruppen ska tas bort, eller om borttagningen endast ska gälla en eller några användare. Enskilda användare tas bort med hjälp av användarnamnet.

Varje användare har en viss profil. Profilen anger vilka rättigheter en användare har och anges i samband med att administratören skapar användaren, men profilbyte är möjligt. Administratören kan byta profil på en hel grupp av användare eller på enskilda användare. Bytet går till på samma sätt oavsett om det påverkar en eller flera användare. Användare väljs ut med hjälp av gruppnamn respektive användarnamn och administratören får sedan ange den nya profilen.

För att lägga till en profil anger administratören ett profilnamn, samt kryssar i vilka

rättigheter som är aktuella för profilen. Det finns fyra olika rättigheter: rättighet att göra övningar, rättighet att ställa frågor, rättighet att läsa meddelanden och administratörsrättighet. Det går att rensa inställningarna och börja om från början genom att använda 'Rensa'-knappen, och för att lägga till profilen så används 'Lägg till'-knappen.

Administratören kan även ändra på existerande profiler. Först väljs aktuell profil ur en lista, sedan finns det möjlighet att kryssa i rättigheter enligt ovan. Inställningarna kan återställas genom att använda 'Återställ'-knappen. För att spara profilens nya inställningar så används 'Spara'-knappen.

Profiler kan tas bort på ett enkelt sätt. Administratören väljer aktuell profil ur en lista och trycker på 'Ta bort'-knappen.

På statistiksidan visas genomsnittlig användarstatistik och total statistik för systemet. För att visa hur systemet används av studenterna är administratörer inte medräknade i den genomsnittliga användarstatistiken. Den information som visas på sidan är: inloggad tid i dagar, timmar och minuter, antal inloggningar, antal sökningar, antal utförda kursövningar och antal övningar som studenterna genererat själva. Administratören kan även välja att titta på samma information för en speciell grupp.

7.2.7 Inställningar

På sidan Inställningar kan administratören ändra sina personliga inställningar. Han eller hon kan även ändra inställningar som påverkar hela mEduWeb-systemet. Till vänster på sidan finns en meny med följande alternativ: Personliga inställningar, kategorier, Ignorerade attribut och E-postlista.

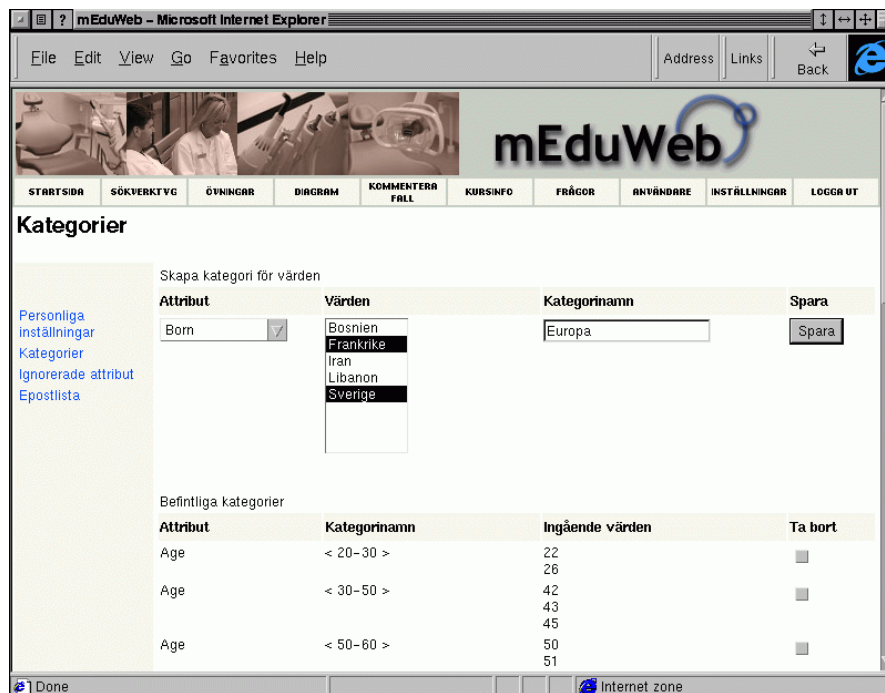
Personliga inställningar

Under Personliga inställningar kan administratören ändra sitt eget lösenord och språk.

Värdena på de olika attribut som finns i MedView-databasen kan delas in i kategorier. På kategorisidan kan administratören skapa nya kategorier, samt administrera de som redan finns. För att skapa en kategori väljs det attribut vars värden ska kategoriseras. Det valda attributets alla värden kommer då att visas. Administratören väljer sedan de värden som ska bilda den nya kategorin genom att klicka på dem. Ett kategorinamn måste också anges. Slutligen används 'Spara'-knappen för att spara den nya kategorin. Befintliga kategorier kan tas bort genom att administratören markerar dem och trycker på 'Ta bort'-knappen.

På sidan Ignorerade attribut kan attribut som inte ska vara sökbara väljas ut, t.ex. attributet Namn. De attribut som redan är markerade som ej sökbara visas också på sidan.

Under E-postlista finns det möjlighet att bestämma vilka e-postadresser som ska vara aktiva, vilket betyder att de meddelas då en fråga har ställts. Administratören kan lägga till och ta bort e-postadresser, samt se vilka e-postadresser som är aktiva för tillfället.



Skapande av kategorier

8. Vidareutveckling

En av de stora fördelarna med ett webbaserat och modulärt system som mEduWeb är att tillägg och förändringar är lätta att genomföra och att förändringarna får snabbt genomslag hos användarna av systemet. Eftersom ingen programvara behöver laddas ner av slutanvändaren kan denne omedelbart ta del av uppdateringar av systemet. Detta faktum leder till att mEduWeb med största sannolikhet kommer att utvecklas under många år och i flera omgångar. Förbättringar av befintliga funktioner kommer att göras och ny funktionalitet kommer att utvecklas.

8.1 Möjliga förbättringar av befintliga funktioner

Efter längre och mer ingående utvärderingar av mEduWeb, tillsammans med slutanvändare och beställaren, kommer med största sannolikhet en del förbättringar av befintliga systemdelar att behöva göras. Även om stora arbetsinsatser har gjorts under utvecklingen av systemet, kan mer djupgående utvärderingar leda till förändringar. Framför allt har den begränsade tiden för projektet inte gett möjlighet att utvärdera det färdiga systemet på ett tillfredställande sätt.

De förbättringar som kan behöva göras rör med största sannolikhet pedagogiken hos systemet. Förändringar av websidornas design och hjälptexternas utformning samt en utökning av hjälpfunktionerna med t.ex. fler exempel, kan behöva göras för att underlätta användandet av mEduWeb.

Systemet kan även behöva göras snabbare gentemot den enskilda användaren. Som det ser ut idag skickas stora mängder data, främst bilder, till användaren, vilket kan orsaka långa väntetider för de som använder mEduWeb via modem. Dessa väntetider kan förkortas genom att bland annat generera thumbnails, mindre kopior, av MedView-databasens bilder.

8.2 Förslag på utökningar

Endast fantasin och tekniken sätter gränser för ytterligare tillägg av funktionalitet i systemet. Ett par utökningar kan dock anses vara viktigare än andra:

- Kategorisering av MedView-databasens attribut för att kunna generera mer avancerade övningar.
- Mer dynamiska visualiseringar. Som det ser ut idag väljs attribut och värden ut av användaren och därefter genereras ett statiskt diagram. Diagram som kan manipuleras kontinuerligt, och ge en bättre uppfattning om samband mellan olika sjukdomstillstånd och andra faktorer, skulle kunna utvecklas med hjälp av Java Applets. I dessa skulle val av attribut och värden kunna ändras, med hjälp av skjutreglage samt knappar, och på så sätt förändra diagrammet medan det visas.
- Mer avancerade diagramformer. Ett alternativ till ovanstående diagram i Appletform är att utveckla de befintliga, statiska, diagrammen så att fler diagramtyper kan genereras. Det är även tänkbart att fler attribut med tillhörande värden kan användas i diagrammen.
- Möjlighet att begränsa vilka fall som visas för olika typer av användare.

- Möjlighet för administratören att redigera innehållet i patientfallen. Detta kan göras på flera sätt, med två huvudinriktningar. För det första kan de redigerade fallen sparas i mEduWebs databas. Fördelen med detta är att MedView-databasen inte påverkas och således är förändringarna inte permanenta. Nackdelen är att förändringar i MedView-databasen inte propagerar till de redigerade fallen. Det andra alternativet är att utveckla ett verktyg för att redigera innehållet direkt i MedView-databasen, vilket medför att förändringarna blir permanenta.
- Ett verktyg för att redigera de mallar som används för att presentera innehållet i patientfallen.
- Möjlighet att utföra examinationer med hjälp av systemet. Denna funktion skulle kunna innebära nya och bättre tentamensformer.
- Funktioner som möjliggör digital kommunikation mellan studenterna.

Det här är som sagt bara en delmängd av alla de utökningar som kan göras. Förhoppningsvis kommer en del av dessa att genomföras, tillsammans med flera andra utökningar som inte nämnts här, där fler övningstyper är ett exempel. mEduWeb av idag är en bra bas för ett beständigt undervisningssystem som hela tiden kan utvecklas.

9. Slutsatser

mEduWeb är idag ett färdigt undervisningssystem och finns installerat hos beställaren. De krav som ställdes i kravspecifikationen är uppfyllda och beställaren är nöjd med resultatet. Vi i projekt mEduWeb anser därmed uppgiften vara löst och vårt arbete avslutat.

Vi har inte fått anledning att ångra några av de konstruktionsval som gjorts under arbetets gång. Kombinationen med websidor kodade i JSP och Datahanteraren / Datalagret kodade i Java har visat sig vara mycket bra och har tillåtit en hög komplexitet på websidorna. PostgreSQL-databasen har fungerat tillfredsställande och visat sig vara både snabb och stabil. Lösningen med en XML-fil för språkoberoendet fungerar också den bra även om filen växt och blivit större än vi hade räknat med.

Det enda egentliga problemområdet har varit Apache Tomcat webservern, vilken har visat sig vara känslig för ändringar i importerade Java-klasser. Detta har under utvecklingsarbetet varit ett irritationsmoment eftersom många omstarter av webservern blivit nödvändiga när källkoden modifierats under körning. Dock är detta inte ett problem under normal körning av mEduWeb eftersom källkoden då aldrig ändras.

Projektgruppen har fungerat mycket bra och den projektplan, se Appendix A, som gjordes upp har i stort följts även om implementeringen drog ut på tiden. Den förlängda implementeringsfasen ledde i sin tur till att tiden för testning och utvärdering blev något begränsad. Utvärderingen visade ändå att systemet har alla förutsättningar att bli ett välanvänt hjälpmedel för tandläkarstudenterna på Odontologen.

10. Referenser

- [1] L.Hallnäs, P. Hammond, B.Magnusson, P. Holmstrup, M. Jontell, *MedView - A computerised teaching aid in oral medicine and oral pathology*, 2000
- [2] *HIVdent*, http://www.hivdent.org/slides/aphthous_ulcer_fisch.htm, 2002-05-23
- [3] *DermIS*, http://www.dermis.net/index_e.htm, 2002-05-23
- [4] *Pathology of oral ulcers*, <http://www.cal.eastman.ucl.ac.uk/ulcerspath/>, 2002-05-23
- [5] Java: <http://java.sun.com>, 2002-05-23
- [6] JavaBeans: <http://java.sun.com/products/javabeans/>, 2002-05-23
- [7] CGI: <http://www.w3.org/CGI/>, 2002-05-23
- [8] ASP och ASP.NET: <http://msdn.microsoft.com>, 2002-05-23
- [9] DHTML: <http://developer.netscape.com/tech/dynhtml/index.html>, 2002-05-23
- [10] Java Applets: <http://java.sun.com/applets>, 2002-05-23
- [11] PHP: <http://www.php.net>, 2002-05-23
- [12] JSP: <http://java.sun.com/products/jsp/index.html>, 2002-05-23
- [13] mySQL: <http://www.mysql.org>, 2002-05-23
- [14] PostgreSQL: <http://www.pgsql.com>, 2002-05-23
- [15] XML: <http://java.sun.com/xml/>, 2002-05-23
- [16] KavaChart: <http://www.ve.com>, 2002-05-23
- [17] Java-paketet mySmartUpload: <http://www.jspsmart.com>, 2002-05-23

Appendix A - Projektplan

Appendix B - Kravspecifikation

Appendix C - Studentintervju

Appendix D - Systemskiss

Appendix E - Funktionsbeskrivning

Appendix F - Prototyputvärdering

Appendix G - Enhetskonstruktion

Appendix H - Testspezifikation

Appendix I - Studentutvärdering

Appendix J - Flödesschema för websidorna