



Datavetenskap

Daniel Jansson och Mikael Jansson

Utvecklingen av ett tidregistrerings- och faktureringsystem

Examensarbete, C-nivå

2006:10

Utvecklingen av ett tidregistrerings- och faktureringsystem

Daniel Jansson och Mikael Jansson

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

Daniel Jansson

Mikael Jansson

Godkänd, 2006-06-07

Handledare: Hans Hedbom

Examinator: Stefan Lindskog

Sammanfattning

Vi har utvecklat ett tidregistrerings- och faktureringsystem åt Xeratech Systems AB i Karlstad. Målet med vårt arbete var att utveckla ett system som var mer användarvänligt och mer funktionsenligt än Xeratechs gamla system – XSIS. Huvudfunktionen med systemet är att kunna registrera tider som anställda på Xeratech arbetat på olika projekt, vilket sedan ska kunna sammanställas till fakturor. Vi har skrivit en kravspecifikation för systemet efter diskussion med Xeratech, och sedan implementerat tidregistreringsdelen av specifikationen. Resultatet av vårt arbete är ett väl fungerande system som fått positiv respons vid testkörning hos Xeratech. En MySQL databas har använts för att lagra all information om registrerade tider, projekt och kunder. Till utvecklingen av webbapplikationen användes .NET. Systemet färdigutvecklas på Xeratech och kommer senare ersätta deras gamla system.

The development of a timeregistration and invoicing system

We have developed a time registration and invoicing system for Xeratech Systems AB in Karlstad. The goal with our work was to develop a system that was more user friendly and more functional than Xeratechs old system – XSIS. The main function of the system is to be able to register time which the employees at Xeratech have worked in different projects, which can then be compiled into invoices. We have written a requirements-specification for the system after discussions with Xeratech, and then implemented the time registration part of the specification. The result of our work is a well performing system which has gotten a positive response from testing at Xeratech. A MySQL database has been used to store all information regarding registered time, projects and customers. For the development of the web application .NET was used. The system will be finished at Xeratech and will later replace their old system.

Innehåll

1	Introduktion	1
1.1	Disposition	2
2	Bakgrund	3
2.1	Existerande system	3
2.1.1	Varför ett egenutvecklat system?	5
3	Avgränsad kravspecifikation	7
3.1	Översikt	7
3.2	Inloggning och menysystem	8
3.3	Tidregistrering	8
3.3.1	Transaktionsöversikt	9
3.3.2	Lägg till transaktion	11
3.4	Admininställningar	14
3.5	Projekt	15
4	Verktyg	16
4.1	.NET	18
4.2	JavaScript	19
4.3	MySQL	21

5	Databasdesign	22
5.1	E/R-diagram	22
5.1.1	Entiteter	23
5.1.2	Relationer	27
5.2	Mappning till relationsmodellen	28
5.3	De färdiga tabellerna	31
5.4	Kapitelsammanfattning	37
6	Design och implementation	38
6.1	Implementation	38
6.1.1	Databasen	40
6.1.2	Inloggning	41
6.1.3	Menyn	42
6.1.4	Admininställningar och projekt	43
6.1.5	Tidregistrering	44
6.2	Användargränssnitt	45
6.2.1	Inloggning	46
6.2.2	Användaröversikt	46
6.2.3	Lägg till användare	47
6.2.4	Transaktioner - Allt i lista	47
6.2.5	Lägg till transaktion	49
6.3	Kapitelsammanfattning	49
7	Resultat och framtida arbete	51
7.1	Problem	53
7.2	Framtida arbete	53
	Litteraturförteckning	55

A	Kravspecifikationen	56
A.1	Menysystem	56
A.2	Login	57
A.3	Tidregistrering	57
A.3.1	Översikt	57
A.3.2	Registrera ny tid	61
A.4	Fakturering	64
A.5	Attestering	66
A.6	Admininställningar	67
A.6.1	Användare	67
A.6.2	Systeminställningar	70
A.6.3	Kunder	71
A.7	Projekt	72
A.7.1	Översikt	72
A.7.2	Lägg till projekt	73
A.7.3	Editera projekt	74
A.8	Mina egenskaper	75
A.8.1	Inställningar	75
A.8.2	Projektanmälningar	75
A.8.3	Ny projektanmälan	75
A.9	Om	76

Figurer

3.1	Översiktsdiagram av systemet	8
3.2	Allt i lista	9
3.3	Lägg till transaktioner	11
4.1	.NET Framework	18
4.2	En enkel tabell och dess DOM	20
5.1	exempel av E/R-diagram	23
5.2	E/R-diagram	26
6.1	Klassdiagram över systemet	39
6.2	Exempel på hur databindningen går till	40
6.3	Exempel på hur uppdatering av data sker	41
6.4	Den färdiga loginsidan	46
6.5	Den färdiga “Användaröversikt”-sidan	47
6.6	Den färdiga “Lägg till användare”-sidan	48
6.7	Den färdiga “Transaktionsöversikt”-sidan	48
6.8	Den färdiga “Lägg till transaktion”-sidan	49
A.1	Xeratechs menysystem	56
A.2	Loginrutan	57
A.3	Allt i lista	58

A.4 Kund-projekt översikt	60
A.5 Kalender	61
A.6 Lägg till nya tider	61
A.7 Fakturering	65
A.8 Allt i lista	66
A.9 Användaröversikt	68
A.10 Lägg till användare	68
A.11 Systeminställningar	70
A.12 Kundöversikt	71
A.13 Lägg till kund	71
A.14 Projektöversikt	72
A.15 Lägg till projekt	73
A.16 Prisrutan vid avtal	75

Tabeller

5.1	customer	31
5.2	price_type	32
5.3	agr_payment	32
5.4	project	32
5.5	user	33
5.6	access_level	33
5.7	start_page	33
5.8	project_phase	34
5.9	work_type	34
5.10	transaction	35
5.11	extra	35
5.12	status	36
5.13	supplier	36
5.14	cost_for_worktype	36
5.15	signed_up	37

Kapitel 1

Introduktion

Att kunna skriva in och lagra data om anställdas arbete och arbetstid är en viktig del av arbetet på Xeratech Systems AB i Karlstad. Denna information används för att bland annat kontrollera utgifter och inkomster, och till att generera fakturor till kunder. Det system som tidigare använts av Xeratech innehåller flertalet brister och de som använder det finner systemet långsamt och felanpassat för deras behov (se kapitel 2).

Syftet med detta arbete är att designa ett tidregistrerings- och faktureringsystem som uppfyller kraven ställda av Xeratech Systems AB i Karlstad, och sedan implementera tidregistreringsdelen. De delar av kravspecifikationen som återstår (attestering¹- och faktureringsdelen) kommer sedan utvecklas på Xeratech. Som en del i arbetet ingår även att ta fram en detaljerad kravspecifikation efter diskussion med de som ska använda systemet. Denna rapport tar upp de steg vi gått igenom för att nå det slutgiltiga systemet – val av verktyg, kravspecifikationen, databasen, design och implementation, och till sist resultatet av vårt arbete och en blick framåt i tiden.

De första veckorna av arbetet togs upp av insamling av information till systemet och skrivande av kravspecifikationen. I slutet av skrivandet togs beslut om vilka

¹Med attestering menas här att godkänna registrerade tider, vilket måste ske innan fakturering kan äga rum

utvecklingsverktyg som skulle användas, och en kärna av kravspecifikationen valdes. Kärnan består av de delar som vi senare implementerade och står att läsa i den avgränsade kravspecifikationen (se kapitel 3). När specifikationen var klar designades en databas med hjälp av E/R-diagram och relationsmodellen (se kapitel 5). Databasen används till att lagra all data om tiden som användare registrerar och vilka kunder och projekt som finns. Efter databasens färdigställande och implementation med MySQL började vi utvecklandet av webbsidorna i utvecklingsmiljön .NET. Resultatet av vårt arbete är ett fungerande system för tidregistrering som fått positiv respons från de på Xeratech som testat det, både vad gäller användargränssnitt och funktionalitet. Systemet kommer nu fortsätta utvärderas i Xeratechs testmiljö och sedan vidareutvecklas så att det i framtiden kan ta över efter deras gamla system – XSIS².

1.1 Disposition

I kapitel två går vi igenom bakgrunden till arbetet – varför vi utvecklar systemet. Kapitel tre innehåller en överblick över systemet, och här beskrivs den avgränsade kravspecifikationen som är en kort version av den del av kravspecifikationen som vi implementerat. I kapitel fyra tar vi upp de verktyg som använts under utvecklingen, motivationen bakom valet av dessa, och en kort beskrivning av vardera. I kapitel fem diskuteras alla delar av utvecklingen av databasen med E/R-diagram och relationsmodeller. Kapitel sex är en beskrivning av hur systemet implementerades, vilka delar som implementerades, och vilka val vi gjorde och varför. Här visas även bilder från det färdiga systemet. I kapitel sju redovisas hur vårt arbete utfallit, vilka problem vi stött på, vad som finns kvar att utveckla på systemet, och delar som inte står med i kravspecifikationen men skulle kunna tillföra något i framtiden.

²Det nya systemet har i skrivande stund inget namn utan endast utvecklingsnamnet “XSIS 2.0”

Kapitel 2

Bakgrund

Detta examensarbete är utfört åt Xeratech Systems AB. Xeratechgruppen består av totalt sju bolag med cirka femtio medarbetare som samverkar för att, enligt dem själva, “ge sina kunder den bästa totallösningen och konkurrenskraften”. Holdingbolaget är ett helägt familjebolag som har sitt ursprung i Karlstad. Genom ett brett utbud av IT- och dokumentrelaterade tjänster och produkter skapar företaget ökad effektivitet för sina kunder. En stor del av Xeratechs arbete består av konsultjobb i olika projekt hos olika företag. För att hålla reda på hur många timmar som varje konsult arbetat på respektive projekt, och sedan enkelt få ut fakturor på detta, utvecklade Xeratech ett tidregistrerings- och faktureringsystem som de kallade XSIS – Xeratech Systems Invoicing System. Detta system visade sig dock ha flertalet brister i både användargränssnittet och den bakomliggande logiken. Det ansågs rimligt att bygga om systemet från grunden, för att få ett system som från början var byggt efter användarnas krav.

2.1 Existerande system

Systemet som används idag, XSIS, utvecklades av en person på fritiden. En användare kan logga in i XSIS och registrera den tid som användaren jobbat och ange i vilket

projekt användaren jobbat, vilken kostnaden var, vad för typ av arbete som utförts, om det ska komma med på fakturan, och flera nivåer av beskrivningar av arbetet som utförts. En administratör kan sedan attestera användarnas registrerade tider, det vill säga godkänna dem för fakturering. Faktureringen sker genom att administratören väljer vad som ska faktureras och får sedan ut en faktura i pdf-format.

XSIS har dock många brister som det nya systemet ska lösa:

- Systemet är väldigt långsamt, även när det anropas över ett LAN i byggnaden.
- Standardvärden saknas, vilket gör att om man registrerar en transaktion varje dag och alltid fyller i samma värde på till exempel projektfas, måste detta ändå göras varje gång. Detta är tidsödande, speciellt när det handlar om något som ska fyllas i tiotals gånger per vecka, eller kanske till och med flera gånger per dag!
- I vissa formulär behöver man veta exakt information från någon viss tabell, till exempel projektnumret till ett visst projekt, vilket gör att man tvingas hoppa mellan sidor för att ta reda på denna information.
- Det går bara att registrera en transaktion åt gången. Om man har femtio transaktioner som ska registreras blir det väldigt mödosamt. Det tillvägagångssätt som finns för att göra processen snabbare är att kopiera en transaktion, men man tvingas fortfarande editera den transaktionen och ändra de värden som ska ändras.
- Det går bara att attestera en transaktion åt gången. Om administratören har flera hundra transaktioner att attestera kommer detta att ta väldigt lång tid om varje transaktion måste attesteras för sig.
- Projektledare kan inte attestera registrerade tider.
- Möjlighet att fakturera avtalsprojekt saknas och måste alltså göras manuellt.

- Databasen innehåller viss onödig och/eller felplacerad information, till exempel kan varje enskild transaktion ha egenskapen avtal trots att alla transaktioner i ett avtalsprojekt ska ha den egenskapen.
- Det är svårt att navigera bland sidorna eftersom menysystemet inte visar var i systemet man för tillfället befinner sig. Menyn består av två rader av knappar, där den undre raden beror av den övre. Det går dock inte att se vilken knapp som är vald i vare sig den övre eller undre raden.

XSIS är dock inte helt utan fördelar. Även om flera tabeller kan vara svåra att tyda på grund av den stora mängd kolumner de innehåller är den mesta av informationen av sådan art att den behöver visas. Alla tabeller har även möjlighet att sökas genom att fylla i textrutor ovanför kolumnerna, vilket är en bra funktion eftersom mängden data är mycket stor.

2.1.1 Varför ett egenutvecklat system?

Varför utveckla ett eget tidregistreringssystem när det redan finns färdiga kommersiella system? System som TimeLog är uppbyggda i komponenter, och kan anpassas efter kundens behov. TimeLog erbjuder även undervisning i hur systemet ska användas, och har mängder av komponenter för att lättare sammanställa information, få ut grafer, planera tid, göra rapporter, och många andra användbara funktioner för ett tidsregistreringssystem. Ett system utvecklat av experter inom området borde rimligen vara robust, säkert, funktionellt och användarvänligt, vilket är egenskaper som eftersöks [8].

Med ett egenutvecklat system kan alla aspekter av systemet designas för att passa företagets behov. Även om vissa kommersiella system säger sig kunna anpassas efter kundens behov kan denna nivå av valfrihet aldrig mäta sig med den totala kontroll som fås av ett egenutvecklat system. I Xeratechs fall hade de även en färdig struktur på

information som var tvungen att finnas i databasen. Detta leder till att ett färdigt system skulle vara tvunget att innehålla denna struktur. Sen är även kostnaden en stor aspekt att ta hänsyn till. Ett kommersiellt system hyrs ofta per månad för en viss kostnad. Genom att utvecklandet av detta system gjordes som ett examensarbete av två studenter kunde denna kostnad undvikas. Kostnaden för intern vidareutveckling av systemet kvarstår dock, men vid den tiden finns redan ett relativt komplett system.

Kapitel 3

Avgränsad kravspecifikation

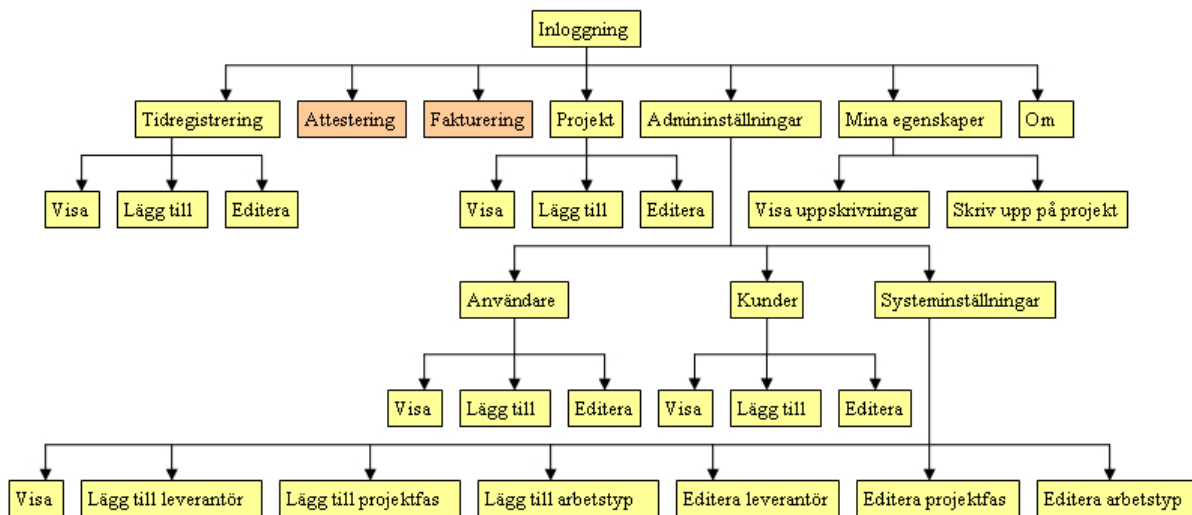
Det första vi fick göra när arbetet med tidregistreringssystemet började var att samla in en stor mängd information relevant till systemet vi skulle utveckla. Vi gick igenom det gamla systemet och observerade dess svagheter, vi undersökte liknande kommersiella system, och vi diskuterade med de som skulle komma att använda systemet vilka krav de hade. Därefter började vi skissa upp de viktiga delarna i systemet, och skrev sedan kravspecifikationen utifrån dessa. Den slutgiltiga och kompletta kravspecifikationen finns att läsa i appendix A.

Specifikationen för tidregistreringsdelen följer här i mer detalj, medan de andra delarna som vi implementerade är sammanfattade. All information som finns i tabellerna som beskriver bland annat företag, personer och priser är fingerad.

3.1 Översikt

Systemet består av en mängd fristående webbsidor som kopplas samman med en meny vilken visas längst upp på alla sidor. I figur 3.1 visas en karta över de sidor som bygger upp systemet. Rutorna med mörkare bakgrund är inte implementerade, utan blir implementerade senare av utvecklarna på Xeratech. Det är endast rutorna på den lägsta

nivån i träden i figuren som symboliserar webbsidor (med undantag för inloggningssidan som står överst), de andra rutorna är grupperingar eller indelningar vilket sammanfaller med katalogstrukturen som används.



Figur 3.1: Översiktsdiagram av systemet

3.2 Inloggning och menysystem

För att komma åt någon av systemets funktioner måste användaren logga in i systemet på en login-sida. Därefter kan användaren manövrera sig genom systemet via menyn längst upp på sidorna. Användarens rättigheter anger vilka sidor denne har tillgång till.

3.3 Tidregistrering

På tidregistreringssidorna kan användaren få en överblick över sina registrerade sidor, registrera nya tider, och editera tider som registrerats förut.

3.3.1 Transaktionsöversikt

Kund	Projekt	Projektfas	Datum	Arbetstyp	Extra	Fakturareferens	Kvanbitet	Tid kvar	Å Pris	Kostnad	Total	Omkostnader	Avtalstotal	Status
Gabab AB	Tapetutskrift	Konfigurering	060110	Konsult	Avtal		4h	20h	500	2000	20000	-	70000	● 🟢 ✖
Gabab AB	Tapetutskrift	Konfigurering	060109	Konsult	Avtal		8h	20h	500	4000	20000	-	70000	● 🟢 ✖
Gabab AB	Färgsampling	Konfigurering	060106	Konsult	Normal		8h	16h	500	4000	17000	-	-	● 🟢 ✖
Gabab AB	Färgsampling	Konfigurering	060105	Konsult	Normal		8h	16h	500	4000	17000	-	-	● 🟢 ✖
Stora Enso	Papperssorteringssystem	Konfigurering	060104	Konsult	Internutgift		8h	57h	-	4500	20000	4500	-	● 🟢 ✖
Stora Enso	Papperssorteringssystem	Konfigurering	060103	Konsult	Normal		8h	57h	300	2400	20000	4500	-	● 🟢 ✖
Stora Enso	Papperssorteringssystem	Konfigurering	060102	Konsult	Normal		8h	57h	300	2400	20000	4500	-	● 🟢 ✖
							52h	93h		23300	57000	4500	70000	

Figur 3.2: Allt i lista

Här visas alla användarens registrerade tider i en tabell (figur 3.2). På höger sida om varje rad finns en knapp för att editera posten, och en för att ta bort den. De tider som attesterats eller fakturerats går inte att editera eller ta bort som det går med de andra tiderna. Genom att klicka på titeln till en kolumn sorteras alla rader efter detta attribut. På varje rad visas:

- Vilken kund arbetet utförts åt
- I vilket projekt det utförts
- I vilken projektfas det utförts
- När arbetet utfördes (vilket datum)
- Vilken typ av arbete som utförts
- Om transaktionen har någon extrauppgift (till exempel om det är en intern avgift så anges det här)
- Om det finns någon fakturareferens
- Hur många timmar som arbetats
- Hur många timmar som är kvar i det budgeterade antalet timmar för det projektet

- Kostnaden per tidsenhet/avståndsenhet
- Kostnaden för transaktionen
- Den totala kostnaden inom projektet
- Den totala summan omkostnader i projektet (omkostnader är ej en delmängd av den totala kostnaden inom projektet)
- Den totala summan för avtalet om det är ett avtalsprojekt

Statuskolumnen förtäljer vilken status posten har:

gul boll – Registrerad, men inte attesterad

blå boll – Attesterad av projektledaren

grön boll – Attesterad av admin

röd boll – Underkänd

ett F – Fakturerad

ett D – Delfakturerad, det vill säga att minst en avbetalning på ett projekt med prisavtal är gjort, och minst en betalning är kvar

ett I – Attesterad av admin och är intern

ett A – Attesterad av admin och anger att det finns ett prisavtal

Om användaren trycker på editera-knappen, som står till höger om status-ikonen, kommer han/hon till en annan sida där all information om den posten visas, och går att ändra på. Denna knapp går dock inte att trycka på om denna post blivit attesterad av projektledare eller admin. Om användaren trycker på “ta bort”-knappen, som står till höger om editera-knappen, kommer det upp ett meddelande som frågar “Vill du verkligen

ta bort den här posten?”. Om användaren klickar på “Ja” tas posten bort, men om användaren klickar på “Nej” händer ingenting. Denna knapp går dock inte att trycka på om denna post blivit attesterad av projektledare eller admin. Posten kan endast tas bort om den har status underkänd eller registrerad.

På den sista raden i tabellen visas totaler för kolumnerna kvantitet, tid kvar, kostnad, total, omkostnader, och avtalstotal.

3.3.2 Lägg till transaktion

Kund: Xeratech Systems AB	Projekt: Tidregistreringssystem	Projektfas: Konfigurering	Upparbetat: 64/200h
Datum: 060101	Arbetstyp: Konsult	å Pris: 500 kr	Kvantitet: 8 h
Extra: Normal	Fakturareferens:		Kostnad: 5000 kr
Beskrivning:	Fortsatt utvecklingen av specifikationen		
Utförlig beskrivning:	Uppdaterat tidregistreringsdelen till att kunna fylla i flera samtidigt		
Intern information:	Denna sida är bara till för specifikationen, och kommer se bättre ut efter implementering.		
Datum: 060102	Arbetstyp: Konsult	å Pris: 500 kr	Kvantitet: 8 h
Extra: Normal	Fakturareferens:		Kostnad: 5000 kr
Beskrivning:	Fortsatt utvecklingen av specifikationen		
Utförlig beskrivning:	Uppdaterat tidregistreringsdelen till att kunna fylla i flera samtidigt		
Intern information:	Denna sida är bara till för specifikationen, och kommer se bättre ut efter implementering.		
Kopiera			Registrera

Figur 3.3: Lägg till transaktioner

På denna sida kan användaren registrera en eller flera tider samtidigt. Om användaren registrerar flera tider får alla dessa samma kund, projekt och projektfas. För att få fler poster att fylla i klickar användaren på kopiera-knappen varefter ett popup-fönster visas där användaren får fylla hur många kopior av den nedersta posten som ska skapas. Efter att ha tryckt på OK-knappen läggs det till det antal poster som användaren valt med samma information som den nedersta posten, och där datumen är stigande. Man kan stega sig fram mellan alla attribut, och genom att hovra muspekaren över ett objekt visas en förklarande text om just det objektet. Alla uppgifter utom fakturareferens och beskrivningarna är obligatoriska.

När användaren skrivit in sina tider trycker denne på registrera-knappen, varefter ett popupfönster frågar om användaren vill fortsätta. Väljer denne "Ja" sparas informationen i databasen och användaren flyttas till transaktionsöversiktssidan. Väljer han/hon "Nej" försvinner popupfönstret och inget annat händer.

Kund – Kunden väljs i en rullista fylld med alla kunder som har projekt som användaren är uppskriven på. När en kund väljs i denna lista uppdateras projektlistan med de projekt som tillhör den valda kunden.

Projekt – Projektet väljs i en rullista fylld med de projekt som användaren är registrerad på samt associerade med den valda kunden. Om ingen kund är vald är denna lista tom. När en kund väljs och denna lista uppdateras med projekt, sätts det projekt med lägst projektnummer som standardvärde. Om ett projekt med prisavtal väljs, väljs avtal i Extra-listan automatiskt.

Projektfas – Projektfasen är en delaktivitet av ett projekt, där exempel skulle kunna vara konfigurerings, validering, eller löpande stöd. Användaren väljer projektfasen i en rullista där värdena i listan är samma för alla projekt. Om ett standardvärde för projektfas för ett projekt har angivits av projektledaren eller admin, är rullistan satt till detta.

Upparbetat – En text visar hur många timmar som upparbetats totalt inom projektet, och hur många timmar som är beräknat.

Datum – Användaren kan skriva in datumet för det som rapporteras direkt i en textbox, där det som standardvärde står dagens datum. Användaren kan även klicka på en ikon bredvid textboxen och få upp en kalender ur vilken han/hon kan välja relevant datum.

Arbetstyp – Arbetstypen säger vilken roll användaren haft under den tid som registreras, till exempel konsult, projektledare, restid, eller milersättning.

Användaren väljer arbetstyp ur en rullista där värdena i listan är samma för alla projekt. Om ett standardvärde för arbetstyp i detta projekt är satt av projektledaren eller admin, är rullistan satt till detta.

á Pris – Kostnaden per arbetsenhet för det utförda jobbet. Har som standardvärde det värde som är satt för den valda arbetstypen i det valda projektet.

Kvantitet – Antalet timmar användaren jobbat, eller antal km han/hon rest.

Användaren skriver in kvantiteten i en textbox, till höger om vilken det står antingen h eller km, beroende på vilken arbetstyp som valts.

Extra – I extra-listan kan användaren välja en extra egenskap för den registrerade tiden. Det finns 7 val:

Normal – Anger att denna tidregistrering ska faktureras normalt. Denna är förvald om det inte är ett avtalsprojekt.

Intern – Anger att denna tidregistrering är intern, och ska inte faktureras kund. Den kommer inte med på fakturan.

Avtal – Anger att detta projekt har ett fast pris kopplat till sig, och tidregistreringen ska inte faktureras kund. Denna väljs automatiskt om projektet som valts har ett avtalat pris, och detta val kan inte ändras. Den kommer inte med på fakturan.

Kundutgift – Anger att denna tidregistrering är ett utlägg som ska faktureras kund. Den kommer med på fakturan.

Internutgift – Anger att denna tidregistrering är ett utlägg som inte ska faktureras kund. Den kommer inte med på fakturan.

Kundreseutgift – Anger att denna tidregistrering är en restid eller milersättning som ska faktureras kund. Den kommer med på fakturan.

Internreseutgift – Anger att denna tidregistrering är en restid eller milersättning som inte ska faktureras kund. Den kommer inte med på fakturan.

Fakturareferens – Användaren väljer en referens i en rullista. Standardvärdet är ingenting, det vill säga tomt. Denna ska endast användas om någon annan än användaren ska vara med på fakturan som referens.

Kostnad – Kostnaden för det registrerade arbetet står i en textbox och har som standardvärde priset för arbetstypen multiplicerat med kvantiteten.

Beskrivning – Användaren kan skriva in en kortfattad beskrivning om vilket arbete som utförts i en textbox. Denna text skrivs med på fakturan.

Utförlig beskrivning – Användaren kan skriva in en extra rad beskrivning i en textbox, som bara skrivs med på fakturan om kunden vill ha det.

Intern information – Användaren kan skriva in interna meddelanden i en textbox. Dessa kommer inte med på fakturan. Om en tidregistrering underkänns av projektledare eller admin kommer denna text skickas som e-post till ägaren av den underkända tidregistreringen.

3.4 Admininställningar

Dessa sidor har endast admin tillgång till och innehåller:

Användarsidor – Admin kan lista alla användare i systemet, lägga till nya användare, och editera redan existerande användare.

Systeminställningar – Admin kan ändra inställningar för leverantörer, arbetstyper och projektfaser. För arbetstyper kan även ett standard timpris sättas.

Kundsidor – Admin kan lista alla kunder i systemet, lägga till nya kunder, och editera redan existerande kunder.

3.5 Projekt

Endast admin och projektledare har tillgång till dessa sidor, och kan här lista alla projekt i systemet, lägga till nya projekt, och editera redan existerande projekt. I ett projekt går det att ställa in vilken pristyp som ska användas (timmar, dagar, veckor, eller avtal), och specificera priser per arbetstyp. Om pristypen avtal väljs får användaren välja antalet delbetalningar som kunden ska göra, och hur mycket som ska betalas vid varje delbetalning.

Kapitel 4

Verktyg

De primära delarna i tidregistrerings- och faktureringsystemet är ett användargränssnitt som nås via webbläsare och något system för lagring och manipulering av information. Statisk data visas enkelt med HTML, men vårt system kräver även dynamisk funktionalitet – att kunna visa tabeller med information lagrad på servern, att kunna manipulera samma information, att bara ge tillgång till användare som kan logga in i systemet med rätt användarnamn och lösenord, att kunna validera formulärdata på klientsidan, att göra ändringar i sidor utan anrop till servern, och att kunna generera sidor dynamiskt.

Med dessa systemkrav i åtanke stod det klart att vi behövde använda minst tre språk för visning av data, och ett databassystem för lagring av data. Förutom HTML behövs ett språk för dynamisk sidhantering på klientsidan och ett på serversidan.

När vi började utvecklingen av systemet hade vi fått som preliminärt besked att vi kunde utveckla systemet i Java. Detta var ett önskemål från vår sida, och motiverades av flera faktorer:

- Vi har utvecklat i Java förut, och funnit det att vara väl genomtänkt, enkelt att använda, och givit bra resultat. Tack vare vår tidigare erfarenhet av utveckling i Java skulle tiden för inläring av nya språk och dylikt vara minimal och vi skulle

hinna med att implementera fler delar av specifikationen.

- Java är plattformsoberoende vilket gör att servern där systemet ligger skulle kunna ha vilket operativsystem som helst [6].
- Med Java kan man dela upp serverlogiken och klientdelen på ett smidigt sätt, och särskilja på användargränssnittet och serverlogiken för att underlätta underhåll.

Det beslutades dock senare av företagsledningen på Xeratech att systemet måste utvecklas i .NET. Beslutet var baserat på en IT-strategi som Xeratech måste följa för att behålla enligheten i sin IT-miljö. Eftersom Xeratech har expertis inom .NET kan de lättare fortsätta utvecklingen av systemet efter att vårt arbete är slutfört. Vi kan även få hjälp av dem om vi skulle fastna på någon del.

Språken för dynamisk sidhantering var alltså nu tvungna att vara en del av .NET. På serversidan används normalt två språk i .NET – ASP och C# eller ASP och Visual Basic. Vi valde C# för att det är relativt likt Java.

Till vissa dynamiska delar på klientsidan av systemet behöver vi använda oss av JavaScript. Bland annat behövs JavaScript för att dynamiskt ändra vilken sida ett formulär skickar sin information till, eftersom .NET sätter som standard att alla formulär skickar sin information till samma sida som det kommer ifrån (läs mer om detta i 6.1.4). För att kunna använda systemet måste JavaScript vara påslaget i webbläsaren.

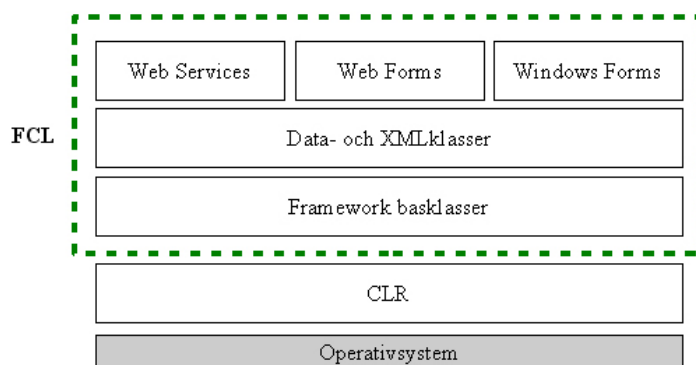
För att lagra all data om användare, registrerade tider, kunder med mera, är det motiverat att använda en databas. MySQL är ett databassystem vi använt tidigare och känner oss vana vid. Dessutom är det enkelt att installera, och GPL-versionen är gratis att ladda ner.

I 4.1-4.3 står korta introduktioner till vad .NET, JavaScript och MySQL är för något.

4.1 .NET

.NET är en utvecklingsplattform som började utvecklas av Microsoft i slutet av 1990-talet. Flera projekt som var under utveckling inom företaget kom att bli viktiga delar i vad som skulle komma att kallas .NET.

Hjärtat av .NET, och vad som oftast associeras med .NET, är .NET Framework. En beskrivning av .NET kan skilja sig på många punkter, men den viktiga .NET Framework måste alltid finnas med. .NET Framework består av ett antal komponenter som arbetar ovanpå operativsystemet [7][1] och visas i figur 4.1:



Figur 4.1: .NET Framework

- Till skillnad från vad många tror behöver man inte använda operativsystemet Windows för att kunna använda .NET [9]. Specifikationen för .NET Frameworks kärna, där applikationer skrivna i olika högnivåspråk kan exekveras i olika system utan att koden behöver skrivas om, kallas CLI – Common Language Infrastructure, och är en ECMA och ISO standard¹. Microsofts implementation av denna CLI kallas CLR – Common Language Runtime, och är gjord för Windows. Men det

¹ECMA och ISO är båda standardiserings-organisationer

finns även implementationer av CLI för andra operativsystem, till exempel “Mono Development Platform” som är en CLI implementation för (primärt) Linux.

- CLR är .NET Frameworks virtuella maskin och runtime bibliotek. Det abstraherar det underliggande systemet, och exekverar kod från flera olika språk.
- Basklasserna är en mängd klasser som innehåller objekt som är tillgängliga för alla språk i .NET. Här finns vanlig funktionalitet som I/O, grafik, och stränghantering.
- I data- och XMLklasserna finns funktionalitet för databashantering och XML-manipulation².
- Den översta nivån består av klasser för de tre olika typer av applikationer som kan utvecklas i .NET – Web Services, Web Forms, och Windows Forms, vilket kan enkelt beskrivas som webbsidor, program som har uppkoppling mot Internet, och program utan uppkoppling mot Internet.

Tillsammans bildar de tre översta nivåerna (Framework basklasser, Data- och XMLklasser, och applikationstyp-klasserna) den så kallade FCL – Framework Class Library. FCL är det största biblioteket i .NET.

4.2 JavaScript

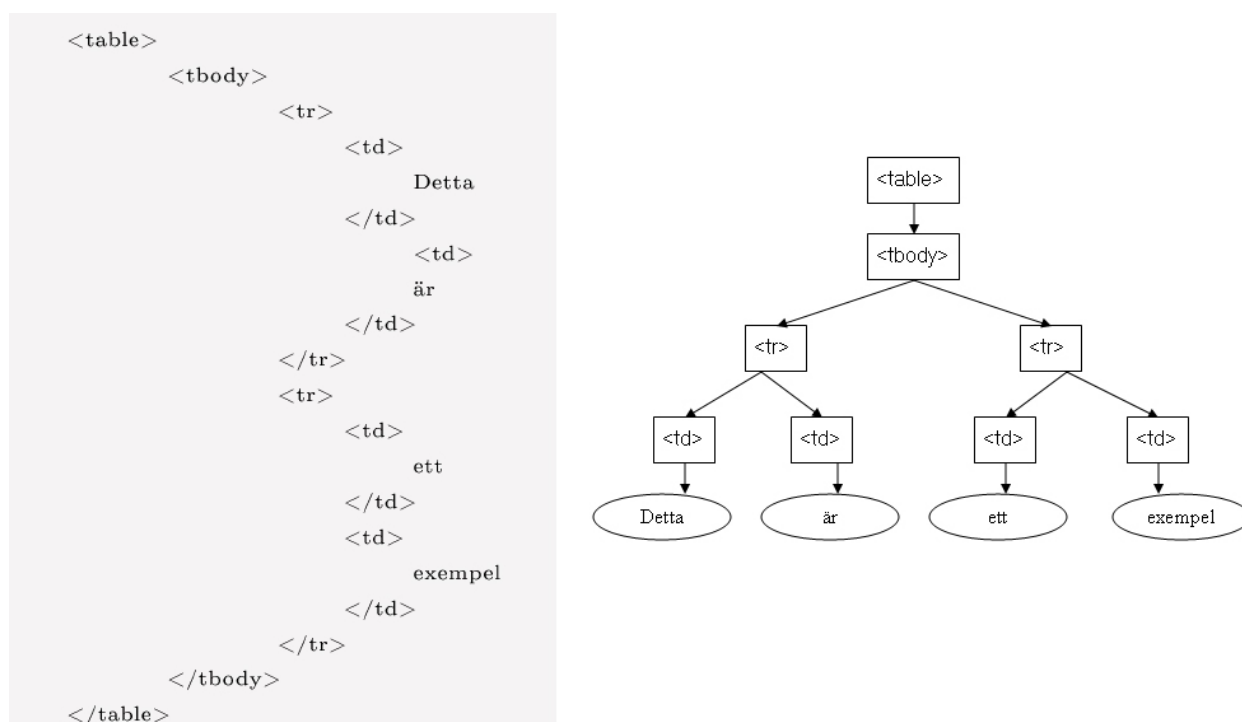
I december 1995 tillkännagav företagen Netscape och Sun att skriptspråket som tidigare kallats LiveScript hädanefter skulle kallas JavaScript [11].

JavaScript var komplementet som saknades för att skapa dynamiska webbsidor. Statisk information visades med HTML, dynamisk information på serversidan kunde fås med CGI – Common Gateway Interface, men det fanns ett behov att kunna utföra operationer dynamiskt på klientsidan. Med CGI krävs ett anrop till servern varje gång ny information

²XML står för eXtensible Markup Language och är ett språk som kan beskriva många olika typer av data

ska genereras, vilket ger en väldig belastning på servern som kanske har tusentals klienter uppkopplade mot sig, medan klienterna knappt utnyttjas alls.

Med JavaScript kom bland annat möjligheten att validera textfält i formulär, visa popup-meddelanden, hantera händelser som musklickning och formulär-submit, och hantera Java applets [4].



Figur 4.2: En enkel tabell och dess DOM

En av de viktigaste aspekterna av JavaScript när det handlar om webbsidor är att kunna dynamiskt ändra i en webbsidas DOM – Document Object Model. Med DOM kan alla delar av ett HTML-dokument nås och editeras i ett trädformat[10]. På detta sätt kan till exempel en tabell fyllas på med nya rader om användaren skulle behöva det. DOM strukturen är standardiserad av W3C³ och består av fyra nivåer, där högre nivå innebär större funktionalitet. Nivå 1 har varit den rekommenderade nivån sedan 1998. I figur 4.2

³W3C är en organisation som utvecklar standardiseringar för internet

visas både en enkel HTML-tabell och en grafisk representation av DOM till denna tabell.

4.3 MySQL

MySQL är ett databassystem som utvecklas av företaget MySQL AB. Företaget grundades 1995 av tre personer som hade behov av ett mer avancerat lättvikts-DBMS (Database Management System) än vad som fanns tillgängligt på den tiden. Innan MySQL grundades hade de använt mSQL (Mini SQL) för att ansluta till sina egna lågnivå-datastrukturer. De upptäckte dock att mSQL inte levde upp till deras krav. Det saknades funktionalitet och det var inte snabbt nog. Detta var upptakten till grundandet av MySQL. Även om MySQL inte var alltför imponerade i startskedet har det stadigt växt under årens lopp, och idag är det ett av världens mest populära DBMS [5].

Anledningen till dess popularitet kan delas in i några punkter:

- Det är öppen källkod under GPL – GNU Public License, vilket betyder att det kan användas gratis i de flesta tillämpningar.
- Det är – enligt MySQL's dokumentation – resurssnålt, vilket betyder att det kan köras på äldre hårdvara.
- Det är plattformsoberoende, vilket betyder att man lätt kan migrera till andra operativsystem.

I MySQL's databassystem är det – som i alla andra databassystem vi känner till – servern som utgör navet. All manipulation av databasen utförs på servern, och klienter kopplar upp sig mot denna för att utföra sina uppgifter. Klienten skickar en fråga till servern skriven i SQL (Standard Query Language), servern utför åtgärderna och skickar tillbaks resultatet till klienten [2]. MySQL är flertrådat, vilket innebär att varje gång en klient kopplar upp sig mot servern skapas en ny tråd för denna uppkoppling. Det är därför flera SQL-frågor kan hanteras samtidigt, vilket ökar prestandan markant.

Kapitel 5

Databasdesign

Vårt tidregistrerings- och faktureringsystem använder sig av en databas för att lagra och hantera data för registrerad tid, kundinformation, och så vidare. För att få en väl fungerande databas krävs en väl genomtänkt design, varför vi utnyttjade några metoder för att få fram en sådan. För det första använde vi E/R-diagram¹ för att rita upp databasen i ett abstrakt och lätthanterligt format. När denna sedan var klar kunde vi enkelt använda en färdig algoritm för att överföra denna design till en implementationsvänlig relationsmodell. Normalformen BCNF – Boyce-Codd normalform, uppnåddes automatiskt genom en välstrukturerad design.

I detta kapitel kommer vi gå igenom de steg vi tog för att få ut den färdiga² databasdesignen. Denna process innebar att skapa ett E/R-diagram, och sedan föra över detta E/R-diagram till en relationsmodell.

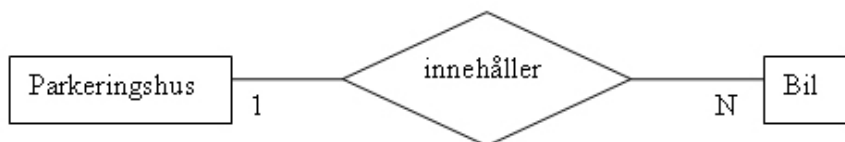
5.1 E/R-diagram

Ett E/R-diagram är ett konceptuellt schema som beskriver en abstraktion av verkligheten. Till exempel skulle man kunna göra en abstraktion av ett parkeringshus

¹E/R – Entity Relationship.

²Färdiga används här som ett relativt begrepp eftersom en databas sannolikt aldrig blir riktigt färdig.

med två entiteter – en som heter parkeringshus och en som heter bil – med en relation mellan dem som definierar att ett parkeringshus kan innehålla många bilar, men en bil kan bara stå i ett parkeringshus (se figur 5.1). E/R-diagram kan alltså användas till många saker, men vi kommer använda det till att abstrahera tidregistrerings- och faktureringsystemet för att få en databas utan redundans och anomalier. Första steget i att göra ett E/R-diagram är att finna de entiteter som ska vara med. Dessa beskrivs i 5.1.1, och i 5.1.2 beskrivs relationerna mellan dessa entiteter.



Figur 5.1: exempel av E/R-diagram

5.1.1 Entiteter

Här beskrivs entiteterna i E/R-diagrammet (se figur 5.2).

Customer – Denna entitet representerar en kund. Primärnyckeln för denna entitet är `customer_nr`.

Price_type – Denna entitet representerar en pristyp. Pristypen anger för ett projekt om kostnaden ska beräknas per timme, dag, vecka, eller om ett fast pris ska sättas upp. Primärnyckeln för denna entitet är `price_type_id`.

agr_payment – Denna entitet representerar en delbetalning som används när pristypen är satt till att vara ett fast pris. Attributet `payment_nr` anger vilken delbetalning i ordningen det är (1 ska betalas först, sedan 2, och så vidare). Primärnyckeln för denna entitet är `agr_payment_id`.

Project – Denna entitet representerar ett projekt. Primärnyckeln för denna entitet är `project_id`.

User – Denna entitet representerar en användare. Primärnyckeln för denna entitet är `user_id`.

Access_level – Denna entitet representerar en rättighetsnivå. Alla användare har en rättighetsnivå som specificerar vilka delar av systemet användaren har tillgång till. Primärnyckeln för denna entitet är `access_level_id`.

Start_page – Denna entitet representerar en startsida, det vill säga den sida användaren kommer till när denne loggat in i systemet. Primärnyckeln för denna entitet är `start_page_id`.

Project_phase – Denna entitet representerar en projektfas. En projektfas är ett delmoment i ett projekt som till exempel konfigurering och testning. Primärnyckeln för denna entitet är `project_phase_id`.

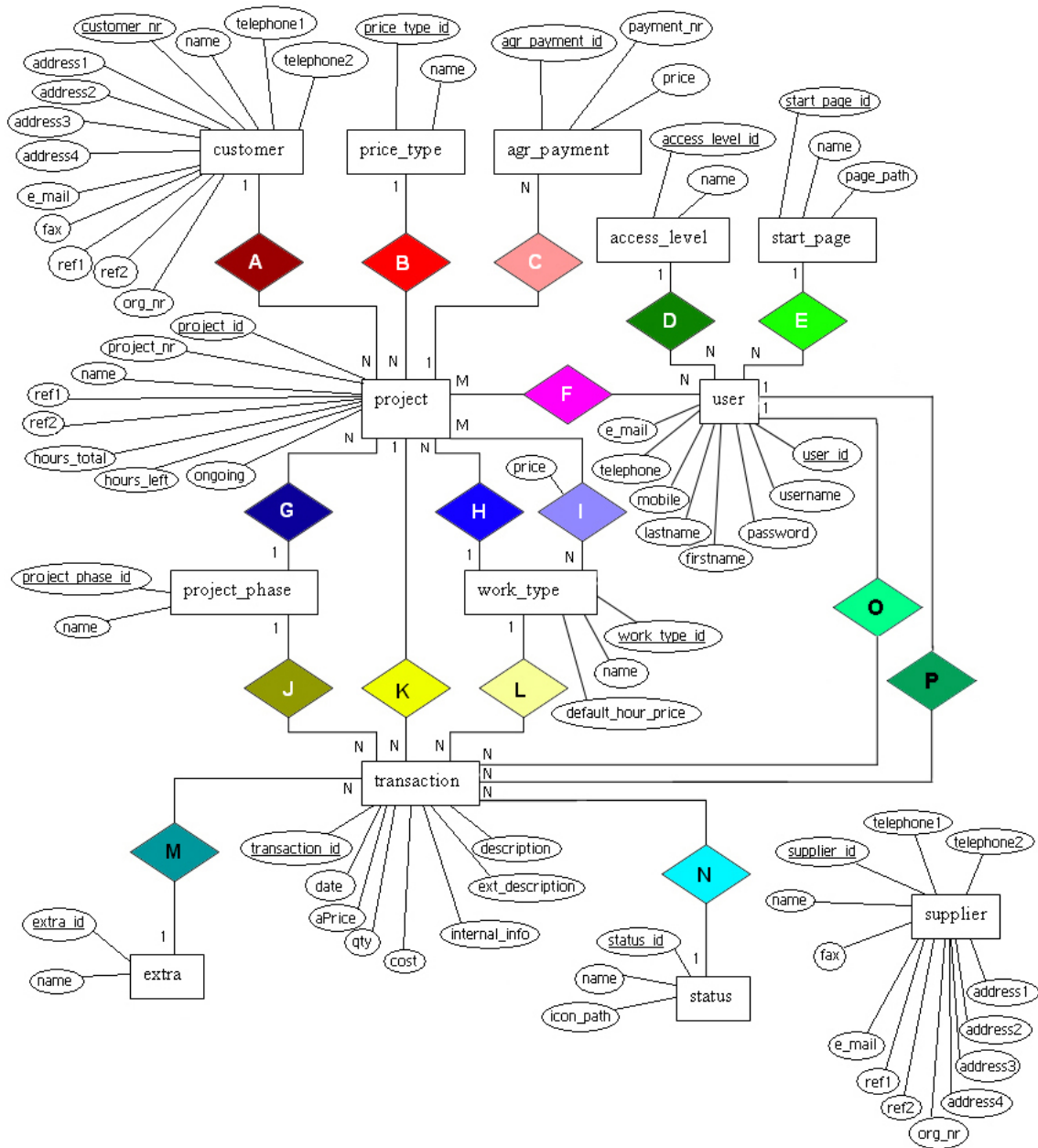
Work_type – Denna entitet representerar en arbetstyp, som används när transaktioner ska registreras för att definera vilken typ av arbete som utförts. Primärnyckeln för denna entitet är `work_type_id`.

Transaction – Denna entitet representerar en transaktion, med andra ord ett arbete utfört av en användare under en viss tid. Primärnyckeln för denna entitet är `transaction_id`.

Extra – Denna entitet representerar en extra uppgift hos en transaktion. En extra uppgift kan till exempel ange om transaktionen den är knuten till ska skrivas med på fakturan. Primärnyckeln för denna entitet är `extra_id`.

Status – Denna entitet representerar en status, eller tillstånd hos en transaktion. Primärnyckeln för denna entitet är `status_id`.

Supplier – Denna entitet representerar en leverantör. Primärnyckeln för denna entitet är `supplier_id`.



Figur 5.2: E/R-diagram

5.1.2 Relationer

Här beskrivs vad alla relationer i E/R-diagrammet innebär. Notera att entiteten “supplier” inte har någon relation till någon annan entitet. Detta har sin förklaring i att den endast används till att lagra information om leverantören som utfört arbetet, vilket ska skrivas med vid fakturering.

- A** – En kund kan ha flera projekt, men ett projekt kan bara ha en kund.
- B** – En pristyp kan tillhöra flera projekt, men ett projekt kan bara ha en pristyp.
- C** – En delbetalning kan bara tillhöra ett projekt, men ett projekt kan ha flera delbetalningar.
- D** – En rättighetsnivå kan användas av flera användare, men en användare kan bara ha en rättighetsnivå.
- E** – En startsida kan användas av flera användare, men en användare kan bara ha en startsida.
- F** – En användare kan vara uppskrivem på flera projekt, och ett projekt kan ha flera användare uppskrivna på sig.
- G** – En projektfas kan stå som standardvärde i flera projekt, men ett projekt kan bara ha en projektfas som standardvärde.
- H** – En arbetstyp kan stå som standardvärde till flera projekt, men ett projekt kan bara ha en arbetstyp som standardvärde.
- I** – En arbetstyp kan användas i flera projekt med projektspecifikt pris för varje, och ett projekt kan ha flera arbetstyper med projektspecifikt pris för varje.
- J** – En projektfas kan användas i flera transaktioner, men en transaktion kan bara vara utförd i en projektfas.

- K** – Ett projekt kan användas i flera transaktioner, men en transaktion kan bara vara utförd i ett projekt.
- L** – En arbetstyp kan användas i flera transaktioner, men en transaktion kan bara utföra en arbetstyp.
- M** – En extrauppgift kan användas i flera transaktioner, men en transaktion kan bara ha en extrauppgift.
- N** – En status kan innehas av flera transaktioner, men en transaktion kan bara ha en status.
- O** – En användare kan ha flera transaktioner, men en transaktion kan bara ägas av en användare.
- P** – En användare kan stå som referens på flera transaktioner, men en transaktion kan bara ha en användare som referens.

5.2 Mappning till relationsmodellen

För att mappa E/R-diagrammet till relationsmodellen använde vi oss av en algoritm[3]. Här följer en kort beskrivning av algoritmen (se [3] för en grundligare förklaring, och beskrivning av termer):

Steg 1 - Mappning av starka entiteter

Varje stark entitet blir en basrelation där primärnyckeln blir nyckelattributet/nyckelattributen i entiteten.

Steg 2 - Mappning av svaga entiteter

Efersom vi inte har någon svag entitet i vårt E/R-diagram hoppade vi över detta steg. Varje svag entitet blir en basrelation som får alla enkla attribut som den svaga

entiteten har, och primärnyckeln blir kombinationen av primärnyckeln hos ägandeentiteten och (om det finns) den partiella nyckeln hos den svaga entiteten. Om ägandeentiteten också är en svag entitet, ska denna mappas först för att få ut dess primärnyckel. En svag entitet kan inte existera utan en relaterad ägandeentitet, så ägandeentiteten får aldrig tas bort ur databasen om den svaga entiteten finns kvar. Detta kan implementeras med ON DELETE CASCADE och ON UPDATE CASCADE.

Steg 3 - Mappning av 1 till 1 relationer

Det finns två tillvägagångssätt att mappa 1 till 1 relationer:

- Välj en av basrelationerna och låt primärnyckeln i den andra relationen bli en främmandenyckel i denna. Flytta även över (om det skulle finnas) alla enkla attribut i 1 till 1 relationen (alltså attribut som på E/R-diagrammet är kopplade till diamanten/relationen) till denna basrelation.
- Sätt ihop de två entiteterna till en och samma entitet och mappa därefter.

Steg 4 - Mappning av 1 till N relationer

Lägg till som främmandenyckel i basrelationen på N-sidan, primärnyckeln hos den andra basrelationen.

Steg 5 - Mappning av N till M relationer

Gör varje N till M relation till en basrelation. Sätt primärnyckeln i varje sådan basrelation till kombinationen av primärnycklarna i de basrelationer denna relation kopplar samman. Alternativt kan primärnyckeln vara ett nytt attribut men primärnycklarna från de sammankopplade basrelationerna måste finnas med som främmandenycklar.

Steg 6 - Mappning av attribut

Alla egenskaper hos entiteter blir attribut i den relation de tillhör. Skillnaden för

mångvärdes-/flervärdesattribut är att man för varje flervärdesattribut skapar en ny basrelation som innehåller ett attribut som är flervärdesattributet och en främmandenyckel som är primärnyckeln hos den relation som hade flervärdesattributet. Primärnyckeln i denna basrelation blir kombinationen av dessa två attribut.

Efter mappningen befinner sig databasen i en viss normalform. Normalisering används för att minimera redundans och avvikande beteenden vid insättning, uppdatering och borttagning av data. Högre normalform är vanligtvis bättre eftersom risken för anomalier och redundans minskar. I vissa fall kan dock prestandan sjunka med högre normalform, och det kan vara motiverat att välja en lägre.

- Första normalformen (1NF) innebär att alla attribut är atomära, det vill säga att ett attribut i en tuple ej får ha flera värden än ett. Tabellerna i vår databas uppfyller detta krav.
- Andra normalformen (2NF) innebär att relationsschemat är i första normalform och att alla icke-nyckel attribut är fullständigt funktionellt beroende av primärnyckeln. Detta innebär att ett icke-nyckel attribut inte får vara delvis beroende av en kandidatnyckel. Ett relationsschema där alla tabeller endast har enkla primärnycklar är alltid minst i andra normalformen, medan ett relationsschema där någon tabell har en sammansatt primärnyckel måste kontrolleras. Tabellerna i vår databas uppfyller kraven för andra normalformen.
- Tredje normalformen (3NF) innebär att relationsschemat är i andra normalformen och varje attribut som inte ingår i primärnyckeln är icke-transitivt beroende av primärnyckeln. Med andra ord får det i en tabell inte finnas icke-nyckel attribut som är beroende av ett annat icke-nyckel attribut. Tabellerna i vår databas uppfyller kraven för tredje normalformen.

- Boyce-Codd normalform (BCNF) innebär att relationsschemat är i tredje normalformen och varje determinant är en kandidatnyckel. Detta innebär alltså att det varken får finnas något nyckelattribut eller ickenyckel attribut som är beroende av ett icke-nyckel atribut. Tabellerna i vår databas uppfyller kraven för BCNF.

5.3 De färdiga tabellerna

Här följer en listning av alla tabeller i den färdiga databasen, vad de olika attributen är för något och vad de har för datatyper. Datatyperna som används finns mer utförligt beskrivna i [5].

Tabell 5.1: customer

customer		
customer_nr	Beskrivning	Primärnyckel, ett värde i 40000-serien.
	Datatyp	smallint(5) unsigned
	Obligatorisk	Ja
name	Beskrivning	Kundens namn
	Datatyp	varchar(40)
	Obligatorisk	Ja
address1	Beskrivning	Första adressraden
	Datatyp	varchar(30)
	Obligatorisk	Ja
address2	Beskrivning	Andra adressraden
	Datatyp	varchar(30)
	Obligatorisk	Ja
address3	Beskrivning	Tredje adressraden
	Datatyp	varchar(30)
	Obligatorisk	Ja
address4	Beskrivning	Fjärde adressraden
	Datatyp	varchar(30)
	Obligatorisk	Nej
telephone1	Beskrivning	Första telefonnumret
	Datatyp	varchar(20)
	Obligatorisk	Nej
telephone2	Beskrivning	Andra telefonnumret
	Datatyp	varchar(20)
	Obligatorisk	Nej
fax	Beskrivning	Faxnummer
	Datatyp	varchar(20)
	Obligatorisk	Nej
e_mail	Beskrivning	E-post-adress
	Datatyp	varchar(40)
	Obligatorisk	Nej
ref1	Beskrivning	Första referensen
	Datatyp	varchar(30)
	Obligatorisk	Nej
ref2	Beskrivning	Andra referensen
	Datatyp	varchar(30)
	Obligatorisk	Nej
org_nr	Beskrivning	Organisationsnummer
	Datatyp	varchar(11)
	Obligatorisk	Ja

Tabell 5.2: price_type

price_type		
price_type_id	Beskrivning	Primärnyckel
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
name	Beskrivning	Pristypens namn
	Datatyp	varchar(20)
	Obligatorisk	Ja

Tabell 5.3: agr_payment

agr_payment		
agr_payment_id	Beskrivning	Primärnyckel
	Datatyp	int(11) unsigned
	Obligatorisk	Ja
payment_nr	Beskrivning	Anger vilken delbetalning det är
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
price	Beskrivning	Kostnaden för denna delbetalning
	Datatyp	double(10,2) unsigned
	Obligatorisk	Ja
project_id	Beskrivning	Främmandenyckel till project_id i project
	Datatyp	smallint(5) unsigned
	Obligatorisk	Ja

Tabell 5.4: project

project		
project_id	Beskrivning	Primärnyckel
	Datatyp	smallint(5) unsigned
	Obligatorisk	Ja
project_nr	Beskrivning	Projektnummer på formen G4001100
	Datatyp	varchar(8)
	Obligatorisk	Ja
name	Beskrivning	Projektets namn
	Datatyp	varchar(40)
	Obligatorisk	Ja
refl	Beskrivning	Första referensen
	Datatyp	varchar(30)
	Obligatorisk	Nej
telephone1	Beskrivning	Andra referensen
	Datatyp	varchar(30)
	Obligatorisk	Nej
hours_total	Beskrivning	Totala antalet upparbetade timmar i projektet
	Datatyp	smallint(5) unsigned
	Obligatorisk	Nej
hours_left	Beskrivning	Totala antalet timmar kvar i projektet
	Datatyp	smallint(5)
	Obligatorisk	Nej
ongoing	Beskrivning	Anger om projektet är igång eller om det är avslutat
	Datatyp	tinyint(1) unsigned
	Obligatorisk	Ja
customer_nr	Beskrivning	Främmandenyckel till customer_nr i customer
	Datatyp	smallint(5) unsigned
	Obligatorisk	Ja
price_type_id	Beskrivning	Främmandenyckel till price_type_id i price_type
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
project_phase_id	Beskrivning	Främmandenyckel till project_phase_id i project_phase
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
work_type_id	Beskrivning	Främmandenyckel till work_type_id i work_type
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja

Tabell 5.5: user

user		
user_id	Beskrivning	Primärnyckel
	Datatyp	mediumint(8) unsigned
	Obligatorisk	Ja
username	Beskrivning	Inloggningsnamn
	Datatyp	varchar(10)
	Obligatorisk	Ja
password	Beskrivning	Lösenord för inloggning
	Datatyp	varchar(20)
	Obligatorisk	Ja
firstname	Beskrivning	Förnamn
	Datatyp	varchar(20)
	Obligatorisk	Ja
lastname	Beskrivning	Efternamn
	Datatyp	varchar(20)
	Obligatorisk	Ja
mobile	Beskrivning	Mobilnummer
	Datatyp	varchar(20)
	Obligatorisk	Nej
telephone	Beskrivning	Telefonnummer
	Datatyp	varchar(20)
	Obligatorisk	Nej
e_mail	Beskrivning	E-post-adress
	Datatyp	varchar(40)
	Obligatorisk	Nej
access_level_id	Beskrivning	Främmandenyckel till access_level_id i access_level
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
start_page_id	Beskrivning	Främmandenyckel till start_page_id i start_page
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja

Tabell 5.6: access_level

access_level		
access_level_id	Beskrivning	Primärnyckel
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
name	Beskrivning	Namnet på accessnivån
	Datatyp	varchar(20)
	Obligatorisk	Ja

Tabell 5.7: start_page

start_page		
start_page_id	Beskrivning	Primärnyckel
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
name	Beskrivning	Namnet på sidan
	Datatyp	varchar(40)
	Obligatorisk	Ja
page_path	Beskrivning	Sökvägen till sidan
	Datatyp	varchar(40)
	Obligatorisk	Ja

Tabell 5.8: project_phase

project_phase		
project_phase_id	Beskrivning	Primärnyckel
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
name	Beskrivning	Namnet på projektfasen
	Datatyp	varchar(20)
	Obligatorisk	Ja

Tabell 5.9: work_type

work_type		
work_type_id	Beskrivning	Primärnyckel
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
name	Beskrivning	Namnet på arbetstypen
	Datatyp	varchar(20)
	Obligatorisk	Ja
default_hour_price	Beskrivning	Standard timpris
	Datatyp	double(7,2) unsigned
	Obligatorisk	Nej

Tabell 5.10: transaction

transaction		
transaction_id	Beskrivning	Primärnyckel
	Datatyp	int(10) unsigned
	Obligatorisk	Ja
date	Beskrivning	Datum när transaktionen utfördes
	Datatyp	date
	Obligatorisk	Ja
aPrice	Beskrivning	Kostnad per enhet
	Datatyp	Float(7,2) unsigned
	Obligatorisk	Ja
qty	Beskrivning	Mängden timmar/veckor/dagar/km
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
cost	Beskrivning	Kostnaden för transaktionen
	Datatyp	Double(10,2) unsigned
	Obligatorisk	Ja
description	Beskrivning	Beskrivande text på fakturan
	Datatyp	varchar(110)
	Obligatorisk	Nej
ext_description	Beskrivning	Mer utförlig text på fakturan
	Datatyp	varchar(110)
	Obligatorisk	Nej
internal_info	Beskrivning	Intern information
	Datatyp	text
	Obligatorisk	Nej
project_id	Beskrivning	Främmandenyckel till project_id i project
	Datatyp	smallint(5) unsigned
	Obligatorisk	Ja
project_phase_id	Beskrivning	Främmandenyckel till project_phase_id i project_phase
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
work_type_id	Beskrivning	Främmandenyckel till work_type_id i work_type
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
extra_id	Beskrivning	Främmandenyckel till extra_id i extra
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
status_id	Beskrivning	Främmandenyckel till status_id i status
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
user_id	Beskrivning	Främmandenyckel till user_id i user
	Datatyp	mediumint(8) unsigned
	Obligatorisk	Ja
invoice_user_id	Beskrivning	Främmandenyckel till user_id i user (står för referens)
	Datatyp	mediumint(8) unsigned
	Obligatorisk	Nej

Tabell 5.11: extra

extra		
extra_id	Beskrivning	Primärnyckel
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
name	Beskrivning	Namnet på extrauppgiften
	Datatyp	varchar(20)
	Obligatorisk	Ja

Tabell 5.12: status

status		
status_id	Beskrivning	Primärnyckel
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
name	Beskrivning	Namnet på statusen
	Datatyp	varchar(60)
	Obligatorisk	Ja
icon_path	Beskrivning	Sökvägen till ikonen
	Datatyp	varchar(30)
	Obligatorisk	Ja

Tabell 5.13: supplier

supplier		
supplier_id	Beskrivning	Primärnyckel
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
name	Beskrivning	Leverantörens namn
	Datatyp	varchar(40)
	Obligatorisk	Ja
address1	Beskrivning	Första adressraden
	Datatyp	varchar(30)
	Obligatorisk	Ja
address2	Beskrivning	Andra adressraden
	Datatyp	varchar(30)
	Obligatorisk	Ja
address3	Beskrivning	Tredje adressraden
	Datatyp	varchar(30)
	Obligatorisk	Ja
address4	Beskrivning	Fjärde adressraden
	Datatyp	varchar(30)
	Obligatorisk	Nej
telephone1	Beskrivning	Första telefonnumret
	Datatyp	varchar(20)
	Obligatorisk	Nej
telephone2	Beskrivning	Andra telefonnumret
	Datatyp	varchar(20)
	Obligatorisk	Nej
fax	Beskrivning	Faxnummer
	Datatyp	varchar(20)
	Obligatorisk	Nej
e_mail	Beskrivning	E-post-adress
	Datatyp	varchar(40)
	Obligatorisk	Nej
ref1	Beskrivning	Första referensen
	Datatyp	varchar(30)
	Obligatorisk	Nej
ref2	Beskrivning	Andra referensen
	Datatyp	varchar(30)
	Obligatorisk	Nej
org_nr	Beskrivning	Organisationsnummer, ex: 445566-1234
	Datatyp	varchar(11)
	Obligatorisk	Ja

Tabell 5.14: cost_for_worktype

cost_for_worktype		
project_id	Beskrivning	Del av primärnyckel och främmandenyckel till project_id i project
	Datatyp	smallint(5) unsigned
	Obligatorisk	Ja
work_type_id	Beskrivning	Del av primärnyckel och främmandenyckel till work_type_id i work_type
	Datatyp	tinyint(3) unsigned
	Obligatorisk	Ja
price	Beskrivning	Kostnaden för arbetstypen i projektet
	Datatyp	double(7,2) unsigned
	Obligatorisk	Ja

Tabell 5.15: signed_up

signed_up		
user_id	Beskrivning	Del av primärnyckel och främmandenyckel till user_id i user
	Datatyp	mediumint(8) unsigned
	Obligatorisk	Ja
project_id	Beskrivning	Del av primärnyckel och främmandenyckel till project_id i project
	Datatyp	smallint(5) unsigned
	Obligatorisk	Ja

5.4 Kapitelsammanfattning

Vi har skapat en databas som befinner sig i BCNF. Den består av femton tabeller, där alla utom en är sammankopplade med varandra. Fler egenskaper och tabeller skulle kunna tänkas vara med, som till exempel en tabell med två status symboler som kunde användas till projektets status. Nu används istället endast ett attribut i tabellen som heter "ongoing", vilket är antingen 0 eller 1. Vilken ikon som sedan ska visas i användargränssnittet där projektstatus visas, är nu hårdkodat in och beror på denna siffra. Vår databas uppfyller dock kraven som ställdes på systemet av den avgränsade kravspecifikationen, och även den kompletta specifikationen.

Databasen är konstruerad för att kunna användas av andra program. Tack vare detta skulle databasen kunna återanvändas i framtiden om Xeratech bestämmer sig för att göra om systemet på nytt.

Kapitel 6

Design och implementation

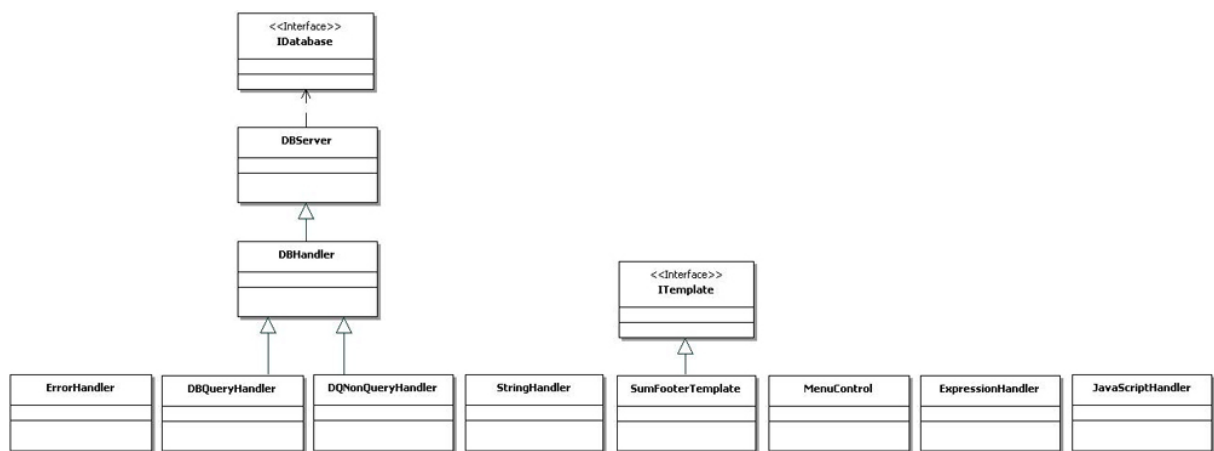
I detta kapitel går vi igenom tillvägagångssättet för att implementera delarna från den avgränsade kravspecifikationen. Vi redogör för hur det färdiga systemet blev, hur det ser ut, och hur väl det fungerar. Här står även på vilka punkter vår implementation skiljer sig från specifikationen och varför den gör det.

6.1 Implementation

Utvecklingen av systemet skedde genom att vi först tillsammans med vår handledare på universitetet och vår handledare på Xeratech valde ut de delar som var viktigast att göra klart. Dessa delar bildade den avgränsade specifikationen, som kan läsas i kapitel 3.

Därefter implementerade vi de delar som var viktiga för att kunna lägga till den information som behövs för att lägga till tider, med andra ord allt i admininställningar och projekt. Till sist implementerades tidregistreringsdelen. Trots denna avgränsning av arbetet tvingades vi prioritera bort vissa delar senare under utvecklingen för att hinna klart med de väsentligaste bitarna.

Koden för server-logiken är skriven i C# och innehåller en mängd klasser. I figur 6.1 visas ett klassdiagram över dessa och hur de förhåller sig till varandra.



Figur 6.1: Klassdiagram över systemet

Här följer förklaringar av de olika klasserna:

- ErrorHandler tar hand om fel som uppstår på sidor genom att förflytta användaren till en annan sida och skriva ut ett felmeddelande.
- DBQueryHandler och DBNonQueryHandler beskrivs i 6.1.1, och MenuControl beskrivs i 6.1.3.
- StringHandler hanterar alla strängar som används på flera ställen i systemet som till exempel sökvägar till bilder.
- SumFooterTemplate hanterar den nedersta raden på transaktionsöversikt-sidan där alla summor räknas ihop.
- ExpressionHandler hanterar reguljära uttryck som används för att kontrollera fälten där användaren skriver in information för att uppdatera och lägga till poster (till exempel på “Lägg till användare”-sidan och “Editera projekt”-sidan).
- JavaScriptHandler hanterar javascripten som laddas in dynamiskt på flera sidor, som till exempel ett script för att visa ett popup-meddelande.

Förutom dessa klasser har även varje webbsida en egen klass som hanterar alla funktioner på den sidan, som bland annat att ladda in data i en tabell.

I 6.1.1 till 6.1.5 beskrivs de fem huvuddelarna i systemet, hur vi gick tillväga för att utveckla dessa, och varför vi gjorde de val vi gjorde.

6.1.1 Databasen

Det första steget i implementationen var att sätta upp databassystemet. Alla tabeller i relationsmodellen från kapitel 5.3 lades in och alla främmandenycklar och index lades till. För att kunna använda databasen inifrån systemet måste först en uppkoppling skapas mot databasen. Uppkoppling och anrop till MySQL databasen sker via MySQL Connector/Net. Klassen IDatabase är ett interface för anrop till databasen, och implementeras av DBServer. DBHandler ärver från DBServer och sätter databasinställningarna för den databas som används i systemet. Två klasser ärver från DBHandler – DBQueryHandler och DBNonQueryHandler. DBQueryHandler hanterar sql-frågor till servern som hämtar information från databasen, och DBNonQueryHandler hanterar sql-frågor som ändrar informationen i databasen.

```
DBQueryHandler handler = new DBQueryHandler();
MySqlDataReader myReader = handler.getAllUsers();
users.DataSource = myReader;
users.DataBind();
myReader.Close();
handler.Close()
```

Figur 6.2: Exempel på hur databindningen går till

I figur 6.2 visas ett exempel på hur inhämtningen av data till en tabell går till. DBQueryHandler används för att skicka en fråga till databasen, och svaret tas emot av en läsare. Tabellen som ska visa informationen heter users, och källan för varifrån den ska hämta data sätts till en läsare. Datan binds till tabellen, läsaren och uppkopplingen stängs ner. Att stänga både läsaren och uppkopplingen är viktigt för annars avallokerar

inte mysql resurserna förrän uppkopplingen “timar ut”, det vill säga att en förutbestämd väntetid tar slut. Vad som ska visas från den data som hämtas in och var den ska visas hanteras med ASP.NET i webbsidan.

I figur 6.3 visas ett exempel på hur ny data läggs till i databasen. Här används DBNonQueryHandler för hantering av sql-frågorna istället för DBQueryHandler, eftersom inget resultat ska returneras utan det handlar om en uppdatering av databasen. Data från de fält användaren fyllt i hämtas in och används som inparametrar till funktionen som uppdaterar databasen.

```
string pass_word = password.Text;
string user_name = username.Text;
string first_name = firstName.Text;
string last_name = lastName.Text;
string mobile_nr = mobile.Text;
string tel_nr = phone.Text;
string e_mail = e_post.Text;
string access = access_rights.SelectedItem.ToString();
string page = start_page.SelectedItem.ToString();
string accNr = access_rights.Text;
string startID = start_page.Text;

DBNonQueryHandler handler = new DBNonQueryHandler();
handler.createNewUser(user_name, pass_word, first_name, last_name, mobile_nr, tel_nr, e_mail, accNr, startID);
```

Figur 6.3: Exempel på hur uppdatering av data sker

6.1.2 Inloggning

För att komma åt systemet krävs att användaren loggar in med användarnamn och lösenord. Det var en önskan av konsulterna på Xeratech att användarnamnet på inloggningssidan skulle ligga i en rullista, vilket är anledningen till att sidan även blev implementerad på det sättet istället för med en vanlig textbox. Både användarnamn och lösenord ligger sparade i user-tabellen i databasen. Det finns ingen kryptering på lösenordet eftersom säkerhetsnivån inte ansågs behöva ha en sådan hög nivå.

För att slippa välja sitt användarnamn i rullistan varje gång man loggar in används

cookies för att spara användarnamnet. En HTTP-cookie är ett paket med information i textformat som skickas från en webserver till en webbläsare, varpå webbläsaren lagrar cookien i en textfil. Denna cookie skickas sedan tillbaka till servern varje gång webbläsaren försöker komma åt en sida på servern. En cookie som innehåller det senast använda användarnamnet lagras på användarens dator. När denna användare går in på inloggningssidan skickas denna cookie till servern som sätter användarnamnet i rullistan till det som specificeras av cookien.

Vi använder även en session-cookie för att hantera information om vilken användare som är inloggad för tillfället. En session-cookie är en cookie som finns kvar tills webbläsaren stängs ner. Cookien tas även bort om användaren väljer att logga ut. I session-cookien lagras bland annat information om vilka rättigheter användaren har.

6.1.3 Menyn

Menyn är den del av systemet som kopplar samman alla sidor för användaren och låter denne manövrera sig genom systemet. Eftersom menyn är en viktig del av systemet behöver den både vara användarvänlig och ha bra prestanda, och därför tog det lång tid att designa den till en acceptabel nivå. I ett första skede tänkte vi använda den menykontroll som redan finns färdig i .NET, men denna gav inte en tillräckligt klar överblick av var i systemet man för tillfället befinner sig. I nästa försök användes JavaScript för att dynamiskt skapa nya undermenyer till en enkel första rad bestående av html-komponenten "div". Detta ansågs bli onödigt komplicerat utan att denna komplexitet var motiverad av ökad användarvänlighet. Den slutgiltiga designen blev en meny bestående endast av html-komponenter som genereras på serversidan med hjälp av C#-kod. Menyn implementerades som en .NET User Control, vilket är en webbkontroll i .NET som kan inkluderas på vilken ASP.NET-sida som helst. Detta medför att det räcker med några rader kod på webbsidan för att inkludera menyn, en rad som instansierar den, och tre rader i C#-koden som sätter vad som ska finnas på varje nivå av menyn.

I utvecklingen av systemet har vi eftersträvat hög prestanda och därför försökt undvika att använda för mycket bilder¹. Menyn använder inte en enda bild, utan består endast av html-komponenter och stilmallar i CSS².

6.1.4 Admininställningar och projekt

Sidorna för inställningar av användare, leverantörer, arbetstyper, projektfaser, kunder och projekt består alla av sidor för att visa informationen i tabeller, editera informationen, lägga till ny information, och ta bort information.

När en post i en tabell som till exempel tabellen över alla användare ska editeras, klickar användaren på en knapp på den rad i tabellen som posten ligger. Primärnyckeln för posten som ska ändras skickas som formulärdata till en sida där användaren kan editera postens information och trycka på en knapp för att skicka uppdateringarna till databasen. Problemet vi stötte på var att i .NET antas det att alla formulär ska skicka sin formulärdata till samma sida den skickas från. Trots att vi skrivit in vilken sida informationen skulle skickas till ändrades detta när sidorna kompileras. I .NET 2.0 finns tre lösningar för detta: Response.Redirect, Server.Transfer, och Cross Page Posting. Alla dessa har dock nackdelar:

- Med Response.Redirect måste all formulärdata skickas som en textsträng i URL:en.
- Med Server.Transfer ändras inte URL:en när användaren förflyttas till den nya sidan.
- Med Cross Page Posting laddas sändarsidan om en gång till, trots att användaren är på en ny sida.

¹Bilder har dock används vid de tillfällen det ansetts tillföra något till användarvänligheten, som till exempel för att visa vilken status en transaktion har.

²CSS är ett språk som används för att beskriva presentationen av ett annat språk (som till exempel HTML)

Vi undersökte om det fanns något enkelt sätt att med JavaScript lösa våra problem med att skicka formulärdata till en annan sida, och det fanns det. Genom att köra ett JavaScript när användaren klickar på en knapp för att skicka formulärdata till en annan sida, kan vi ändra i formulärets egenskaper och sätta vilken sida som ska anropas. Med JavaScript kan vi även samtidigt skapa ett gömt fält som innehåller primärnyckeln till den post som ska editeras, och skicka med detta fält samtidigt.

Förutom att ändra vilken sida formulärdata ska skickas till måste även .NET-fältet “__VIEWSTATE” byta namn. Detta fält innehåller specifik valideringsinformation till denna sida, och om den skickas med i formuläret kommer den mottagande sidan att kasta ett undantag.

För att detta ska fungera krävs dock att JavaScript är påslaget i den webbläsare som används. Men vi vet redan att det finns andra delar av systemet som kräver JavaScript, därför var detta inget problem, eftersom JavaScript redan är ett krav för användandet av systemet.

När en post i en tabell som till exempel tabellen över alla användare ska tas bort klickar användaren på en knapp på den rad i tabellen som posten ligger. Primärnyckeln för posten som ska tas bort skickas som formulärdata till samma sida som formuläret ligger på. Eftersom formulärdatan ska skickas till originalsidan behöver vi inte använda samma teknik som för editsidorna. JavaScript används dock fortfarande för att skapa ett gömt fält att lagra primärnyckeln till posten. JavaScript används också till att visa ett popup meddelande där användaren frågas om posten verkligen ska tas bort eller inte.

När information ska läggas till, till exempel om en ny användare ska skapas, går användaren in på en “lägg till”-sida, fyller i relevant information och klickar på OK.

6.1.5 Tidregistrering

MySQL-frågan till översiktssidan för tidregistrering blev den längsta och mest avancerade av alla våra MySQL-frågor. Anledningen till detta var att den kräver uträkningar med

flera tabeller involverade, och längst ner i sista raden av tabellen ska vissa kolumner summeras på ett korrekt sätt.

En del av översiktssidan för tidregistrering var att det skulle gå att söka bland transaktionernas olika fält. Ovanför varje kolumn i tabellen skulle det gå att specificera värden för sökning. Vi insåg i slutet av utvecklingen av systemet att vi inte skulle hinna implementera detta, varför det inte finns med för tillfället utan lämnas till att implementeras av Xeratech senare. Denna sökfunktion skulle även finnas med på andra sidor, men finns inte med där heller av samma skäl.

För att användaren ska kunna lägga till transaktioner för ett projekt måste denne vara uppskriven på det. Därför implementerade vi en liten del av “Mina egenskaper”-sidan. En sida där användaren kan se vilka projekt denne är uppskriven på, och en sida där användaren kan registrera sig på projekt. Detta var en viktig del att hinna implementera eftersom användare i annat fall skulle vara tvungna att editera data i databasen manuellt för att bli uppskrivna på projekt.

“Lägg till transaktion”-sidan är den mest avancerade sidan vad JavaScript beträffar. Nya transaktioner läggs till dynamiskt på sidan genom att klicka på en knapp. Detta görs med hjälp av JavaScript som manipulerar sidans DOM (se kapitel 4.2). Eftersom nya transaktioner kopieras upp med JavaScript måste kontrollen av textboxarna ske med JavaScript också, vilket skiljer sig från andra sidor där vi kunna göra allt med ASP.NET och C#. Vi hann dock inte att implementera någon felhantering av denna sort (detta gäller även för “Lägg till projekt”-sidan där JavaScript används för pristyper).

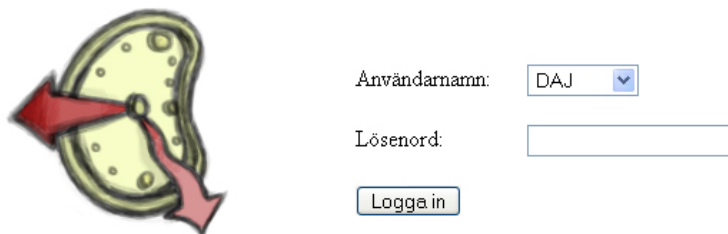
6.2 Användargränssnitt

Det slutgiltiga användargränssnittet skiljer sig i utseende från bilderna i kravspecifikationen. Bilderna i kravspecifikationen är endast tänkta som en mall eller skiss för hur det ska se ut. CSS används för att definiera utseendet på alla sidor, vilket gör

att utseendet på alla sidor kan ändras genom att göra uppdateringar i en fil. Färgerna på sidorna valdes för att likna färgskalan på Xeratechs hemsida, och designen optimerades för att få en tydlig översikt och plats för mycket information utan att det ser rörigt ut. De enstaka bilder som finns på sidorna är ritade med en Wacom Graphire penna i Adobe Photoshop. Detta medförde att man kunde rita bilder av en kvalitet som hade varit mycket svårt utan den digitala pennan.

6.2.1 Inloggning

Inloggningsidan (figur 6.4) är nästan identiskt med den i kravspecifikationen med den skillnaden att bilden till vänster om formulärfälten ser annorlunda ut och har en popup-text när man håller pekaren över den som lyder “Välkommen” (bilderna i specifikationen ska som tidigare nämnts endast ses som en mall eller en skiss för hur det slutgiltigen ska se ut). Både bilden i kravspecifikationen och den på den slutgiltiga sidan föreställer dock en böjd klocka.



Figur 6.4: Den färdiga loginsidan

6.2.2 Användaröversikt

Alla sidor för att visa information i tabeller ser ungefär likadana ut, och vi har valt att visa användaröversikt-sidan för att exemplifiera dessa (se figur 6.5). I den översta raden i tabellen står kolumnnamnen som alla är klickbara för att sortera tabellen efter den

kolumnen. Raderna i tabellen alternerar bakgrundsfärg för att förtydliga vilken rad värdena ligger på.

Längst upp i bilden ser vi menyn med tre nivåer som tydligt markerar på vilken sida användaren befinner sig. De svarta linjer som markerar kanterna av varje nivå fanns från början även mellan varje rad. I Internet Explorer visades dock inte dessa linjer på korrekt sätt, så vi valde att hoppa över dessa.

The image shows a navigation menu at the top with items: Tidrapportering, Attestering, Fakturering, Projekt, Admininställningar, Mina egenskaper, Om, and Logga ut. Below it are sub-menus: Användare, Kunder, Systeminställningar, Visa användare, and Lägg till. Below the menu is a table with the following data:

Användarnamn	Förnamn	Efternamn	Rättigheter	Mobil	Telefon	E-post		
XeraTech	Admin	Adminsson	Admin				€	✗
PEJ	Pelle	Jansson	Konsult				€	✗
KAP	Karl	Persson	Projektledare				€	✗

Figur 6.5: Den färdiga “Användaröversikt”-sidan

6.2.3 Lägg till användare

De flesta sidor för att lägga till eller editera information ser ungefär likadana ut, och vi har valt att visa “Lägg till användare”-sidan för att exemplifiera dessa (se figur 6.6). Alla textfält på webbsidan har den bredd som det datafält i databasen som överensstämmer med textfältet har. Rullistorna får den bredd som det längsta värdet i rullistan har.

6.2.4 Transaktioner - Allt i lista

Transaktionsöversikt-sidan skiljer sig från andra översiktssidor på ett antal punkter (se figur 6.7). Precis som på andra översiktssidor visas data rad för rad, och längst till höger finns en mängd knappar. Skillnaden här är dock att summor visas längst ned i tabellen, och dessutom är några av kolumnerna summor.

Tidrapportering	Attestering	Fakturering	Projekt	Admininställningar	Mina egenskaper	Om	Logga ut
		Användare	Kunder	Systeminställningar			
		Visa användare	Lägg till				

Användarnamn:

Nytt lösenord:

Bekräfta nytt lösenord:

Startsida: Visa alla tider

Förnamn:

Efternamn:

Mobil:

Telefon:

E-post:

Rättigheter: Admin

Figur 6.6: Den färdiga “Lägg till användare”-sidan

	Tidrapportering	Attestering	Fakturering	Projekt	Admininställningar	Mina egenskaper	Om	Logga ut
			Visa transaktioner	Lägg till transaktion				

Kund	Projekt	Projektfas	Datum	Arbetstyp	Extra	Fakturareferens	Kvantitet	Tid kvar	å Pris	Kostnad	Total	Omkostnader	Avtalstotal	Status		
Stora Enso	Pappershanteringssystem	jasha	06-05-11	Konsult	Normal		8	88	50	400	1600					
Stora Enso	Pappershanteringssystem	jasha	06-05-10	Konsult	Normal		8	88	50	400	1600					
Stora Enso	Pappershanteringssystem	jasha	06-05-09	Konsult	Normal		8	88	50	400	1600					
Stora Enso	Pappershanteringssystem	jasha	06-05-08	Konsult	Normal		8	88	50	400	1600					
Dormy Golf	Bollplockningsmaskin	Konfigurering	06-05-03	Konsult	Intern		8	185	200	1600	700	1600				
Dormy Golf	Bollplockningsmaskin	Konfigurering	06-05-02	Konsult	Normal		7	185	100	700	700	1600				
							47	273		3900	2300	1600	0			

Figur 6.7: Den färdiga “Transaktionsöversikt”-sidan

6.2.5 Läggtill transaktion

Sidan för att lägga till nya transaktioner är den kanske viktigaste sidan i hela systemet. Den skiljer sig från andra “Lägg till”-sidor i det avseende att det går att lägga till flera poster samtidigt genom att använda Kopiera-knappen längst ner till vänster. En bild på sidan visas i figur 6.8. Den översta raden är samma för alla transaktioner som läggs till, och alla transaktioner registreras då användaren klickar på Registrera-knappen.

The screenshot shows a web application interface for adding a transaction. At the top, there is a navigation bar with links: Tidrapportering, Attestering, Fakturering, Projekt, Administrinställningar, Mina egenskaper, Om, and Logga ut. Below this, a header section contains: Kund: Stora Enso, Projekt: Pappershanteringssystem, Projektfas: Konfigurering, and Upparbetad tid: 32/120. The main form area includes: Datum: 060508, Arbetsstyp: Konsult, å Pris: 120 kr, and Kvantitet: 8. Below these are Extra: Normal, Fakturareferens: (empty), and Kostnad: 960 kr. The Beskrivning field contains: Designat databasen för pappershantering. The Utförlig beskrivning field contains: Ett E/R-diagram och en relationsmodell är skapad och finns att ladda ner. The Intern information field contains: Undersök hur det blir i praktiken i tabell massa_deg. At the bottom, there are two buttons: Kopiera and Registrera.

Figur 6.8: Den färdiga “Lägg till transaktion”-sidan

6.3 Kapitelsammanfattning

I detta kapitel har vi berättat om vilka delar av kravspecifikationen som implementerats och vad som prioriterats bort. Vi har även berättat om hur vi löst vissa intressanta aspekter av implementeringen. Vi har redogjort för varje del av systemet som implementerats och som vi tyckt värt att nämna i denna rapport.

Uppkoppling mot databasen och hanteringen av SQL-frågor sker med hjälp av klasserna DBQueryHandler och DBNonQueryHandler och använder sig av MySQL Connector/NET.

För inloggningen används cookies, både till att lagra användarinformation om den som är inloggad för tillfället, men även för att spara vilken användare som loggade in senast.

Genom att informationen finns i en cookie, kan systemet välja användaren automatiskt i listan nästa gång login-sidan besöks.

Menyn är en .NET User Control, vilket gör att den enkelt kan instansieras på alla sidor, och genom att skicka tre parametrar till den anges vad som ska visas i menyn. Den använder inga bilder vilket gör att den har hög prestanda.

För att kunna skicka formulärdata till en annan sida används ett JavaScript som ändrar formulärets action-attribut, och dessutom “__VIEWSTATE” som är ett .NET-fält som innehåller valideringsinformation för en sida.

Användargränssnittet skiljer sig lite i utseende från kravspecifikationen, men kravspecifikationens bilder var endast tänkta som en skiss. Det slutgiltiga användargränssnittet har ett mer professionellt utseende.

Kapitel 7

Resultat och framtida arbete

Vi har under detta examensarbete utvecklat inte bara en specifikation till ett tidregistrerings- och faktureringsystem åt Xeratech systems AB, utan även implementerat stora delar av det. Systemet går att använda efter vår implementation, men vidare utveckling krävs för att få någon användning av det eftersom det för tillfället inte går att attestera transaktioner eller generera fakturor. Vår implementation står dock som en grund, och är implementerad på ett sätt som gör vidareutveckling enkel.

Vi hade först trott att vi skulle hinna implementera mycket mer än vad vi till slut hann med, men desto längre arbetet gick framåt med specifikationen, desto tydligare blev det att vi inte skulle hinna implementera mer än den absolut väsentligaste delen – tidregistrering. För att få tidregistreringsdelen att fungera krävdes även inloggning och en mängd administrativa sidor endast för att få de primära funktionerna i systemet att fungera.

I kapitel 2 slogs det fast ett antal punkter som beskrev brister i det gamla systemet. Har vårt nya system löst dessa?

- Det gamla systemet var väldigt långsamt. Vi har försökt optimera prestanda i alla delar av systemet genom god design och systemet har utfört funktioner inom rimliga tider vid enkla tester.

- Inga standardvärden fanns i det gamla systemets, men i det nya systemet finns dessa på alla ställen där det är rimligt.
- Man behöver inte längre gå mellan olika sidor för att hämta information till inmatningsrutor och rullistor. Information som är av den art att den redan ska vara inskriven är inskriven.
- Flera transaktioner kan registreras samtidigt.
- Flera transaktioner kan attesteras samtidigt (detta är dock ej implementerat än, men står i kravspecifikationen).
- Projektledare kan attestera transaktioner (detta är dock ej implementerat än, men står i kravspecifikationen).
- Avtalsprojekt kan faktureras på samma sätt som vanliga projekt (detta är dock ej implementerat än, men står i kravspecifikationen).
- Databasen innehåller ingen onödig information, men saknar dock vissa delar för att vidare utveckling ska fungera, och måste då utökas.
- Menysystemet visar vilken sida användaren befinner sig på och är enkelt att manövrera sig fram med.

Systemet vi utvecklat är större och implementationen mer komplicerad än vad vi från början trodde, även om vi endast utvecklat en del av allt i kravspecifikationen.

Utvecklingsmiljön och språken som vi använt var till största del nya för oss, och trots allt detta har vi utvecklat ett väl fungerande, användarvänligt system. Resultatet av vårt arbete blev över förväntan, både i avseende på denna rapport och med avseende på kravspecifikationen och implementation.

7.1 Problem

Det som tagit mest tid att få att fungera som det ska är SQL-frågorna på transaktionöversikt-sidan. Här krävs nämligen inte bara olika SQL-frågor beroende på vilken accessnivå användaren har, utan det ska även räknas ut ett flertal olika sorters summer beroende på transaktioners egenskaper, och flera av dessa ska skrivas ut per projekt. Förutom SQL-frågan som ger all information till tabellen över transaktioner finns även en SQL-fråga som hämtar information till den nedersta raden där totaler summeras ihop. Att få ut rätt information med dessa SQL-frågor visade sig vara mer komplicerat, och tog mycket mer tid än väntat.

Några problem vi haft har att göra med hur saker är implementerade i .NET. För att en sida ska skicka sin formulärdata när en rullista ändras används en egenskap som kallas `AutoPostBack` och sätts till `True`. Dessutom används `SelectedIndexChanged` som sätts till den funktion som ska köras när sidans formulärdata skickas. Vi upptäckte dock efter mycket krångel att det inte får finnas någon knapp på sidan som har `ID=submit` för att detta ska fungera. Vi har inte lyckats utröna någon förklaring till varför det inte fungerar när en knapp har `ID=submit`.

Om vi fick möjlighet att göra alltsammans igen skulle vi skapat skript med väldefinierad testdata. Med data för alla möjliga utfall som vi dokumenterat hade vi kunnat enkelt jämföra detta med våra resultat efter implementation. På så sätt kunde vi enkelt ha testat SQL-frågorna för att se om de stämde.

7.2 Framtida arbete

Eftersom vi endast implementerat en del av systemet finns mycket kvar för Xeratech att utveckla. De tre stora delarna som är kvar är attestering, fakturering, och mina egenskaper. Förutom dessa finns även delar kvar i de sidor vi implementerat, eftersom vi

tvingades prioritera bort vissa bitar för att hinna med det mest väsentliga. Förutom det av kravspecifikationen som inte är implementerat har vi ett antal förslag på ytterligare förbättringar av systemet:

- Schemalagd backup av databasen – Det vore önskvärt att efter ett visst tidsintervall göra en kopia av databasen och lagra denna på säker plats utifall att något skulle hända med den nuvarande databasen.
- Kryptering av känslig information – För att utöka säkerheten borde känslig information krypteras. Detta ansågs inte vara så viktigt vid utvecklingen av systemet, men i framtiden kanske säkerhetsnivån kan behöva ökas.
- Möjlighet att sätta accessnivå på startsidor – I en användares inställningar står vilken sida denne ska komma till när han/hon loggat in. Men i listan av möjliga sidor tas inte hänsyn till vilken accessnivå användaren har, vilket gör att en användare kanske sätter att han/hon vill starta på en admin-sida, och kan sedan inte logga in i systemet.
- Ny accessnivå för kunder – En accessnivå för kunder som gör att en kund kan få ett inloggningsnamn och lösenord och sedan kan logga in i systemet för att se vad som gjorts. Tillträde för denna accessnivå måste vara begränsad, det vill säga att användaren kanske endast kommer åt vissa sidor där det går att visa information men inte ändra den.

Till sist hoppas vi att systemet kommer till stor nytta och fortsätter utvecklas till att bli ett bättre, mer funktionsenligt, och mer användarvänligt system än Xeratechs gamla system.

Litteraturförteckning

- [1] Alexander, John & Hollis, Billy (2002). *Developing Web Applications with Visual Basic.NET and ASP.NET*. New York: Wiley.
- [2] DuBois, Paul (2002). *MySQL Cookbook*. New York: O'Reilly.
- [3] Elmasri, Ramez & Navathe, Shamkant B (2004). *Fundamentals of database systems*. 4th edition. Boston: Addison Wesley.
- [4] Goodman, Danny (2001). *JavaScript Bible*. Gold Edition. New York: Hungry Minds.
- [5] MySQL AB (2006). *MySQL 5.1 Reference Manual*. (Elektronisk)
Tillgänglig:<<http://dev.mysql.com/doc/refman/5.1/en/index.html>>. (2006-04-26).
- [6] Sun Microsystems (2006). *Java Technology*. (Elektronisk)
Tillgänglig:<<http://java.sun.com/>>. (2006-04-06).
- [7] Thai, Thuan & Lam, Hoang Q (2002). *.NET Framework Essentials*. 2nd Edition. New York: O'Reilly.
- [8] TimeLog (2006). *Webbaserad tidrapportering och projektstyrning* (Elektronisk)
Tillgänglig:<<http://www.timelog.se>>.
- [9] Turtschi, Adrian, Werry, Jason, Hack, Greg, Albahari, Joseph, Nandu, Saurabh, & Lee, Wei Meng (2002). *C#.NET Web Developer's Guide*. New York: Syngress.
- [10] Vander Veer, Emily (2005). *JavaScript For Dummies*. 4th Edition. New York: Wiley.
- [11] Wyle, R Allan, Gilliam, Jason D, Ting, Charlton & Michaels, Sean (2002). *Pure JavaScript*. 2nd Edition. Indianapolis: Sams.

Bilaga A

Kravspecifikationen

All information som finns i tabellerna som beskriver bland annat företag, personer och priser är fingerad.

A.1 Menysystem

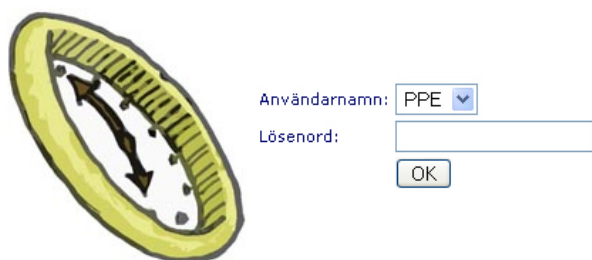
Användaren tar sig fram i systemet genom att använda menysystemet, som även visar vilken sidan användaren för tillfället befinner sig på. Menysystemet liknar det som används på Xeratechs hemsida, men har tre nivåer istället för två, och använder inte bilder för att visa text.



Figur A.1: Xeratechs menysystem

A.2 Login

Första sidan i systemet, dit användaren kommer om denne ej är inloggad, är inloggningsidan. Här väljer användaren sitt namn i en rullista och skriver in sitt lösenord i en textbox. Om användaren loggat in förut är dennes användarnamn redan valt i rullistan, och markören står på lösenordsboxen. När användaren valt sitt användarnamn, skrivit in sitt lösenord i textboxen, och tryckt på login-knappen, så matchas namnet och lösenordet mot en databas, och om det stämmer förflyttas användaren till den sida som denne har inställd i sina användarinställningar som startsida.



Figur A.2: Loginrutan

A.3 Tidregistrering

På tidregistreringssidorna kan användaren få en överblick över sina registrerade tider, registrera nya tider, och editera tider som registrerats förut.

A.3.1 Översikt

Det finns tre sätt för användaren att visa den tid som denne registrerat. I menyn står att välja mellan: Allt i lista, Kund-projekt översikt, och Kalender.

Allt i lista

Kund	Projekt	Projektfas	Datum	Arbetsstyp	Extra	Fakturareferens	Kvantitet	Tid kvar	å Pris	Kostnad	Total	Omkostnader	Avtalstotal	Status
Gabab AB	Tapetutskrift	Konfigurering	060110	Konsult	Avtal		4h	20h	500	2000	20000	-	70000	  
Gabab AB	Tapetutskrift	Konfigurering	060109	Konsult	Avtal		8h	20h	500	4000	20000	-	70000	  
Gabab AB	Färgsampling	Konfigurering	060106	Konsult	Normal		8h	16h	500	4000	17000	-	-	  
Gabab AB	Färgsampling	Konfigurering	060105	Konsult	Normal		8h	16h	500	4000	17000	-	-	  
Stora Enso	Papperssorteringssystem	Konfigurering	060104	Konsult	Internutgift		8h	57h	-	4500	20000	4500	-	  
Stora Enso	Papperssorteringssystem	Konfigurering	060103	Konsult	Normal		8h	57h	300	2400	20000	4500	-	  
Stora Enso	Papperssorteringssystem	Konfigurering	060102	Konsult	Normal		8h	57h	300	2400	20000	4500	-	  
							52h	93h		23300	57000	4500	70000	

Figur A.3: Allt i lista

Här visas alla användarens registrerade tider i en tabell (figur A.3). På höger sida om varje rad finns en knapp för att editera posten, och en för att ta bort den. De tider som attesterats eller fakturerats går inte att editera eller ta bort som det går med de andra tiderna. Genom att klicka på titeln till en kolumn sorteras alla rader efter detta attribut. Ovanför varje kolumn finns ett fält där det går att antingen välja ett av flera valbara värden eller skriva in ett värde, för att sedan visa alla rader som stämmer överens med detta. Det går att fylla i flera av dessa fält för att vidare specificera vad som skall visas. Längst till höger finns en knapp där användaren klickar för att utföra sökningen.

På varje rad visas:

- Vilken kund arbetet utförts åt
- I vilket projekt det utförts
- I vilken projektfas det utförts
- När arbetet utfördes (vilket datum)
- Vilken typ av arbete som utförts
- Om transaktionen har någon extrauppgift (till exempel om det är en intern avgift så anges det här)

- Om det finns någon fakturareferens
- Hur många timmar som arbetats
- Hur många timmar som är kvar i det budgeterade antalet timmar för det projektet
- Kostnaden per tidsenhet/avståndsenhet
- Kostnaden för transaktionen
- Den totala kostnaden inom projektet
- Den totala summan omkostnader i projektet (omkostnader är ej en delmängd av den totala kostnaden inom projektet)
- Den totala summan för avtalet om det är ett avtalsprojekt

Statuskolumnen förtäljer vilken status posten har:

gul boll – Registrerad, men inte attesterad

blå boll – Attesterad av projektledaren

grön boll – Attesterad av admin

röd boll – Underkänd

ett F – Fakturerad

ett D – Delfakturerad, det vill säga att minst en avbetalning på ett projekt med prisavtal är gjort, och minst en betalning är kvar

ett I – Attesterad av admin och är intern

ett A – Attesterad av admin och anger att det finns ett prisavtal

Om användaren trycker på editera-knappen, som står till höger om status-ikonen, kommer han/hon till en annan sida där all information om den posten visas, och går att ändra på. Denna knapp går dock inte att trycka på om denna post blivit attesterad av projektledare eller admin. Om användaren trycker på “ta bort”-knappen, som står till höger om editera-knappen, kommer det upp ett meddelande som frågar “Vill du verkligen ta bort den här posten?”. Om användaren klickar på “Ja” tas posten bort, men om användaren klickar på “Nej” händer ingenting. Denna knapp går dock inte att trycka på om denna post blivit attesterad av projektledare eller admin. Posten kan endast tas bort om den har status underkänd eller registrerad.

På den sista raden i tabellen visas totaler för kolumnerna kvantitet, tid kvar, kostnad, total, omkostnader, och avtalstotal.

Kund-projekt översikt

Kund	Projekt	Kvantitet	Kvantitet för alla	Beräknad projekttid	Tid kvar	Total	Omkostnader	Avtalstotal
Gabab AB	Färgsampling	16h	34h	50h	16h	17000	0	-
	Tapetutskrift	12h	40h	60h	20h	20000	0	70000
Stora Enso	Papperssorteringssystem	24h	63h	120h	57h	20000	4500	-

Figur A.4: Kund-projekt översikt

Här visas en sammanfattning för varje projekt (se figur A.4), där det står hur många timmar som användaren arbetat upp, hur många timmar som arbetats upp totalt på projektet, vad den beräknade arbetstiden är, hur mycket av den beräknade arbetstiden som är kvar (vilken kan vara negativ om antalet arbetade timmar överstigit den beräknade arbetstiden), vad kostnaden för användarens timmar är, om det funnits några omkostnader, och om det finns ett avtalat pris.

Kalender

Här visas användarens upparbetade tid i kalenderform (figur A.5). På varje dag visas vilken kund, vilket projekt, och hur många timmar. En månad i taget visas, och det går att bläddra mellan månader genom att klicka på den vänstra eller högra pilen på varsin sida om månadsnamnet.

Januari 2006						
Måndag	Tisdag	Onsdag	Torsdag	Fredag	Lördag	Söndag
						1
2 Stora Enso Papperssorteringssystem Konsult 8h	3 Stora Enso Papperssorteringssystem Konsult 8h	4 Stora Enso Papperssorteringssystem Konsult 8h	5 Gabab AB Färgsamplig Konsult 8h	6 Gabab AB Färgsamplig Konsult 8h	7	8
9 Gabab AB Tapetutskrift Konsult 8h	10 Gabab AB Tapetutskrift Konsult 4h	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Figur A.5: Kalender

A.3.2 Registrera ny tid

Kund: Xerotech Systems AB	Projekt: Tidregistreringssystem	Projektfas: Konfigurering	Upparbetat: 64/200h
Datum: 060101	Arbetstyp: Konsult	å Pris: 500 kr	Kvantitet: 8 h
Extra: Normal	Fakturareferens:		Kostnad: 5000 kr
Beskrivning:	Fortsatt utvecklingen av specifikationen		
Utförlig beskrivning:	Uppdaterat tidregistreringsdelen till att kunna fylla i flera samtidigt		
Intern information:	Denna sida är bara till för specifikationen, och kommer se bättre ut efter implementering.		
Datum: 060102	Arbetstyp: Konsult	å Pris: 500 kr	Kvantitet: 8 h
Extra: Normal	Fakturareferens:		Kostnad: 5000 kr
Beskrivning:	Fortsatt utvecklingen av specifikationen		
Utförlig beskrivning:	Uppdaterat tidregistreringsdelen till att kunna fylla i flera samtidigt		
Intern information:	Denna sida är bara till för specifikationen, och kommer se bättre ut efter implementering.		
Kopiera		Registrera	

Figur A.6: Lägg till nya tider

På denna sida kan användaren registrera en eller flera tider samtidigt (se figur A.6). Om användaren registrerar flera tider får alla dessa samma kund, projekt och projektfas. För att få fler poster att fylla i klickar användaren på kopiera-knappen, varefter ett

popup-fönster visas där användaren får fylla hur många kopior av den nedersta posten som skall skapas. Efter att ha tryckt på OK-knappen läggs det till så många poster som användaren valt, med samma information som den nedersta posten, och där datumen är stigande. Man kan stega sig fram mellan alla attribut, och genom att hovra muspekaren över ett objekt visas en förklarande text om just det objektet. Alla uppgifter utom fakturareferens och beskrivningarna är obligatoriska.

När användaren skrivit in sina tider, trycker denne på registrera-knappen, varefter ett popup-fönster frågar om användaren ska fortsätta. Väljer denne "Ja" sparas informationen i databasen och användaren flyttas till översiktssidan. Väljer han/hon "Nej" försvinner popup-fönstret och inget annat händer.

Kund – Kunden väljs i en rullista fylld med alla registrerade kunder som har projekt som användaren är uppskriven på. När en kund väljs i denna lista uppdateras projektlistan på så sätt att den fylls med de projekt som tillhör den valda kunden.

Projekt – Projektet väljs i en rullista fylld med de projekt som är associerade med den valda kunden. Om ingen kund är vald är denna lista tom. När en kund väljs och denna lista uppdateras med projekt, sätts det projekt med lägst projektnummer som standardvärde. Om ett internprojekt väljs, det vill säga ett projekt som är satt att ha egenskapen internprojekt, väljs internprojekt i Extra-listan automatiskt. Om ett projekt med prisavtal väljs, väljs avtal i Extra-listan automatiskt.

Projektfas – Projektfasen är en delaktivitet av ett projekt, där exempel skulle kunna vara konfigurerings, validering, eller löpande stöd. Användaren väljer projektfasen i en rullista, där värdena i listan är samma för alla projekt. Om ett standardvärde för projektfas i detta projekt är satt av projektledaren eller admin, så är rullistan satt till detta.

Upparbetat – En text visar hur många timmar som upparbetats totalt inom projektet, och hur många timmar som är beräknat.

Datum – Användaren kan skriva in datumet för det som rapporteras direkt i en textbox, där det som standardvärde står dagens datum. Användaren kan även klicka på en ikon bredvid textboxen och få upp en kalender ur vilken han/hon kan välja relevant datum.

Arbetstyp – Arbetstypen säger vilken roll användaren haft under den tid som registreras, till exempel konsult, projektledare, restid, eller milersättning. Användaren väljer arbetstyp ur en rullista, där värdena i listan är samma för alla projekt. Om ett standardvärde för arbetstyp i detta projekt är satt av projektledaren eller admin, så är rullistan satt till detta.

å Pris – Kostnaden per arbetsenhet för det utförda jobbet. Har som standardvärde det värde som är satt för den valda arbetstypen i det valda projektet.

Kvantitet – Antalet timmar användaren jobbat, eller antal km han/hon rest. Användaren skriver in kvantiteten i en textbox, till höger om vilken det står antingen h eller km, beroende på vilken arbetstyp som valts.

Extra – I extra-listan kan användaren välja en extra egenskap för den registrerade tiden. Det finns 7 val:

Normal – Anger att denna tidregistrering skall faktureras normalt.

Intern – Anger att denna tidregistrering är intern, och skall inte faktureras kund. Den kommer inte med på fakturan.

Avtal – Anger att detta projekt har ett fast pris kopplat till sig, och tidregistreringen skall inte faktureras kund. Denna väljs automatiskt om projektet som valts har ett avtalat pris, och detta val kan då inte ändras. Den kommer inte med på fakturan.

Kundutgift – Anger att denna tidregistrering är ett utlägg som skall faktureras kund. Den kommer med på fakturan.

Internutgift – Anger att denna tidregistrering är ett utlägg som inte skall faktureras kund. Den kommer inte med på fakturan.

Kundreseutgift – Anger att denna tidregistrering är en restid eller milersättning som skall faktureras kund. Den kommer med på fakturan.

Internreseutgift – Anger att denna tidregistrering är en restid eller milersättning som inte skall faktureras kund. Den kommer inte med på fakturan.

Fakturareferens – Användaren väljer en referens i en rullista. Standardvärdet är ingenting, det vill säga tomt. Denna skall endast användas om någon annan än användaren skall vara med på fakturan som referens.

Kostnad – Kostnaden för det registrerade arbetet står i en textbox och har som standardvärde priset för arbetstypen multiplicerat med kvantiteten.

Beskrivning – Användaren kan skriva in en kortfattad beskrivning om vilket arbete som utförts i en textbox, denna text skrivs sedan med på fakturan.

Utförlig beskrivning – Användaren kan skriva in en extra rad beskrivning i en textbox, som bara skrivs med på fakturan om kunden vill ha det.

Intern information – Användaren kan skriva in interna meddelanden i en textbox. Dessa kommer inte med på fakturan. Om en tidregistrering underkänns av projektledare eller admin, så skickas denna text med i ett mail till ägaren av den underkända tidregistreringen.

A.4 Fakturering

Endast admin har tillgång till faktureringsidan (figur A.7). Överst på sidan väljs vilken period som skall visas, och om det skall vara avtalsprojekt eller inte. Som standardvärde är nuvarande månad vald, det vill säga första dagen i nuvarande månad står i den första

textboxen, och sista dagen i nuvarande månad står i den andra textboxen. Om kryssrutan för avtal inte är ifylld visas inga avtalsprojekt. Det som faktureras på ett projekt med prisavtal är nästa ofakturerade betalning (antal betalningar och hur mycket som skall betalas vid varje är satt i projektinställningar). Om det finns kvar betalningar när denna betalning fakturerats, blir statusen “delfakturerad”, annars blir statusen “fakturerad”. Som standard är denna inte ifylld. Användaren trycker sedan på uppdatera-knappen, varefter tabellen därunder fylls med alla registrerade tider under den valda perioden. Genom att klicka på titeln till en kolumn sorteras alla rader efter detta attribut. Ovanför varje kolumn finns ett fält där det går att antingen välja ett av flera valbara värden eller skriva in ett värde, för att sedan visa alla rader som stämmer överens med detta (datum kolumnen har inget sådant fält, eftersom detta väljs i rutan ovanför tabellen). Det går att fylla i flera av dessa fält för att vidare specificera vad som skall visas. Längst till höger finns en knapp där användaren klickar för att utföra sökningen. På den sista raden i tabellen visas totaler för kvantitet, tid kvar, kostnad, total, omkostnader, och avtalstotal.

Kund	Projekt	Projektfas	Konsult	Datum	Arbetstyp	Extra	Fakturareferens	Kvantitet	Tid kvar	å Pris	Kostnad	Total	Omkostnader	Avtalstotal	Status
Gabab AB	Färgsampling	Konfigurering	PPE	060106	Konsult	Normal		8h	16h	500	4000	17000	-	-	🟡
Gabab AB	Färgsampling	Konfigurering	PPE	060105	Konsult	Normal		8h	16h	500	4000	17000	-	-	🟡
Stora Enzo	Papperssorteringsystem	Konfigurering	PPE	060104	Konsult	Internutgift		8h	57h	-	4500	20000	4500	-	🟡
Stora Enzo	Papperssorteringsystem	Konfigurering	PPE	060103	Konsult	Normal		8h	57h	300	2400	20000	4500	-	🟡
Stora Enzo	Papperssorteringsystem	Konfigurering	PPE	060102	Konsult	Normal		8h	57h	300	2400	20000	4500	-	🟡
								40h	73h		17300	37000	4500		

Figur A.7: Fakturering

De värden som visas i tabellen är kund, projekt, projektfas, konsult, datum, arbetstyp, extra, kvantitet, tid kvar, å pris, kostnad, total, omkostnader, avtalstotal, och status. Till höger om varje rad finns en knapp för att editera posten, och en kryssruta som är ifylld som standard och innebär att denna post kommer med på fakturan. En post som inte skall vara med på fakturan, till exempel om det är en internutgift, får denna kryssruta

tom som standard.

När användaren är färdig för att fakturera, eller bara vill se så att det stämmer, trycker denne på förhandsgranska-knappen. Därefter genereras en faktura i html-format och visas. Användaren kan sedan klicka på en knapp på den sidan, vilket gör att en pdf-fil skapas med samma innehåll som den genererade html-sidan, vilken sedan kan skrivas ut, eller sparas, eller bara läsas på skärmen (möjligheten att spara och skriva ut filen tillhandahålls av pdf-läsaren, och är ingen inbyggd funktion i detta system).

A.5 Attestering

Endast admin och projektledare har tillgång till attesteringsidan (figur A.8).

Projektledare har dock endast tillgång till de projekt som denne är uppskriven på. Här kontrolleras alla registrerade tider av projektledare och admin, varefter de kan välja att attestera eller underkänna poster. Förutom dessa två funktioner finns även möjlighet att editera och ta bort en post, precis som på tidregistreringsidan. Attestering och underkänning går till så att användaren kryssar i checkboxar till höger om alla de poster som skall attesteras/underkännas. Därefter trycker användaren på antingen attestera-knappen eller underkänn-knappen, varpå uppdateringar sker och sidan laddas in igen.

Kund	Projekt	Projektfas	Konsult	Datum	Arbetstyp	Extra	Fakturareferens	Kvantitet	Tid kvar	å Pris	Kostnad	Total	Omkostnader	Avalstatotal	Status			
Gabab AB	Tapetutskrift	Konfigurering	PPE	060110	Konsult	Avtal		4h	20h	500	2000	20000	-	70000				
Gabab AB	Tapetutskrift	Konfigurering	PPE	060109	Konsult	Avtal		8h	20h	500	4000	20000	-	70000				
Gabab AB	Färgsampling	Konfigurering	PPE	060106	Konsult	Normal		8h	16h	500	4000	17000	-	-				
Gabab AB	Färgsampling	Konfigurering	PPE	060105	Konsult	Normal		8h	16h	500	4000	17000	-	-				
Stora Enzo	Papperssorteringssystem	Konfigurering	PPE	060104	Konsult	Internutgift		8h	57h	-	4500	20000	4500	-				
Stora Enzo	Papperssorteringssystem	Konfigurering	PPE	060103	Konsult	Normal		8h	57h	300	2400	20000	4500	-				
Stora Enzo	Papperssorteringssystem	Konfigurering	PPE	060102	Konsult	Normal		8h	57h	300	2400	20000	4500	-				
								52h	93h		23300	57000	4500	70000				

Figur A.8: Allt i lista

På den sista raden i tabellen visas totaler för kvantitet, tid kvar, kostnad, total,

omkostnader, och avtalstotal.

Genom att klicka på titeln till en kolumn sorteras alla rader efter detta attribut. Ovanför varje kolumn finns ett fält där det går att antingen välja ett av flera valbara värden eller skriva in ett värde, för att sedan visa alla rader som stämmer överens med detta. Det går att fylla i flera av dessa fält för att vidare specificera vad som skall visas. Längst till höger finns en knapp där användaren klickar för att utföra sökningen.

A.6 Admininställningar

Dessa sidor har endast admin tillgång till.

A.6.1 Användare

Översikt

Denna sida består av en lista av systemets alla användare (figur A.9). Längst till höger finns liknande knappar som på tidrapporteringssidan:

Editera – Om användaren trycker på den första knappen (det vill säga den längst till vänster) kommer han/hon till en annan sida där all information om den posten visas, och går att ändra på.

Ta bort – Trycker användaren på den andra knappen kommer det upp ett meddelande som frågar “Vill du verkligen ta bort den här posten?”. Om användaren klickar på “Ja” här, tas posten bort, men om användaren klickar på “Nej” händer ingenting. (OBS! En användare kan tas bort *om och endast om* denne inte har någon tid registrerad i systemet.)

Användarnamn	Förnamn	Efternamn	Rättigheter	Mobil	Telefon	E-post		
PPE	Pelle	Persson	Konsult	0704-734190	054-511234	pelle.persson@xeratech.se		

Figur A.9: Användaröversikt

Användarnamn:

Lösenord:

Lösenord igen:

Rättigheter: ▼

Förnamn:

Efternamn:

Mobil:

Telefon:

E-post:

Startsida: ▼

Figur A.10: Lägg till användare

Lägg till användare

På denna sida kan admin lägga till en ny användare och specificera dess egenskaper och rättigheter (se figur A.10).

Användarnamn – Admin anger användarens användarnamn i en textbox. Detta namn kan vara vilken sträng av bokstäver och siffror som helst, inte längre än 10 tecken och inte kortare än 3 tecken. Denna uppgift är obligatorisk.

Lösenord – Admin anger användarens lösenord i en textbox. Detta lösenord kan vara vilken sträng av tecken som helst utom mellanslag, tabb, och retur, inte längre än 20 tecken och inte kortare än 7 tecken. Lösenordet måste skrivas in två gånger, det vill säga även i en textbox direkt under den första, för att minska sannolikheten att fel lösenord skrivs in. Denna uppgift är obligatorisk.

Rättigheter – Admin anger i en rullista vilken rättighetsnivå den nya användaren ska

ha. Alternativen är konsult, projektledare, och admin. Som standard står rullistan på konsult. Denna uppgift är obligatorisk.

Förnamn – Admin anger användarns förnamn i en textbox. Detta kan vara vilken sträng av bokstäver som helst, inte längre än 20 tecken. Denna uppgift är obligatorisk.

Efternamn – Admin anger användarens efternamn i en textbox. Detta kan vara vilken sträng av bokstäver som helst, inte längre än 20 tecken. Denna uppgift är obligatorisk.

Mobil – Admin anger användarens mobilnummer i en textbox. Detta kan vara vilken sträng av siffror och bindelsträck som helst, inte längre än 20 tecken, som uppfyller det reguljära uttrycket för telefonnummer.

Telefon – Admin anger användarens telefonnummer i en textbox. Detta kan vara vilken sträng av siffror och bindelsträck som helst, inte längre än 20 tecken, som uppfyller det reguljära uttrycket för telefonnummer.

E-post – Admin anger användarens e-post i en textbox. Detta kan vara vilken sträng av tecken som helst som uppfyller det reguljära uttrycket för e-post-adresser.

Startsida – Admin väljer i en rullista vilken startsida användaren skall ha.

OK – När admin fyllt i relevanta uppgifter trycker denne på OK-knappen. Om någon obligatorisk uppgift inte är ifylld, de två lösenorden inte är lika, eller något av fälten bryter mot någon regel om vilka tecken som får användas, så skrivs detta ut i ett meddelande och admin kan rätta till felen. När alla uppgifter stämmer och admin trycker på OK-knappen, visas ett meddelande att användaren lagts till, och admin förflyttas till översiktssidan över användare.

Editera användare

Denna sida är identisk med “Lägg till användare”-sidan, med den skillnaden att användaren som editeras redan måste existera i systemet, och ingen ny användare läggs till då OK-knappen trycks, utan egenskaperna för den befintliga användaren uppdateras.





A.6.2 Systeminställningar





På denna sida editeras alla systeminställningar (se figur A.11). De delar som kan editeras här är leverantör, arbetstyper och dess standardpris, och projektfaser.





Leverantör – Längst upp finns en tabell över alla leverantörer och deras attribut. Till höger om varje rad finns knappar för att editera en rad, och ta bort en rad.

Arbetstyp – Under leverantörtabellen finns en tabell som innehåller de nuvarande arbetstyperna och dess standardpriser.

Projektfas – Längst ned på sidan finns en tabell som innehåller de nuvarande projektfaserna.

Leverantörer														
Nr	Namn	Adress1	Adress2	Adress3	Adress4	Telefon1	Telefon2	Fax	E-post	Referens1	Referens2	Organisationsnr		
1	Xeratech Systems AB	Box 415	651 09	Karlstad		0702-79 99 71	054-100260	054-210704	eloisa.westerlund@xeratech.se	Eloisa Westerlund		556632-4405		
2	Xeratech Systems AB	Kungstorget 5	411 17	Göteborg		054-100260		054-210704	eloisa.westerlund@xeratech.se	Eloisa Westerlund		556632-4405		

Arbetstyp		Standard timpris			
Konsult		200			
Projektleddare		300			

Projektfas				
Konfigurering				
Validering				




Figur A.11: Systeminställningar

Det finns även en sida för att lägga till en ny leverantör, en sida för att lägga till en ny arbetstyp, och en sida för att lägga till en ny projektfas.

A.6.3 Kunder

Översikt

På denna sida visas en lista av alla kunder inlagda i systemet (se figur A.12), och bredvid varje post finns en knapp för att editera posten, och en för att ta bort den.

Kunder														
Nr	Namn	Adress1	Adress2	Adress3	Adress4	Telefon1	Telefon2	Fax	E-post	Referens1	Referens2	Organisationsnr		
40001	Stora Enso	Ensogatan 1	651 10	Karlstad		0702-12 34 56	054-123456	054-654321	Olle.Enso@stora.se	Olle Enso		556644-4401		
40002	Gabab AB	Gababvägen 12	651 11	Karlstad		0704-13 57 90	054-112233	054-332211	Sven.Svensson@gabab.nu	Sven Svensson		556647-4410		

Figur A.12: Kundöversikt

Lägg till kund



The form contains the following fields:

- Nr: 40003
- Namn: [empty]
- Adress1: [empty]
- Adress2: [empty]
- Adress3: [empty]
- Adress4: [empty]
- Telefon1: [empty]
- Telefon2: [empty]
- Fax: [empty]
- E-post: [empty]
- Referens1: [empty]
- Referens2: [empty]
- Organisationsnr: [empty]

Buttons: OK, Avbryt

Figur A.13: Lägg till kund

På denna sidan kan användaren fylla i uppgifter för en ny kund som läggs till i systemet då användaren klickar på OK-knappen (se figur A.13). Trycker användaren på avbryt-knappen förflyttas han/hon till kundöversiktssidan.

Editera kund

Denna sida är identisk med “Lägg till kund”-sidan, med den skillnaden att kunden som editeras redan måste existera i systemet, och ingen ny kund läggs till då OK-knappen trycks utan egenskaperna för den befintliga kunden uppdateras.

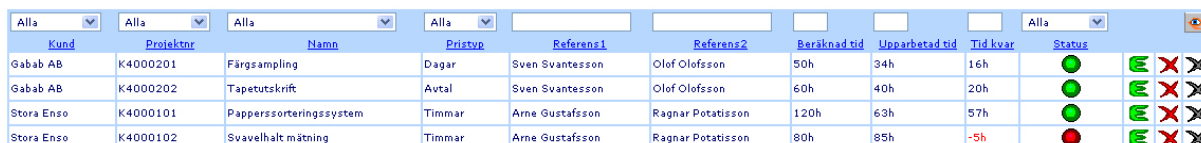
A.7 Projekt

Endast admin och projektledare har tillgång till dessa sidor. Projektledare kan endast ändra på de projekt som denne är anmäld till.

A.7.1 Översikt

På denna sida visas en lista av alla projekt i systemet (se figur A.14), och bredvid varje post finns en knapp för att editera posten, en för att ta bort den, och en för att avsluta projektet. Ett projekt som avslutats försvinner från alla listor, det vill säga att det går inte längre att registrera tid på ett sådant projekt. Ett avslutat projekt finns dock fortfarande kvar i databasen, och markeras i statuskolumnen med en röd boll.

Genom att klicka på titeln till en kolumn sorteras alla rader efter detta attribut. Ovanför varje kolumn finns ett fält där det går att antingen välja ett av flera valbara värden eller skriva in ett värde, för att sedan visa alla rader som stämmer överens med detta. Det går att fylla i flera av dessa fält för att vidare specificera vad som skall visas. Längst till höger finns en knapp där användaren klickar för att utföra sökningen.



Kund	Projekt nr	Namn	Pristyp	Referens1	Referens2	Beräknad tid	Upparbetad tid	Tid kvar	Status			
Gabab AB	K4000201	Färgsampling	Dagar	Sven Svantesson	Olof Olofsson	50h	34h	16h	●	E	X	X
Gabab AB	K4000202	Tapetuskrift	Avtal	Sven Svantesson	Olof Olofsson	60h	40h	20h	●	E	X	X
Stora Enso	K4000101	Papperssorteringsystem	Timmar	Arne Gustafsson	Ragnar Potatisson	120h	63h	57h	●	E	X	X
Stora Enso	K4000102	Svavelhalt mätning	Timmar	Arne Gustafsson	Ragnar Potatisson	80h	85h	-5h	●	E	X	X

Figur A.14: Projektöversikt

A.7.2 Lägg till projekt

Lägg till projekt

Kund:

Projektnr:

Projektamn:

Referens1:

Referens2:

Beräknade projekttimmar:

Pris

Pristyp:

Arbetstyp	Pris
Konsult	<input type="text" value="200"/>
Projektledare	<input type="text" value="300"/>

Standardvärde för projektfas:

Standardvärde för arbetstyp:

Figur A.15: Lägg till projekt

På denna sida (se figur A.15) fyller användaren i uppgifter om projektet och trycker sedan på OK för att lägga till det i systemet, eller avbryt för att komma tillbaka till projektöversiktssidan utan att något ändras.

Kund – Användaren väljer kund i en rullista, varefter kundnumret i projektnummerrutan uppdateras.

Projektnummer – I projektnummerrutan fyller användaren i vilket projektnummer projektet skall ha. Projektnumret står på formen ortprefix (1 tecken) + kundnummer (5 tecken) + löpnummer/projekttyp (2 tecken), till exempel G4001100. De olika projekttyperna är:

- 00 – Förvaltningsprojekt.

- 01-49 – Löpandeprojekt.
- 50 – Supportprojekt.
- 51-98 – Löpandeprojekt.
- 99 – Garanti.

Projektnamn – Användaren anger projektets namn i en textbox.

Referens1 och 2 – Användaren anger projektets referenser i två textboxar.

Beräknade projekttimmar – Användaren anger beräknad projekttid i en textbox.

Pris – I prisrutan väljer användaren först vilken pristyp som skall användas i projektet.

Alternativen är timmar, dagar, veckor, och avtal. Om något av de tre första alternativen väljs kommer det upp en tabell över arbetstyper och de priser som hör till dessa. Om däremot avtal väljs, kommer det upp en textbox där användaren kan fylla i kostnaden för första delbetalningen (se figur A.16). Önskas fler delbetalningar trycker användaren på “Lägg till”- knappen, varpå en ny textbox läggs till under den första. Första gången en faktura görs till detta projekt används delbetalning 1, andra gången delbetalning 2, och så vidare.

Standardvärde för projektfas – Användaren anger vilket värde som ska stå som standard för projektfas när en användare registrerar tid på detta projekt.

Standardvärde för arbetstyp – Användaren anger vilket värde som ska stå som standard för arbetstyp när en användare registrerar tid på detta projekt.

A.7.3 Editera projekt

Denna sida är identisk med “Lägg till projekt”-sidan, med den skillnaden att projektet som editeras redan måste existera i systemet, inget nytt projekt läggs till då OK-knappen

Pris

Pristyp: Avtal ▼

Betaling 1: 12000 Ta bort

Betaling 2: 13000 Ta bort

Lägg till

Figur A.16: Prisrutan vid avtal

trycks utan egenskaperna för det befintliga projektet uppdateras, och projektnumret går inte att ändra.

A.8 Mina egenskaper

A.8.1 Inställningar

På denna sidan kan en användare ändra sina användarinställningar, och är identisk i utseende med “Lägg till användare”-sidan, förutom att rättigheter inte går att ändra här.

A.8.2 Projektanmälningar

Här visas en lista över de projekt som användaren är anmäld till. Det går att avanmäla sig från ett projekt genom att klicka på knappen till höger om en projektrad, men det går endast att göra detta om användaren inte registrerat någon tid på projektet.

A.8.3 Ny projektanmälan

På denna sida kan användaren välja kund i en rullista, därefter ett projekt i en rullista, och klicka på “Anmäl mig”-knappen för att anmäla sig till det projektet.

A.9 Om

Här står att läsa vilka som varit med och utvecklat systemet, när det gjordes, och en länk till den sida där rapporten för arbetet ligger.