

Projekt Mobow

Din väg till laddningsstationen

12/11-2014

Högskolan i Gävle

Karl Lundh, Niklas Sjögren, Rickard Hedlund

INNEHÅLL

| | |
|---|---|
| Inledning | 1 |
| Bakgrund och syfte | 1 |
| Tankar inför projektarbetet | 1 |
| Projektet | 1 |
| Projektets ursprungliga omfattning | 1 |
| Ändringar | 1 |
| Erfarenheter | 2 |
| Projektorganisation | 2 |
| Organisering | 2 |
| Fördelning av arbete | 2 |
| Förändringar av projektorganisationen | 2 |
| Milstolpar och viktiga beslut | 2 |
| Milstolpar | 2 |
| Viktiga beslut | 3 |
| Erfarenheter | 3 |
| Arbetsgång | 4 |
| Första sidan – Kartan | 4 |
| HTML | 4 |
| CSS | 4 |
| JavaScript & jquery | 4 |
| PHP | 4 |
| Databas | 4 |
| Administrering | 5 |
| CSS | 5 |
| Inloggning | 5 |
| Första sidan | 5 |
| Skapa | 7 |
| Faktura | 7 |
| Felrapportering | 8 |

| | |
|---|----|
| Radera | 9 |
| Editera | 9 |
| Meddelande | 10 |
| Arbetsmetoder | 11 |
| Tekniska | 11 |
| Organisatoriska | 11 |
| Planering och uppföljning | 11 |
| Erfarenheter | 11 |
| Beskrivning av systemet | 12 |
| Teknisk beskrivning | 12 |
| Design | 12 |
| Databas | 12 |
| Use Cases | 13 |
| Grafisk Design | 14 |
| Säkerhet | 14 |
| Erfarenheter | 15 |
| Diskussion | 15 |
| Levererade Systemet | 15 |
| Arbetsprocessen - Hur vi har jobbat | 15 |
| Grupparbetet | 15 |
| Kontakt med beställare | 16 |
| Slutsatser | 16 |
| Bilaga | 17 |
| Bilaga 1 – Kravspecifikation | 17 |
| Bilaga 2 – Projektplan | 20 |
| Bilaga 3 – Manual för programmet | 22 |
| Bilaga 4 – Databas mobowdb | 33 |
| Tabellstruktur adress | 33 |
| Tabellstruktur edit_foretag | 33 |

| | |
|------------------------------------|----|
| Tabellstruktur edit_kntper | 33 |
| Tabellstruktur faktura | 33 |
| Tabellstruktur felmeddelande | 34 |
| Tabellstruktur felstatus | 34 |
| Tabellstruktur feltyp | 34 |
| Tabellstruktur foretag | 34 |
| Tabellstruktur ikontyp | 34 |
| Tabellstruktur kontaktperson | 34 |
| Tabellstruktur kontrakt | 35 |
| Tabellstruktur medstatus | 35 |
| Tabellstruktur msg | 35 |
| Tabellstruktur oppettider | 35 |
| Tabellstruktur specialtider | 35 |
| Tabellstruktur veckodagar | 36 |
| Bilaga 5 – SQL | 37 |

INLEDNING

BAKGRUND OCH SYFTE

Mobow (www.mobow.se) är ett Gävlebaserat företag som är verksam med laddningsstationer för mobiltelefoner. Dessa laddningsstationer kan placeras på publika ställen så som caféer, bibliotek, affärer eller liknande. I dessa laddningsstationer har kunder och besökare möjligheten att ladda sin mobiltelefon samtidigt som de nyttjar de publika ställenas erbjudanden. Med dessa laddningsstationer i närheten av kunden så löser detta problemet med kort batteritid för smartphone-användare samtidigt som laddningsstationerna också hjälper de publika ställena att locka besökare.

I dagsläget finns det ingen mobilapplikation som kan hjälpa smartphone-användare att identifiera var Mobows laddningsstationer finns utplacerade. För att en smartphone-användare enkelt och snabbt skall kunna hitta var laddningsstationer finns utplacerade så behövs en mobilapplikation. Detta är vad det föreslagna examensarbetet inriktas på. Uppdraget är att utforska behov och krav kring en mobilapplikation som behandlar geografisk data, samt skapa denna för Android-plattformen. Grundfunktionaliteten är att applikationen enkelt skall visa var laddningsstationer finns på en karta med hjälp av mobiltelefonens GPS-funktion. Utöver detta så kan fler funktioner läggas till. Viktigt är att applikationen är lättanvänd och tilltalande.

TANKAR INFÖR PROJEKTARBETET

Det verkade vara ett intressant projekt. Det är alltid givande att få applicera sina kunskaper i ett skarpt projekt. Första tanken var att det skulle vara spännande att testa något annat än bara Java, att det mynnade ut i ett helt nytt språk trodde jag inte att det skulle.

PROJEKTET

PROJEKTETS URSPRUNGLIGA OMFATTNING

Den ursprungliga omfattningen av projektet var att skapa en mobil applikation för företaget Mobow, som skulle kunna visa var närmaste laddningsstationer finns för mobilanvändarna. Det var tänkt att använda en karta och sätta ut punkter där laddningsstationerna är placerade (som Mobows hemsida). Alltså en kopia av Mobows nuvarande hemsida därav följande ändringar.

ÄNDRINGAR

Eftersom Mobow redan hade implementerat en databas för kartfunktioner på deras hemsida beslutades det att detta var en för simpel uppgift, och därför ändrades projektets omfattning till att skapa en egendatabas, webb-samt mobilapplikation och skapa ett administrativt program för Mobow samt dess företagskunder(se bilaga kravspecifikation).

ERFARENHETER

Då projektgruppens medlemmar har olika bakgrund så varierade tidigare erfarenheter i projektgruppen. Ett beslut om vilka språk som skulle användas genomfördes och HTML5 valdes med motiveringen att den erbjuder möjligheten skapa en plattformsoberoende applikation för alla enheter, inklusive mobila. Utöver HTML5 valdes PHP som skriptspråk på serversidan för enkel integration med databas på server. PHP valdes då Mobows nuvarande webbhotell stödjer språk och för att de flesta i gruppen hade tidigare erfarenheter av detta språk och som databashanteringssystem valdes MySQL då den stöds av webbhotellet och är lätt integrerad med PHP. Dessa val resulterade dock i en del inlärninng då även ytterligare språk och kunskaper behövdes för att kunna skapa detta system.

PROJEKTORGANISATION

ORGANISERING

Projektorganisationen har haft en plattstruktur genom projektets fortlöpning, detta resulterade i att upplägget av projektet har skett på ett demokratiskt tillvägagångssätt, alla har varit med om alla beslut i projektet. Tidigt beslutades det om att Niklas skulle bli projektledare, för att någon behövde ha kontakt med kund samt handläggare. Projektmetodiken som projektorganisationen har använt sig utav är den agila metoden SCRUM.

FÖRDELNING AV ARBETE

I början av projektet skapades en visuell design av applikationen som hela gruppen var delaktig i utformningen. Utifrån denna skapades en lista med uppgifter som behövde utföras för att nå projektets mål. Fördelningen av arbetet har varit jämt mellan gruppens medlemmar och arbetsuppgifterna har valts utav medlemmarna själva ifrån en lista utav uppgifter som skapats, editrats och fyllts på under arbetets gång.

FÖRÄNDRINGAR AV PROJEKTORGANISATIONEN

Den största förändringen i projektorganisationen var när Elias beslutade att han skulle hoppa av kursen, detta orsakade en oförväntad tidsbrist då projektgruppen förlorade en fjärdedel av dess arbetskraft.

MILSTOLPAR OCH VIKTIGA BESLUT

MILSTOLPAR

1. Vecka 43 Start av projekt
2. Vecka 45 Visualisering av systemet
3. Vecka 47 Börjat koda
4. Vecka 47 Ny godkänd kravspecifikation
5. Vecka 48 Ny Projektplan
6. Vecka 49 Mellanseminarium
7. Vecka 49 Börjat med mobilapplikationen
8. Vecka 50 Börjat med rapporten
9. Vecka 50 Satt upp Server som David haft tillgång till
10. Vecka 3 Färdigställt programmet

VIKTIGA BESLUT

1. I början av projektet insågs det att uppgiften var för enkel, därför ändrades projektplanen avsevärt till att infatta en ny databas, hemsida samt mobilapplikationen som ursprungligen beställdes.
2. Valet av att använda oss utav HTML5, PHP, JavaScript, JQuery, CSS och MySQL har varit viktiga beslut då det har utmynnat i vår vision om ett hållbart system.

ERFARENHETER

Gällande punk 1 i viktiga beslut så har det visat sig vara ett bra val att göra ett större program än vad som i början krävdes. Det hade tagit ungefär en månad för att färdigställa applikationen och en databas hade ej behövt skapats en databas då Mobow använder sig utav en egen i dagsläget.

Gällande punk 2 så har det visat sig vara bra språk att jobba med, trots den stora kompetens skillnaden har inläringen varit givande då gapet mellan kompetensnivåerna har blivit reducerat. På grund av detta har utvecklingen av programmet inte varit något problem.

ARBETSGÅNG

Under denna rubrik kommer vi att gå igenom arbetsgången och problem för varje flik i hemsidan.

FÖRSTA SIDAN – KARTAN

Första sidan innehåller kartan för alla platser. Ingen hade tidigare någon erfarenhet av Google Maps API därför har ett flertal funktioner testats och byts ut.

HTML

HTML-koden innehåller ett HTML-skelett med samma grundläggande struktur som använts på alla andra sidor. I kroppen av sidan finns även en div som används som mål för JavaScriptet som utnyttjar Google Maps API.

CSS

Den del av CSS-filen som är specifik för denna sida specificerar hur rutorna som visas av JavaScriptet ska se ut.

JAVASCRIPT & JQUERY

Scriptet implementerar Google Maps API, med hjälp av detta API skapas innehållet i den div som ligger i HTML-koden ("div" heter googleMap). För varje plats som finns i databasen skapas egendesignade rutor på kartan som visas vid ett klickevent. JQuery används mestadels för att förenkla JavaScriptet genom att använda dess redan fördefinierade funktioner.

PHP

Utnyttjas för att hämta alla punkter och relevant information ur databasen och skickar dessa till JavaScript och JQuery på klientsidan. SQL-frågan som hämtar relevant data ur databasen kontrollerar också om det, enligt databasen, är öppet eller stängt på varje individuell punkt och väljer utifrån dessa vilken bild som ska visas för varje punkt utav JavaScriptet.

DATABAS

I början av projektet gjordes en övergripande databas design som sedan implementerades och första sidan utvecklades med denna databas. Efter att ett flertal nya funktioner behövdes har databasen förändrats och det har krävt stora förändringar för JavaScriptkoden.

PROBLEM & LÖSNING

Då mobiltelefoner använder touch för styrning och användarnas fingrar inte är lika precisa som en muspekare så får informationsrutorna problem när de har scrollbar. Då både scrollbar och bakgrunden känner av förflyttning vilket ofta resulterar i en, till känslan, slumpvis scrolling av antingen scrollbar eller bakomliggande karta så blir detta ett problem. Detta har resulterat i förändringar i CSS för att ta bort behovet av scrollbar vid användande av mobil enhet.

ADMINISTRERING

CSS

Designen som har valts här är en simpel design där allt är centrerat och har en ram runt sig. Alla former som behöver fyllas i lyser grönt om fältet är ifyllt och rött om det inte är ifyllt. Fieldset används som en ram runt alla former för att lättare se vad som hör till vad.

INLOGGNING

Inloggningssidan kollar först om det är en accepterad användare sedan skickar den användaren vidare till administreringssidan.

HTML

HTML koden för denna flik innehåller en ”Form” för att kunna skriva in sitt användarnamn samt lösenord.

PHP

PHP-scriptet för denna sida kopplar först till databasen sedan kontrollerar den om användarnamnet och lösenordet stämmer överens med de som har matats in. Den krypterar först det inmatade lösenordet för att sedan kolla om det är samma krypterade lösenord som finns i databasen. Detta används filen checklogin för. Den sätter sedan alla sessionsvariabler från databasen som stämmer överens med användarnamnet (förnamn, efternamn, mobil, mejl, användarnamn och om man är admin sätts en 1 annars 0).

Funktionen ”password_hash” används för kryptering och ”password_verify” för att verifiera lösenordet.

På alla andra sidor kollas sessionsvariablerna, om de är fel kastas man ut till inloggningen igen.

DATABAS

Första designen av databasen behövde ej editeras under projektets gång. Den enda påverkan på databasen är lösenordet för att kunna identifiera användaren. All information ifrån tabellen kontaktperson tas fram.

FÖRSTA SIDAN

På första sidan för administrering hittas en överblick över alla kontrakt som användaren äger, all information visas i en tabell.

HTML

HTML koden för denna flik innehåller en ”Form” för att visa all information och en ”dropdown” meny för att välja kontrakt.

PHP

PHP-scriptet för denna sida kopplar först till databasen för att hämta informationen sedan skriver den ut alla labels och texttrutor. En funktion används här för att spara det valda kontraktet i ”*dropdown*” menyn så att den inte ändras till standardvärdet. Kontrakten behöver sparas för att sedan att man sedan skall kunna använda det SQL-kommandot.

DATABAS

SQL-kommandot har ändrats många gånger för att databasens struktur uppdaterats. SQL-kommandot tar fram all information ifrån tabellerna ”*kontaktperson*”, ”*kontrakt*”, ”*ikontyp*” och ”*faktura*”. Villkoret för ”WHERE” delen sätts in med hjälp av värdet ifrån ”*dropdown*” menyn då ett specifikt kontrakt skall listas. samt sessionsvariabeln ”*username*”.

PROBLEM & LÖSNING

Det som var problematiskt med denna flik var att få till SQL-frågan för att få fram all information som skulle visas. Eftersom flera tabeller används så behövdes flera joinsatser, detta gjorde att frågan behövde vara mer komplex och det var svårt att se om man fick all information eller inte. Första SQL-satsen som gjordes slogs ihop med fel villkor. Detta resulterade i att exempelvis antal stationer räknades dubbelt.

Detta löstes genom att ändra villkoret för join mellan faktura och kontrakt där kolumnen ID i faktura kopplades med ID i kontrakt. Lösningen blev att istället koppla agarid i faktura med ID i kontrakt.

SKAPA

Under fliken skapa skapas användarkonton och kontrakt samt företag, de fungerar på samma sätt.

HTML

HTML koden för denna flik innehåller former för att visa allt som kan läggas till i ett kontrakt. Dropdown menyerna finns för att kunna välja företag samt användare.

PHP

PHP-scriptet för denna sida kollar först om alla fält som behöver fyllas i är ifyllda, om de är de skapas en "SQL-QUERY" som lägger till ett nytt kontrakt och företag respektive användare. Funktionen `mysqli_real_escape_string` används för att göra det svårare för "SQL injections", "empty" används för att kolla om de utvalda fälten är tomma.

DATABAS

Användandet av databasen är det som skiljer sig åt mellan dessa flikar. För skapa kontakt så görs bara en insert query för att sätta in det nya användarkontot. För att skapa kontrakt så skapas företag om användaren har klickat i checkboxen för att skapa ett nytt företag, annars tas värdet från "dropdown" menyn för företag. Användarkontot tas ifrån "dropdown" menyn för användare. Dessa värden används sedan i inser frågan för att skapa kontrakten.

PROBLEM & LÖSNING

Första problemet med att skapa kontrakt var att välja vilka fält som behövdes för att kunna skapa ett kontrakt. Detta löstes med hjälp av "require" i fälten och en koll i PHP för varje fält om det var tomt eller inte (med funktionen "empty").

Ett annat problem var SQL-satserna där PHP skulle avgöra om det behövdes skapa ett nytt företag och/eller ett nytt användarkonto. Eftersom "CASCADE" används samt att vissa fält måste finnas så skapar det vissa problem. Företag och användare måste skapas före kontrakt. Problemet med användare löstes med att flytta skapandet till en annan flik(detta ger också en möjlighet att ha flera kontrakt knutna till en användare, något som vi hade tänkt på i databasen men inte när PHP-skriptet kodats).

Företag fick vara kvar i samma flik som kontrakt där man kan skapa bäggedera, vi löste detta med en koll: när en checkbox är ifylld så skapas ett nytt företag annars används företaget i dropdown menyn. SQL-satsen blev lite besvärligare då "auto-increment" används i databasen. Vi behövde ett ID för kontrakt så att skapandet av en ny adress skulle fungera. Det löstes med hjälp av SQL-funktionen "LAST_INSERT_ID" vilket ger det senaste ID:et som blivit inmatat.

FAKTURA

Denna sida används för att visa, ta bort samt ladda upp fakturor.

HTML

HTML koden för denna flik innehåller två *"Forms"* en för att visa och ta bort fakturor och en för att ladda upp fakturor. En dropdown meny finns för att välja kontrakt för visning samt för uppladdning.

PHP

PHP-scriptet för denna sida laddar in alla kontrakt i dropdown menyn för att välja faktura. När man trycker på knappen listas alla fakturor för det valda kontraktet med en URL för att komma till en PDF (fakturan) samt en checkbox som används för att ta bort fakturan.

För att ladda upp fakturor används *"move_uploaded_file"* för att flytta filen till servern.

DATABAS

Det finns tre användningsområden för databasen i denna flik uppladdning, borttagning och visning av fakturor. För uppladdning så skapas med hjälp av SQL- kommandot *"INSERT"* en ny rad i tabellen faktura där URL och namn skapas med hjälp av PHP.

För borttagning så används SQL-kommandot *"DELETE"* där villkoret sätts med hjälp av URL:en. Den väljs med hjälp av PHP där värdet tas ifrån checkboxarna som är kopplade till fakturorna.

För visning används SQL-kommandot *"SELECT"* där man listar upp alla fakturor som hör till användaren.

PROBLEM & LÖSNING

Det blev problem med CSS:en då en *"fullösning"* hade använts som skickade båda formulären åt vänster för att sedan skicka tillbaka dem för centrerung. Detta löstes med att använda *"inline-block"* och *"float left"*

FELRAPPORTERING

Denna sida används för att skapa och visa felmeddelanden.

HTML

HTML koden för denna flik har olika innehåll för användare och admins. Admin kan inte skapa felmeddelande men kan ändra status när de svarar på felmeddelande. Användare kan skapa nya felmeddelande men inte ändra status. Det används *"Form"* för att visa och spara information.

PHP

Först så tas alla felmeddelanden fram ifrån databasen som hör till användaren och visas i en dropdown meny. Skriptet kollar om ett felmeddelande är valt eller om ett nytt felmeddelande skall skapas. Sedan skriver det ut en *"Form"* för inmatning av meddelandet.

DATABAS

För att skicka meddelanden så används SQL-kommandot *"INSERT"* där en ny rad i tabellen felmeddelande skapas med hjälp av PHP. För att skicka ett nytt meddelande så skapas ett nytt ämne. Om man svarar på ett meddelande används samma ämne som rubrik.

För borttagning så används SQL-kommandot *"DELETE"* där villkoret sätts med hjälp av informationen ur dropdown menyn.

För visning används SQL-kommandot *"SELECT"* där man listar upp alla felmeddelandet som hör till användaren.

RADERA

Denna sida används för att ta bort kontrakt, företag och användare. Endast administratörer kan komma åt denna sida. Det finns två olika flikar där man raderar antingen *"Kontrakt och Företag"* eller *"Användare"*, de fungerar på lika sätt.

HTML

HTML koden för dessa flikar innehåller två dropdown menyer för att välja vad som skall tas bort.

PHP

Skriptet kollar först om det finns ett värde för det som skall tas bort är valt annars visas ett varningsmeddelande. Om värdet är valt så visas ett varningsmeddelande som frågar om man verkligen vill ta bort det valda innehållet. Knapparna Ja och Nej används för att svara på förfrågan.

DATABAS

För borttagning så används SQL-kommandot *"DELETE"* där villkoret sätts med hjälp av informationen ur dropdown menyn. När ett företag tas bort tas även alla kontrakt som är kopplade till företaget bort, samma gäller även för användare. Detta sker med hjälp av *"CASCADE"*.

För visning används SQL-kommandot *"SELECT"* där man listar upp information om företaget, kontrakt samt användare ifrån databasen.

EDITERA

Denna sida används för att editera databasen. För användare skapas en förfrågan till administratören som måste godkänna denna för att editeringen skall gå igenom.

HTML

HTML koden för alla flikar under editera använder sig av en *"form"* för att ändra informationen i databasen.

PHP

Skriptet skriver först ut all information i en dropdown meny där man kan välja vilken information som skall editeras beroende på vilken flik som väljs. När editeringen ska slutföras kollar skriptet om det är en användare eller administratör som gör förändringen.

DATABAS

För uppdatering så används SQL-kommandot *"UPDATE"* där villkoret sätts med hjälp av informationen ur formuläret. Sedan beroende på om det är en användare eller administrator så skapas ett meddelande till administratören (med hjälp av *"INSERT"*) eller så görs uppdateringen.

För visning används SQL-kommandot *"SELECT"* där man listar upp information som ska editeras.

MEDDELANDE

Denna sida används för att visa meddelanden som har skapats av användarna. Fliken "meddelande" lyser rött om det skulle finnas olästa meddelanden.

HTML

HTML koden använder sig av en *"form"* för att visa all information. Det finns *"nuvarande värde"* samt *"Nytt värde"* där administratören ser vad användaren vill byta för information.

PHP

Skriptet skriver först ut alla meddelanden i en dropdown meny där man kan välja vilket meddelande som skall godkännas eller nekas. När Administratören har valt ett meddelande så skrivs alla förändrade samt nuvarande värden ut så att man kan få en enkel överblick på vad som skall förändras. Skulle förändringen nekas så skrivs ett meddelande tillbaka till användaren där administratören förklarar varför ändringen nekades.

DATABAS

När administratören väljer ett meddelande så används SQL-kommandot *"SELECT"*. När ett meddelande är valt och det skall godkännas eller nekas så används SQL-kommandona *"INSERT"* och *"UPDATE"*. Endast *"UPDATE"* används om det är nekat för att ändra status medan om det är godkänt så används även *"INSERT"* för att lägga in den nya förändringen i databasen.

Skulle administratören vilja ta bort ett meddelande så används SQL-kommandot *"DELETE"* som raderar meddelandet från databasen.

ARBETSMETODER

TEKNISKA

Apache används som webserver för exekvering av alla serverscript.

Github har använts för versionshantering samt delning av filer för att få bra kontroll över dem och slippa trassel med olika versioner av filer.

ORGANISATORISKA

Samarbetet har varit bra i projektgruppen. Den största delen av tiden har spenderats tillsammans i projektrummet, vilket har underlättat samarbetet i gruppen då informationsflödet mellan medlemmar varit direkt och fördröjningar minimerats. Scrum har också använts för att underlätta och snabba på fördelningen av arbetsuppgifter.

PLANERING OCH UPPFÖLJNING

För att underlätta planering har en grundläggande mall över projektet tagits fram. Denna mall har använts för att skapa programmet och gett en insikt på vad som systemet begär. Scrum har använts för att lättare hålla koll på vad som skall göras, de större delarna har skrivits in i en backlogg.

För att underlätta uppföljning har en tidsrapport gjorts som visar vilka personer som har gjort vad. Daily Scrum har också varit en del av uppföljningen under projektets gång då man har fått större insyn i vad alla i gruppen har gjort och hur det går.

ERFARENHETER

I detta projekt har erfarenheterna för hur viktigt det är att strukturera och planera ett projekt. I detta fall har Scrum använts vilket visade sig vara en bra utvecklingsmetod. Erfarenheterna av versionshanteringssystem med hjälp av GIT har också ökat

BESKRIVNING AV SYSTEMET

TEKNISK BESKRIVNING

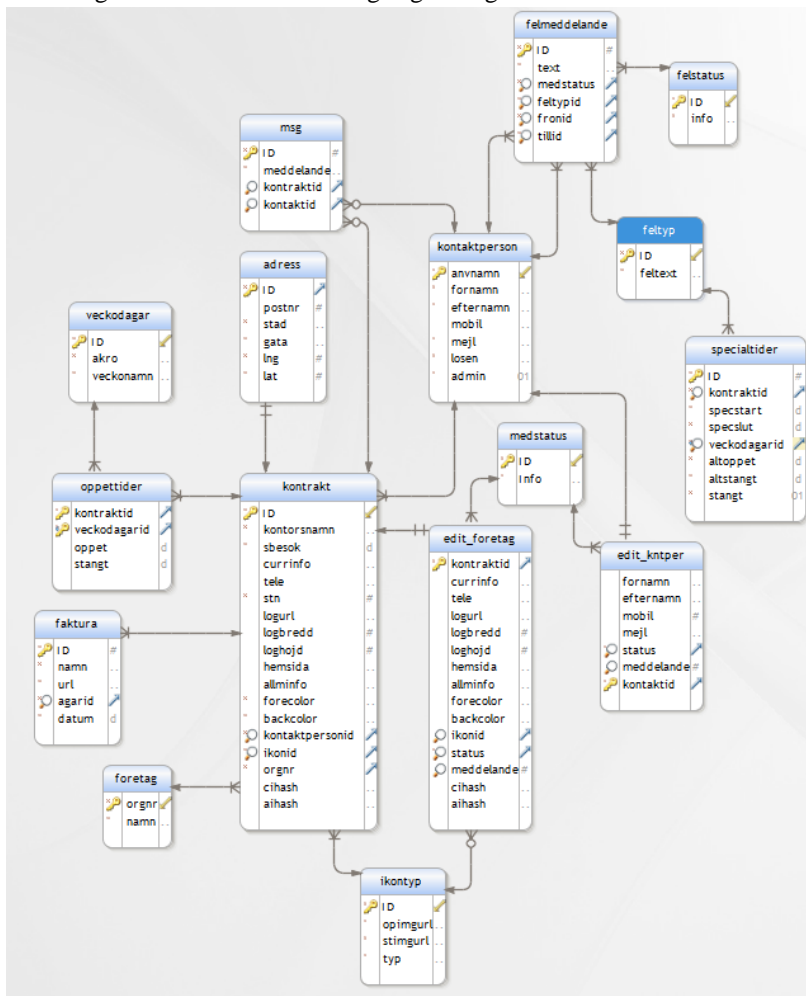
Ett plattformsoberoende system har utvecklats med HTML5 för att underlätta kundens användning av systemet. Genom att ha ett plattformsoberoende system behöver kunden inte införskaffa sig olika program för olika plattformar(ex: Android, IOS osv.).

Den mjukvara som har utvecklats har gjorts i PHP, HTML5, MySQL, CSS, JQuery och JavaScript.

DESIGN

DATABAS

Designen för databasen har förändrats över tiden då ett flertal nya funktioner har lagts till, ändrats eller tagits bort. I figur 1 kan man se den slutgiltiga designen över databasen i ett ER-diagram med kråkfots notation.



Figur 1: ER-diagram utav databasen

USE CASES

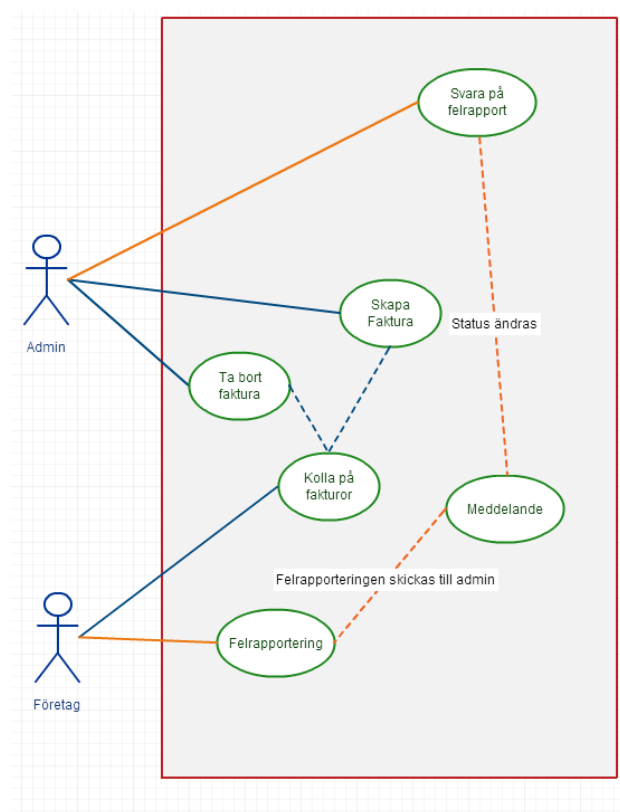
För att lättare kunna förklara vad hemsidan gör så har use cases använts. I efterföljande text används "Admin" som en beskrivning av huvudanvändaren på hemsidan. "Admin" syftar således på en användare som har full kontroll över hemsidans alla funktioner, lämpligen en anställd på Mobow. I efterföljande text refereras det till "företag" och syftar på ett användarkonto som varje kund hos Mobow givits. Dessa kunder skall inte förväxlas med slutanvändare som utnyttjar laddningsstationen utan refererar till de företag/organisationer som hyr/köpt/lånat en eller flera laddningsstationer ifrån Mobow. I efterföljande text refereras även till "kund" och dessa syftar på de användare som utnyttjar Mobows laddningsstationen för att ladda deras handhållna enheter.

FELRAPPORTERING

För felrapportering så skapar företaget en felrapport sen skickas ett meddelande till Admin. Admin kan svara på detta meddelande och ändra status(Mottagen, Påbörjad eller Avslutad) sedan skickas ett meddelande tillbaks till företaget. Företaget kan också svara på detta meddelande. För grafisk use case se figur 2.

FAKTUROR

Admin kan ladda upp och ta bort fakturor som sedan företaget kan kolla på. För grafisk use case se figur 2.



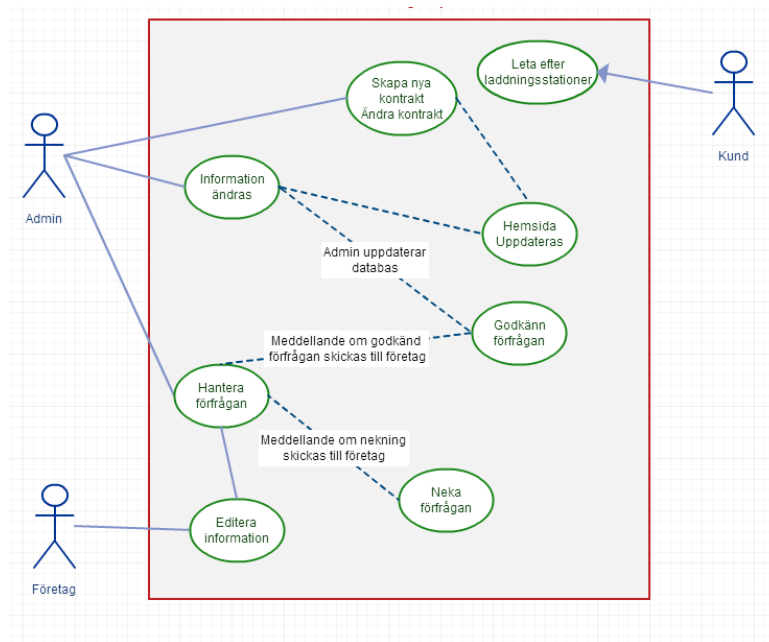
Figur 2: use case för felrapportering och fakturor

EDITERING AV INFORMATION

Editeringen av information i databasen från företagets sida sker som följande. Först skickar företaget en förfrågan till Admin. Admin kan godkänna eller neka förfrågan, om han godkänner den så uppdateras databasen.

Admin kan editera informationen som han vill i databasen.

När informationen har uppdaterats så kommer hemsidan att uppdateras och kunden få tillgång till den nya informationen. För grafisk use case se figur 3.



Figur 3: use case för editering av information

GRAFISK DESIGN

Valet av designen grundades på färgkoderna ifrån Mobows hemsida (<http://mobow.se/>). En lättarbetad hemsida prioriterades, där oerfarna datoranvändare samt mer erfarna skulle kunna hantera och administrera hemsidan. Därför valdes det att administreringen av hemsidan bara skulle innehålla lätthanterliga former för editering samt information av text. Det skulle inte vara onödiga effekter som skulle göra hemsidan mer tilltalande, men kanske svårare att hantera. Funktionalitet prioriterades också framför design.

SÄKERHET

För att få en grundläggande säkerhet har vissa koddesigner och domändesigner använts. Eftersom databasen lagras i klartext så ligger den utanför roten för hemsidan och kan därför ej komma åt utifrån. Kommandot `"mysql_real_escape_string"` har använts flitigt för att förebygga hemsidan mot eventuella "SQL-Injections".

Vissa säkerhetsspärrar har lagts till för administreringen av kontrakt och användarkonton för företag. Då varje gång de försöker uppdatera någon skickas en förfrågan till admin först. Detta implementerades för att företag inte skall kunna lägga upp felaktig eller stötande information.

ERFARENHETER

Erfarenheterna av språken PHP, CSS, HTML, JavaScript samt SQL har ökat väldigt då majoriteten av gruppen inte hade programmerat med dessa innan början av kursen.

DISKUSSION

Den första kravspecifikationen var väldigt simpel, en mobilapplikation skulle göras. Meningen var att samma databas som användes i kunden nuvarande hemsida skulle kopieras till programmet. Det kändes för lite för oss och därför uppdaterades kravspecifikationen till att omfatta en ny hemsida samt en sida för mobil, detta skulle göras i HTML5 för att göra mobilapplikationen plattformsoberoende.

LEVERERADE SYSTEMET

Det levererade systemet uppfyller nästan alla krav i kravspecifikationen förutom beställningsdelen. Som resultat av tidsbrist ansågs det vara lämpligast att avgränsa sig och ta bort den delen då kunden i nuläget inte har någon användning av den.

ARBETSPROCESSEN - HUR VI HAR JOBBAT

I projektets början hade vi inte möjlighet att lägga så många timmar på projektet då det var mitt i en tentaperiod. Därför blev det en långsam start på projektet. De ändringar av kravspecifikationen orsakade också fördröjningar då programmeringen inte kunde påbörjas innan den var bestämd.

Efter kravspecifikationen hade blivit uppdaterad börjades kodningen, vi insåg snabbt att inläring av språken som skulle användas behövdes. Detta orsakade också en liten fördröjning av projektet.

Under den andra hälften av projektet behövdes ingen vidare inläring av språken och verktygen och då tog kodningen fart. Inga större problem har stötts på underprojektet därför har vi kunnat producera så pass mycket kod på relativt liten tid(då majoriteten av gruppen faktiskt var nybörjare på språken).

Man kan märka av att strukturen på koden har ändrats ut eftersom projektets fortlöpning, då det är lättare att få en bra struktur när man kan språket. Detta har orsakat i svårigheter när buggar och nödvändiga förändringar av koden har uppkommit.

GRUPPARBETET

Grupparbetet har varit effektivt genom hela projektet, främst för att vi har för mesta jobbat tillsammans i grupprummet. Alla har fått möjlighet till hjälp från de andra i gruppen. Arbetstilldelningen i projektet har funkat väldigt bra, alla har hjälp till med de flesta flikar i hemsidan vilket resulterar i att man har erfarenhet av de mesta koden i projektet. Detta har resulterat i lättare kopplingar mellan de olika flikarna och större förståelse för hur hemsidan fungerar.

KONTAKT MED BESTÄLLARE

Kontakten med beställaren har varit bra. Vi har kunnat haft möte varje vecka och haft en nära kontakt under arbetet vilket är en viktig del av Scrum.

SLUTSATSER

Öppettider som frångår de normala har inte fullständigt implementerats då tiden var för knapp. I databasen är speciella öppettider implementerade men utnyttjas inte då det inte funnits tid att skapa ett gränssnitt som kan editera denna information. Utöver så saknas även beställningsfliken. Utöver dessa delar så har målen i kravspecifikationen uppfyllts. I nuläget har vi levererat ett fungerande system bortsett från beställningsfunktionen, det största som behöver göras är att få hemsidans grafiska design att se bättre ut för att kunna locka fler kunder då det är inget vi har fokuserat på.

BILAGA

BILAGA 1 – KRAVSPECIFIKATION

PRODUKTEN

Produkten skapas för att få en lösning på det problemet som finns idag för Mobow, att det inte finns någon mobilapplikation som kan hjälpa smartphone-användare att identifiera var Mobows laddningsstationer finns utplacerade i till exempel en stad.

För att en smartphone-användare enkelt och snabbt skall kunna hitta var laddningsstationer finns utplacerade så behövs en mobilapplikation. Applikationen skall enkelt visa var laddningsstationer finns på karta med hjälp av mobiltelefonens GPS-funktion. Applikationen skall vara användarvänlig och tilltalande.

En hemsida skapas för att företag samt Mobow skall kunna hantera data som används i mobilapplikationen och även skapa en kommunikation mellan företag och Mobow för att skapa kontrakt och lägga beställningar

FUNKTIONELLA KRAV

KUND:

- Skall inte vara beroende av plattform
- Det skall finnas ett webbanvändargränssnitt
- Det skall finnas ett mobilanvändargränssnitt
- Kunden skall kunna hitta närmaste laddningsstationer
- Kunden skall se alla företag med laddningsstationer i en stad
- Ikonerna skall vara klickbara
- Genom klick på en ikon skall kund få fram:
 - Bild på företagets logga (med en bestämt max storlek samt anpassningsbar till inforutan)
 - Namn
 - Öppettid
 - Adress
 - Meny/hemsida som skall vara klickbar(länk)
 - Antal stationer
 - Vägbeskrivning som skall vara klickbar(länk)
 - Telefonnummer
 - Allmän info

ADMIN:

- Skall inte vara beroende av plattform
- Skall ha egna inloggningsuppgifter
- Får endast ett webbanvändargränssnitt
- Skall kunna lägga till ikoner
- Skall ha eget ansvar över hemsidan
- Skall kunna Skapa nya företagsanvändare

- Skall kunna editera information
- Skall kunna ändra status på beställning
- Skall kunna se felrapportering
- Skall kunna lägga upp och skicka faktura som PDF

FÖRETAG:

- Skall inte vara beroende av plattform
- Skall ha egna inloggningsuppgifter/kontor
- Får endast ett webbanvändargränssnitt
- Skall kunna lägga beställningar
- Skall kunna editera egen information
- Skall kunna skapa ett nytt kontrakt
- Skall kunna skapa en felrapportering
- Skall kunna se faktura som PDF
- Skall kunna lägga till en företagslogga

ICKE-FUNKTIONELLA KRAV

KUND:

- Det skall vara en simpel och användarvänligt applikation
- Tiden att starta applikationen skall inte överstiga 5 sekunder
- Vyn för kartan skall vara in zoomad till en stads gräns

ADMIN:

- Visningen av editnotis skall vara tydlig
- Hemsidan skall följa Mobows grafiska profil

FÖRETAG:

- Tiden för att företaget skall få en notis om status efter att Mobow ändrat informationen skall inte överstiga 60 sekunder.

ANVÄNDARVÄNLIGHET

- Systemet skall vara intuitivt och applikationen skall vara tillfredsställande och lättförståelig.

KAPACITET

- Serverberoende av antal användare
- Administratörer samt företag skall inte kunna editera samma information samtidigt

UNDERHÅLLBARHET

- Efter levererad produkt kommer vi ej ha ansvar för programmet längre

TILLGÄNGLIGHET

- 90 % (serverberoende samt internetberoende)

LEVERANSVILLKOR

- Klart senast kursens deadline(13 januari)

BILAGA 2 – PROJEKTPLAN

PROJEKTBESKRIVNING

BAKGRUND

Mobow (www.mobow.se) är ett Gävlebaserat företag som är verksam med laddningsstationer för mobiltelefoner. Dessa laddningsstationer kan placeras på publika ställen så som caféer, bibliotek, affärer eller liknande. I dessa laddningsstationer har kunder och besökare möjligheten att ladda sin mobiltelefon samtidigt som de nyttjar de publika ställenas erbjudanden. Med dessa laddningsstationer i närheten av kunden så löser detta problemet med kort batteritid för smartphone-användare samtidigt som laddningsstationerna också hjälper de publika ställena att locka besökare.

SYFTE/UPPGIFT

I dagsläget finns det ingen mobilapplikation som kan hjälpa smartphone-användare att identifiera var Mobows laddningsstationer finns utplacerade. För att en smartphone-användare enkelt och snabbt skall kunna hitta var laddningsstationer finns utplacerade så behövs en mobilapplikation. Detta är vad det föreslagna examensarbetet inriktas på. Uppdraget är att utforska behov och krav kring en mobilapplikation som behandlar geografisk data, samt skapa denna för Android-plattformen. Grundfunktionaliteten är att applikationen enkelt skall visa var laddningsstationer finns på en karta med hjälp av mobiltelefonens GPS-funktion. Utöver detta så kan fler funktioner läggas till. Viktigt är att applikationen är lättanvänd och tilltalande.

MÅL, AVGRÄNSNINGAR OCH VETENSKAPLIGHET

Målet från Mobows sida är främst en kartläggning av behov och krav för ovan beskrivna mobilapplikation samt själva utvecklandet av själva applikationen. Mobow ser dock att Högskolan i Gävles krav på studentprojekt/examensarbeten också uppfylls gällande vetenskaplighet. Detta innebär att ovan mer praktiska uppgift kan kombineras med ett lämpligt vetenskapligt syfte. Detta vetenskapliga syfte utformas i dialog med studenter och lärare vid högskolan. Det är också möjligt att uppgiften är lämplig att genomföras i flera projekt/examensarbeten vilket i så fall innebär en uppdelning av uppgiften i flera delar. Möjliga aktiviteter i denna uppgift är:

- Kartläggning av behov och krav hos slutanvändare
- Kartläggning av behov och krav hos publika ställen
- Utformande av en kravspecifikation
- Idégenerering
- Programmering
- Prototyper
- Testning

ADMINISTRATIVA MÖJLIGHETER FÖR MOBOW SAMT FÖRETAG

För att göra en applikation till mobilen kommer det att behövas en hemsida där företag och Mobow kan gå in och redigera samt skapa vital information. Även kommunikationen mellan Mobow och företag skall kunna gå att genomföra via programmet för att på så vis kunna skapa felrapporter samt lägga beställningar på varor och hantera kontrakt.

Utvecklingsarbetet kan utföras självständigt eller i nära samarbete med Mobow. Mobow är också öppen för studenters egna önskemål kring uppgiften.

BILAGA 3 – MANUAL FÖR PROGRAMMET

INSTALLERING AV HEMSIDA

PROGRAMVARA

Programvaran som används för att driva hemsida behöver klara av PHP v5.5.0 och MySQL 5.1 eller nyare. Server med UTF8 som teckenuppsättning.

DATABAS

De kommandon som krävs för att skapa databasen ligger under bilagor(se bilaga Databas).

INLOGGNING

För att logga in första gången har ett konto skapats med användarnamnet AdminM och lösenord Mobow, inloggningen kan ses i figur 1.

INDEX SIDAN

KARTA

På första sidan hittas kartan, där kan man se alla restauranger samt caféer. Deras öppettider representeras av färger där rött betyder stängt och grönt betyder öppet. Om man klickar på ikonerna för platsen så öppnas en ruta där man kan hitta ytterligare information som adress, allmän information, specifika öppettider, antal stationer, vägbeskrivning samt en länk till deras hemsida.

ADMINISTRERING

Första sidan innehåller all relevant information om dina kontrakt(alla kontrakt om man är inloggad som admin) (se figur 2).

SKAPA

För att skapa nya kontrakt så används fliken ”Skapa”, här väljer du mellan att skapa ett nytt kontrakt eller skapa en ny användare som man senare kan länka till ett nytt kontrakt. Denna flik kan bara komma åt med konton som är Admin.

NY ANVÄNDARE

Under fliken ”Användare” kan du skapa nya användare, alla fält som är röda måste fyllas i och gröna är valfria. I rutan ”Administratör” kan man välja om användaren skall vara Admin eller inte(se figur 3).

NYTT KONTRAKT

Här skapas ett nytt kontrakt, alla fält som är röda måste fyllas i och alla gröna fält är valfria.

- Man kan skapa ett nytt företag genom att klicka i rutan "Nytt Företag", annars väljs företaget i dropdown menyn under "Välj Företag".
- Man får välja användare under "Välj Användare" dessa måste skapas separat under fliken skapa användare (se figur 4).

FAKTURA

Under fliken faktura kan du ladda upp fakturor samt ta bort fakturor.

Som användare kommer en lista kunna visas på de fakturor som är kopplade till det kontraktet. Som admin kommer en lista på alla kontrakt att kunna visas, dock listas endast ett kontrakt i taget.

UPPLADDNING

Uppladdningen av faktura görs genom att fylla i alla röda fält, sedan välja faktura(PDF-format) på hårddisken för uppladdning till server. I dropdown menyn väljs det vilket företag som fakturan kommer att tillhöra(se figur 5).

BORTTAGNING

Borttagningen av fakturor görs genom att först läsa in alla fakturor, detta görs genom att välja ett kontrakt och sedan klicka på "Välj Kontrakt". Rutan fylls då med alla fakturor som tillhör detta kontrakt. Man kan klicka i en ruta till höger om länken till fakturan, sen kan fakturan tas bort genom att klicka på "Ta bort Fakturor"(se figur 5).

FELRAPPORTERING

Felrapporteringen mellan kund och Admin sker genom fliken felrapportering, när Admin har fått in en ny felrapport kommer fliken felrapportering lysa rött. Alla statusar har olika färgkoder för felmeddelanden de är:

- Röd: oläst
- Orange: mottagen
- Gul: påbörjad
- Grön: avslutad

SKAPA

Kund kan skicka en felrapport genom att klicka på välj när "Välj Felrapport" är vald i dropdown menyn(admin kan inte skapa nya felrapporter utan bara svara)(se figur 6).

SVARA

Man kan svara på felrapporter genom att välja felrapporten i dropdown menyn sedan får man skriva in ett meddelande, Admin kan ändra statusen på felrapporten(se figur 7).

TA BORT

Man kan ta bort felrapporter genom att välja en felrapport och sedan klicka på ”ta bort”-knappen.

RADERA

Under fliken ”Radera” kan man radera kontrakt, företag samt kontakter. Dessa flikar har endast admin tillgång till.

KONTRAKT OCH FÖRETAG

Under fliken ”Kontrakt och företag” kan man ta bort företag och kontrakt genom att välja det man vill ta bort i dropdown menyerna och sedan trycka på ”Ta Bort Kontrakt” för att ta bort kontrakt och ”Ta bort Företag” för att ta bort företag. Om ett företag tas bort tas även alla kontrakt som tillhör företaget bort.

ANVÄNDARE

Underfliken ”Radera Användare” kan man ta bort användare genom att välja dem i dropdown menyn ”Välj Användare”. Det går även att sortera på företag. Genom att ta bort användare tas även kontraktet som är knutet till användaren bort.

EDITERA

Under fliken ”Editera” kan man editera inloggning, öppettider samt kontrakt.

INLOGGNING

Man kan editera sitt lösenord genom att först skriva in sitt gamla lösenord och sedan skriva in det nya lösenordet två gånger. Admin kan ändra lösenord för alla användare(se figur 8).

ÖPPETTIDER

Man kan editera öppettider genom att välja sitt kontrakt och sedan fylla i när de öppnar samt stänger, man kan även fylla i en ruta om det är stängt(se figur 9). Admin kan editera alla kontrakt.

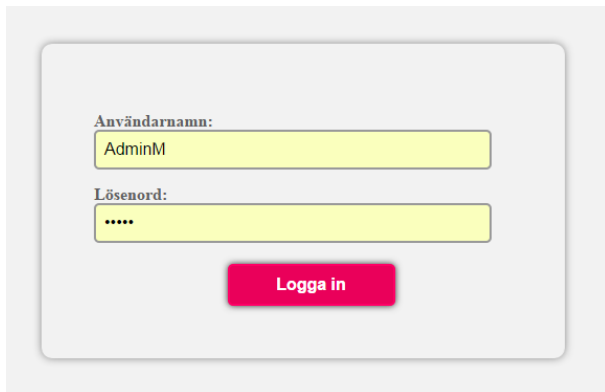
KONTRAKT

Man kan editera kontrakt genom att först välja kontrakt och sedan fylla i de flikar som skall editeras, ny bild kan laddas upp om man trycker på ”Välj fil”(se figur 10). Ändringarna sparas när man trycker på ”Spara”. Admin kan editera alla kontrakt medan kundernas ändringar skickas till Admin först.

BILAGOR

BILDER

INLOGGNING



A login form for administration. It features a light gray background with a white rounded rectangle containing the form elements. The form has two text input fields: the first is labeled 'Användarnamn:' and contains the text 'AdminM'; the second is labeled 'Lösenord:' and contains five dots. Below the password field is a red button with the text 'Logga in'.

Figur 4

FÖRSTA SIDAN FÖR ADMINISTRERING

Karl Lundh, Gevalia Norr

Välj kontakt

Information

Företag:
Rubinola AB

Kontor:
Gevalia Norr

Adress:
Stortorget 1, 80320 Gävle

Organisationsnummer:
225454-3458

Kontorstyp:
Restaurang

Hyr Antal Stationer:
3

Senaste Besök:
2014-06-04

Nästa Besök:
2014-12-04

Kontaktperson:
Karl Lundh

Användarnamn:
KarlL

Telefon:
345678912

Mejl:
nbt11nsn@student.hig.se

Övrig Information

Hemsidan:
[Gå Till Hemsidan](#)

Faktura:
[Öppna Senaste Fakturan](#)

Figur 5

SKAPA ANVÄNDARE

Användare

- Användarnamn: (minst ett icke-numeriskt tecken)
- Förnamn:
- Efternamn:
- Mobilnummer:
- Mejl:
- Lösenord:

Administrator: ☐

Återställ

Spara

Figur 6

SKAPA KONTRAKT

Villj Företag

Villj Användare

Nytt Företag

Företagsnamn:

Organisationsnummer:

Nytt Företag:

☐

Kontor

Namn:

Senaste besök:

Telefon:

Antal Stationer:

Hemsida (som börjar http://):

Information:

Aktivitet:

Förgrundsfärg:

Bakgrundsfärg:

Närvarande Bild: Ingen bild vald

Villj Bild:

 Ingen bild har valts

Bostyp:

Adress

Postnummer:

Stad:

Gata:

Longitude:

Latitude:

Använda
Spara

Figur 7

SKAPA OCH TA BORT FAKTUROR

The screenshot shows a web application interface for managing invoices. It is divided into two main sections. The left section, titled "Ladda upp Faktura", contains form fields for "Namn på Fakturan:", "Fakturan Tillhör:", "Datum:", and a file upload area "Ladda upp en Faktura". The right section, highlighted with a red border, contains a dropdown menu for "Solen", a "Valj Kontrakt" button, a selected contract "faktura4 2013-07-16", and a red "Ta bort fakturor" button.

Figur 8

FELRAPPORTERING

Ny Felrapport

Välj
Ta bort
Skicka

Från:

AndersB

Ämne:

Meddelande:

Figur 9

AndersB Kabelbrott Mottagen

Välj
Ta bort
Skicka

Från:

AndersB

Status:

Mottagen

Ämne

Kabelbrott

Meddelande:

sdsdfhsdfhsdfh

Status

Mottagen

Nytt Meddelande:

Figur 10

EDITERING AV LÖSENORD

Uppdatera Ditt Lösenord

Gammalt Lösenord:

Nytt Lösenord:

Upprepa Nytt Lösenord:

Uppdatera

Byt lösenord på användare

AndersB (Anders Blomkvist)

Admin Lösenord:

Nytt Lösenord:

Upprepa Nytt Lösenord:

Byt

Figur 11

EDITERING AV ÖPPETTIDER

Välj Organisationsnummer
Gevalia Söder
Välj kontrakt

Söndag:

Öppnar: 23:24 Stänger: 12:34 Stängt:

Måndag:

Öppnar: 05:45 Stänger: 12:25 Stängt:

Tisdag:

Öppnar: 12:25 Stänger: 12:36 Stängt:

Onsdag:

Öppnar: 06:00 Stänger: 20:00 Stängt:

Torsdag:

Öppnar: --:-- Stänger: --:-- Stängt: ✓

Fredag:

Öppnar: 11:23 Stänger: 13:25 Stängt:

Lördag:

Öppnar: 05:59 Stänger: 05:12 Stängt:

Återställ Spara

Figur 12

EDITERING AV KONTRAKT

Välj Organisationsnummer
Gevalia Norr (Stortorget 1)

Välj kontrakt

Namn:
Gevalia Norr

Senaste Besök:
2014-06-04

Typ av Verksamhet:
Restaurang

Telefon:
026-78458

Antal Stationer:
3

Hemsida (kom ihåg http://):
http://www.google.com/

Information:
random grejs

Aktuellt:

Förgrundsfärg:

Bakgrundsfärg:

Postnummer:
80320

Stad:
Gävle

Gata:
Stortorget 1

Nuvarande Bild:  Ta Bort Bild

Byt Bild (jpeg,bmp,gif,png <2MB):
Välj fil Ingen fil har valts

Återställ Spara

Figur 13

BILAGA 4 – DATABAS MOBOWDB

TABELLSTRUKTUR ADRESS

| Kolumn | Typ | Null | Standardvärde |
|-----------|--------------|------|---------------|
| ID | int(11) | Nej | |
| postnr | int(11) | Ja | NULL |
| stad | varchar(100) | Nej | |
| gata | varchar(100) | Nej | |
| lng | double | Nej | |
| lat | double | Nej | |

TABELLSTRUKTUR EDIT_FORETAG

| Kolumn | Typ | Null | Standardvärde |
|-------------------|--------------|------|---------------|
| currinfo | text | Ja | NULL |
| cihash | varchar(64) | Ja | NULL |
| tele | varchar(20) | Ja | NULL |
| logurl | varchar(256) | Ja | NULL |
| logbredd | int(11) | Ja | NULL |
| loghojd | int(11) | Ja | NULL |
| hemsida | varchar(256) | Ja | NULL |
| allminfo | text | Ja | NULL |
| aihash | varchar(64) | Ja | NULL |
| forecolor | varchar(7) | Ja | NULL |
| backcolor | varchar(7) | Ja | NULL |
| ikonid | int(11) | Ja | NULL |
| status | int(11) | Nej | 1 |
| meddelande | int(11) | Ja | NULL |
| kontraktid | int(11) | Nej | |

TABELLSTRUKTUR EDIT_KNTPER

| Kolumn | Typ | Null | Standardvärde |
|------------------|-------------|------|---------------|
| fornamn | varchar(50) | Ja | NULL |
| efternamn | varchar(50) | Ja | NULL |
| mobil | int(11) | Ja | NULL |
| mejl | varchar(50) | Ja | NULL |
| status | int(11) | Nej | |
| meddelande | int(11) | Ja | NULL |
| kontaktid | varchar(50) | Nej | |

TABELLSTRUKTUR FAKTURA

| Kolumn | Typ | Null | Standardvärde |
|-----------|---------|------|---------------|
| ID | int(11) | Nej | |

| | | |
|--------|--------------|-----|
| namn | varchar(50) | Nej |
| url | varchar(100) | Nej |
| agarid | int(11) | Nej |
| datum | date | Nej |

TABELLSTRUKTUR FELMEDDELANDE

| Kolumn | Typ | Null | Standardvärde |
|-----------|-------------|------|---------------|
| ID | int(11) | Nej | |
| text | text | Nej | |
| medstatus | int(11) | Nej | 1 |
| feltypid | int(11) | Nej | |
| fronid | varchar(50) | Nej | |
| tillid | varchar(50) | Nej | |

TABELLSTRUKTUR FELSTATUS

| Kolumn | Typ | Null | Standardvärde |
|-----------|-------------|------|---------------|
| ID | int(11) | Nej | |
| info | varchar(50) | Nej | |

TABELLSTRUKTUR FELTYP

| Kolumn | Typ | Null | Standardvärde |
|-----------|--------------|------|---------------|
| ID | int(11) | Nej | |
| feltext | varchar(100) | Nej | |

TABELLSTRUKTUR FORETAG

| Kolumn | Typ | Null | Standardvärde |
|--------------|-------------|------|---------------|
| orgnr | varchar(20) | Nej | |
| namn | varchar(50) | Nej | |

TABELLSTRUKTUR IKONTYP

| Kolumn | Typ | Null | Standardvärde |
|-----------|--------------|------|---------------|
| ID | int(11) | Nej | |
| opingurl | varchar(256) | Nej | |
| stimgurl | varchar(256) | Nej | |
| typ | varchar(100) | Nej | |

TABELLSTRUKTUR KONTAKTPERSON

| Kolumn | Typ | Null | Standardvärde |
|----------------|-------------|------|---------------|
| anvnamn | varchar(50) | Nej | |
| fornamn | varchar(50) | Nej | |
| efternamn | varchar(50) | Nej | |
| mobil | varchar(20) | Ja | NULL |
| mejl | varchar(50) | Nej | |

losen varchar(256) Nej
admin tinyint(1) Nej

TABELLSTRUKTUR KONTRAKT

| Kolumn | Typ | Null | Standardvärde |
|-----------------|--------------|------|---------------|
| ID | int(11) | Nej | |
| kontorsnamn | varchar(50) | Nej | |
| sbesok | date | Nej | 0000-00-00 |
| currinfo | text | Ja | NULL |
| cihash | varchar(64) | Ja | NULL |
| tele | varchar(20) | Ja | NULL |
| stn | int(11) | Nej | |
| logurl | varchar(256) | Ja | NULL |
| logbredd | int(11) | Ja | NULL |
| loghojd | int(11) | Ja | NULL |
| hemsida | varchar(256) | Ja | NULL |
| allminfo | text | Ja | NULL |
| aihash | varchar(64) | Ja | NULL |
| forecolor | varchar(7) | Nej | #000000 |
| backcolor | varchar(7) | Nej | #FFFFFF |
| kontaktpersonid | varchar(50) | Nej | |
| ikonid | int(11) | Nej | |
| orgnr | varchar(20) | Nej | |

TABELLSTRUKTUR MEDSTATUS

| Kolumn | Typ | Null | Standardvärde |
|-----------|-------------|------|---------------|
| ID | int(11) | Nej | |
| Info | varchar(50) | Nej | |

TABELLSTRUKTUR MSG

| Kolumn | Typ | Null | Standardvärde |
|------------|-------------|------|---------------|
| ID | int(11) | Nej | |
| meddelande | text | Nej | |
| kontraktid | int(11) | Ja | NULL |
| kontaktid | varchar(50) | Ja | NULL |

TABELLSTRUKTUR OPPETTIDER

| Kolumn | Typ | Null | Standardvärde |
|---------------------|---------|------|---------------|
| kontraktid | int(11) | Nej | |
| veckodagarid | int(11) | Nej | |
| oppet | time | Ja | NULL |
| stangt | time | Ja | NULL |

TABELLSTRUKTUR SPECIALTIDER

| Kolumn | Typ | Null | Standardvärde |
|--------------|------------|------|---------------|
| ID | int(11) | Nej | |
| kontraktid | int(11) | Nej | |
| specstart | date | Nej | |
| specslut | date | Nej | |
| veckodagarid | int(11) | Nej | |
| altoppet | time | Nej | |
| altstangt | time | Nej | |
| stangt | tinyint(1) | Nej | 1 |

TABELLSTRUKTUR VECKODAGAR

| Kolumn | Typ | Null | Standardvärde |
|-----------|------------|------|---------------|
| ID | int(11) | Nej | |
| akro | varchar(4) | Nej | |
| veckonamn | varchar(8) | Nej | |

BILAGA 5 – SQL

```

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
--
-- Databas: `mobowdb`
--
CREATE DATABASE IF NOT EXISTS `mobowdb` DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci;
USE `mobowdb`;

```

```

-----

```

```

--
-- Tabellstruktur `adress`
--

```

```

DROP TABLE IF EXISTS `adress`;
CREATE TABLE IF NOT EXISTS `adress` (
  `ID` int(11) NOT NULL,
  `postnr` int(11) DEFAULT NULL,
  `stad` varchar(100) NOT NULL,
  `gata` varchar(100) NOT NULL,
  `lng` double NOT NULL,
  `lat` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=12 ;

```

```

-----

```

```

--
-- Tabellstruktur `edit_foretag`
--

```

```

DROP TABLE IF EXISTS `edit_foretag`;
CREATE TABLE IF NOT EXISTS `edit_foretag` (
  `currinfo` text,
  `cihash` varchar(64) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
  `tele` varchar(20) DEFAULT NULL,
  `logurl` varchar(256) DEFAULT NULL,
  `logbredd` int(11) DEFAULT NULL,
  `loghojd` int(11) DEFAULT NULL,
  `hemsida` varchar(256) DEFAULT NULL,
  `allminfo` text,
  `aihash` varchar(64) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
  `forecolor` varchar(7) DEFAULT NULL,
  `backcolor` varchar(7) DEFAULT NULL,
  `ikonid` int(11) DEFAULT NULL,
  `status` int(11) NOT NULL DEFAULT '1',
  `meddelande` int(11) DEFAULT NULL,
  `kontraktid` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-----

```

```

--
-- Tabellstruktur `edit_kntper`

```

--

```
DROP TABLE IF EXISTS `edit_kntper`;
CREATE TABLE IF NOT EXISTS `edit_kntper` (
  `fornamn` varchar(50) DEFAULT NULL,
  `efternamn` varchar(50) DEFAULT NULL,
  `mobil` int(11) DEFAULT NULL,
  `mejl` varchar(50) DEFAULT NULL,
  `status` int(11) NOT NULL,
  `meddelande` int(11) DEFAULT NULL,
  `kontaktid` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

--

-- Tabellstruktur `faktura`

--

```
DROP TABLE IF EXISTS `faktura`;
CREATE TABLE IF NOT EXISTS `faktura` (
  `ID` int(11) NOT NULL,
  `namn` varchar(50) NOT NULL,
  `url` varchar(100) NOT NULL,
  `agarid` int(11) NOT NULL,
  `datum` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

--

-- Tabellstruktur `felmeddelande`

--

```
DROP TABLE IF EXISTS `felmeddelande`;
CREATE TABLE IF NOT EXISTS `felmeddelande` (
  `ID` int(11) NOT NULL,
  `text` text NOT NULL,
  `medstatus` int(11) NOT NULL DEFAULT '1',
  `feltypid` int(11) NOT NULL,
  `fronid` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,
  `tillid` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

--

-- Tabellstruktur `felstatus`

--

```
DROP TABLE IF EXISTS `felstatus`;
CREATE TABLE IF NOT EXISTS `felstatus` (
  `ID` int(11) NOT NULL,
  `info` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
```

```
--
-- Tabellstruktur `feltyp`
--

DROP TABLE IF EXISTS `feltyp`;
CREATE TABLE IF NOT EXISTS `feltyp` (
  `ID` int(11) NOT NULL,
  `feltext` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;

-----

--
-- Tabellstruktur `foretag`
--

DROP TABLE IF EXISTS `foretag`;
CREATE TABLE IF NOT EXISTS `foretag` (
  `orgnr` varchar(20) NOT NULL,
  `namn` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----

--
-- Tabellstruktur `ikontyp`
--

DROP TABLE IF EXISTS `ikontyp`;
CREATE TABLE IF NOT EXISTS `ikontyp` (
  `ID` int(11) NOT NULL,
  `opingurl` varchar(256) NOT NULL,
  `stimgurl` varchar(256) NOT NULL,
  `typ` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;

-----

--
-- Tabellstruktur `kontaktperson`
--

DROP TABLE IF EXISTS `kontaktperson`;
CREATE TABLE IF NOT EXISTS `kontaktperson` (
  `anvnamn` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,
  `fornamn` varchar(50) NOT NULL,
  `efternamn` varchar(50) NOT NULL,
  `mobil` varchar(20) DEFAULT NULL,
  `mejl` varchar(50) NOT NULL,
  `losen` varchar(256) NOT NULL,
  `admin` tinyint(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- Tabellstruktur `kontrakt`
```

```
--
```

```
DROP TABLE IF EXISTS `kontrakt`;
CREATE TABLE IF NOT EXISTS `kontrakt` (
  `ID` int(11) NOT NULL,
  `kontorsnamn` varchar(50) NOT NULL,
  `sbesok` date NOT NULL DEFAULT '0000-00-00',
  `currinfo` text,
  `cihash` varchar(64) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
  `tele` varchar(20) DEFAULT NULL,
  `stn` int(11) unsigned NOT NULL,
  `logurl` varchar(256) DEFAULT NULL,
  `logbredd` int(11) DEFAULT NULL,
  `loghojd` int(11) DEFAULT NULL,
  `hemsida` varchar(256) DEFAULT NULL,
  `allminfo` text,
  `aihash` varchar(64) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
  `forecolor` varchar(7) NOT NULL DEFAULT '#000000',
  `backcolor` varchar(7) NOT NULL DEFAULT '#FFFFFF',
  `kontaktpersonid` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,
  `ikonid` int(11) NOT NULL,
  `orgnr` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=12 ;
```

```
-- -----
```

```
--
```

```
-- Tabellstruktur `medstatus`
```

```
--
```

```
DROP TABLE IF EXISTS `medstatus`;
CREATE TABLE IF NOT EXISTS `medstatus` (
  `ID` int(11) NOT NULL,
  `Info` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=4 ;
```

```
-- -----
```

```
--
```

```
-- Tabellstruktur `msg`
```

```
--
```

```
DROP TABLE IF EXISTS `msg`;
CREATE TABLE IF NOT EXISTS `msg` (
  `ID` int(11) NOT NULL,
  `meddelande` text NOT NULL,
  `kontraktid` int(11) DEFAULT NULL,
  `kontaktid` varchar(50) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=4 ;
```

```
-- -----
```

```
--
```

```
-- Tabellstruktur `oppettider`
```

```
--
```

```
DROP TABLE IF EXISTS `oppettider`;
CREATE TABLE IF NOT EXISTS `oppettider` (
  `kontraktid` int(11) NOT NULL,
  `veckodagarid` int(11) NOT NULL,
  `oppet` time DEFAULT NULL,
  `stangt` time DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----

--
-- Tabellstruktur `specialtider`
--

DROP TABLE IF EXISTS `specialtider`;
CREATE TABLE IF NOT EXISTS `specialtider` (
  `ID` int(11) NOT NULL,
  `kontraktid` int(11) NOT NULL,
  `specstart` date NOT NULL,
  `specslut` date NOT NULL,
  `veckodagarid` int(11) NOT NULL,
  `altoppet` time NOT NULL,
  `altstangt` time NOT NULL,
  `stangt` tinyint(1) NOT NULL DEFAULT '1'
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

-----

--
-- Tabellstruktur `veckodagar`
--

DROP TABLE IF EXISTS `veckodagar`;
CREATE TABLE IF NOT EXISTS `veckodagar` (
  `ID` int(11) NOT NULL,
  `akro` varchar(4) NOT NULL,
  `veckonamn` varchar(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=8 ;

--
-- Index för dumpade tabeller
--

--
-- Index för tabell `adress`
--
ALTER TABLE `adress`
  ADD PRIMARY KEY (`ID`);

--
-- Index för tabell `edit_foretag`
--
ALTER TABLE `edit_foretag`
  ADD PRIMARY KEY (`kontraktid`), ADD KEY `kontraktid` (`kontraktid`), ADD KEY `status` (`status`),
  ADD KEY `ikonid` (`ikonid`), ADD KEY `meddelande` (`meddelande`);

--
```

```
-- Index för tabell `edit_kntper`
--
ALTER TABLE `edit_kntper`
  ADD PRIMARY KEY (`kontaktid`), ADD KEY `kontraktid` (`kontaktid`), ADD KEY `status` (`status`), ADD
  KEY `meddelande` (`meddelande`);

--
-- Index för tabell `faktura`
--
ALTER TABLE `faktura`
  ADD PRIMARY KEY (`ID`), ADD KEY `agarid` (`agarid`);

--
-- Index för tabell `felmeddelande`
--
ALTER TABLE `felmeddelande`
  ADD PRIMARY KEY (`ID`), ADD KEY `fronid` (`fronid`), ADD KEY `tillid` (`tillid`), ADD KEY
  `medstatus` (`medstatus`), ADD KEY `feltypid` (`feltypid`);

--
-- Index för tabell `felstatus`
--
ALTER TABLE `felstatus`
  ADD PRIMARY KEY (`ID`);

--
-- Index för tabell `feltyp`
--
ALTER TABLE `feltyp`
  ADD PRIMARY KEY (`ID`);

--
-- Index för tabell `foretag`
--
ALTER TABLE `foretag`
  ADD PRIMARY KEY (`orgnr`);

--
-- Index för tabell `ikontyp`
--
ALTER TABLE `ikontyp`
  ADD PRIMARY KEY (`ID`);

--
-- Index för tabell `kontaktperson`
--
ALTER TABLE `kontaktperson`
  ADD PRIMARY KEY (`anvnamn`), ADD UNIQUE KEY `anvnamn` (`anvnamn`);

--
-- Index för tabell `kontrakt`
--
ALTER TABLE `kontrakt`
  ADD PRIMARY KEY (`ID`), ADD KEY `ikonid` (`ikonid`), ADD KEY `kontaktpersonid` (`kontaktpersonid`),
  ADD KEY `foretagid` (`orgnr`), ADD KEY `foretagid_2` (`orgnr`), ADD KEY `orgnr` (`orgnr`);

--
```



```
-- Index för tabell `medstatus`
--
ALTER TABLE `medstatus`
ADD PRIMARY KEY (`ID`);

--
-- Index för tabell `msg`
--
ALTER TABLE `msg`
ADD PRIMARY KEY (`ID`), ADD KEY `kontraktid` (`kontraktid`), ADD KEY `kontaktid` (`kontaktid`);

--
-- Index för tabell `opettider`
--
ALTER TABLE `opettider`
ADD PRIMARY KEY (`kontraktid`,`veckodagarid`), ADD KEY `veckodagarid` (`veckodagarid`);

--
-- Index för tabell `specialtider`
--
ALTER TABLE `specialtider`
ADD PRIMARY KEY (`ID`), ADD KEY `kontraktid` (`kontraktid`), ADD KEY `kontraktid_2`
(`kontraktid`,`veckodagarid`), ADD KEY `veckodagarid` (`veckodagarid`);

--
-- Index för tabell `veckodagar`
--
ALTER TABLE `veckodagar`
ADD PRIMARY KEY (`ID`);

--
-- AUTO_INCREMENT för dumpade tabeller
--

--
-- AUTO_INCREMENT för tabell `adress`
--
ALTER TABLE `adress`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=12;

--
-- AUTO_INCREMENT för tabell `faktura`
--
ALTER TABLE `faktura`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT för tabell `felmeddelande`
--
ALTER TABLE `felmeddelande`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT för tabell `felstatus`
--
ALTER TABLE `felstatus`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT för tabell `feltyp`
--
```

```
ALTER TABLE `feltyp`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=5;
--
-- AUTO_INCREMENT för tabell `ikontyp`
--
ALTER TABLE `ikontyp`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=3;
--
-- AUTO_INCREMENT för tabell `kontrakt`
--
ALTER TABLE `kontrakt`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=12;
--
-- AUTO_INCREMENT för tabell `medstatus`
--
ALTER TABLE `medstatus`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT för tabell `msg`
--
ALTER TABLE `msg`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT för tabell `specialtider`
--
ALTER TABLE `specialtider`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT för tabell `veckodagar`
--
ALTER TABLE `veckodagar`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=8;
--
-- Restriktioner för dumpade tabeller
--
--
-- Restriktioner för tabell `adress`
--
ALTER TABLE `adress`
ADD CONSTRAINT `kontrakt_ibfk_10` FOREIGN KEY (`ID`) REFERENCES `kontrakt` (`ID`) ON DELETE
CASCADE;
--
-- Restriktioner för tabell `edit_foretag`
--
ALTER TABLE `edit_foretag`
ADD CONSTRAINT `edit_foretag_ibfk_1` FOREIGN KEY (`ikonid`) REFERENCES `ikontyp` (`ID`),
ADD CONSTRAINT `edit_foretag_ibfk_2` FOREIGN KEY (`meddelande`) REFERENCES `msg` (`ID`) ON
DELETE SET NULL,
ADD CONSTRAINT `kontrakt_ibfk_3` FOREIGN KEY (`kontraktid`) REFERENCES `kontrakt` (`ID`) ON
DELETE CASCADE,
ADD CONSTRAINT `status_ibfk_1` FOREIGN KEY (`status`) REFERENCES `medstatus` (`ID`);
--
-- Restriktioner för tabell `edit_kntper`
--
```

```
ALTER TABLE `edit_kntper`
ADD CONSTRAINT `edit_kntper_ibfk_1` FOREIGN KEY (`kontaktid`) REFERENCES `kontaktperson`
(`anvnamn`) ON DELETE CASCADE,
ADD CONSTRAINT `edit_kntper_ibfk_2` FOREIGN KEY (`meddelande`) REFERENCES `msg` (`ID`) ON
DELETE SET NULL,
ADD CONSTRAINT `status_ibfk_2` FOREIGN KEY (`status`) REFERENCES `medstatus` (`ID`);

--
-- Restriktioner för tabell `faktura`
--
ALTER TABLE `faktura`
ADD CONSTRAINT `kontrakt_ibfk_2` FOREIGN KEY (`agarid`) REFERENCES `kontrakt` (`ID`) ON
DELETE CASCADE;

--
-- Restriktioner för tabell `felmeddelande`
--
ALTER TABLE `felmeddelande`
ADD CONSTRAINT `felmeddelande_ibfk_1` FOREIGN KEY (`fronid`) REFERENCES `kontaktperson`
(`anvnamn`) ON DELETE CASCADE,
ADD CONSTRAINT `felmeddelande_ibfk_2` FOREIGN KEY (`tillid`) REFERENCES `kontaktperson`
(`anvnamn`) ON DELETE CASCADE,
ADD CONSTRAINT `felmeddelande_ibfk_3` FOREIGN KEY (`medstatus`) REFERENCES `felstatus` (`ID`),
ADD CONSTRAINT `felmeddelande_ibfk_4` FOREIGN KEY (`feltypid`) REFERENCES `feltyp` (`ID`);

--
-- Restriktioner för tabell `kontrakt`
--
ALTER TABLE `kontrakt`
ADD CONSTRAINT `ikon_ibfk_1` FOREIGN KEY (`ikonid`) REFERENCES `ikontyp` (`ID`),
ADD CONSTRAINT `kontrakt_ibfk_1` FOREIGN KEY (`kontaktpersonid`) REFERENCES `kontaktperson`
(`anvnamn`) ON DELETE CASCADE,
ADD CONSTRAINT `orgnr_ibfk_1` FOREIGN KEY (`orgnr`) REFERENCES `foretag` (`orgnr`) ON DELETE
CASCADE;

--
-- Restriktioner för tabell `msg`
--
ALTER TABLE `msg`
ADD CONSTRAINT `msg_ibfk_1` FOREIGN KEY (`kontraktid`) REFERENCES `kontrakt` (`ID`) ON
DELETE CASCADE ON UPDATE NO ACTION,
ADD CONSTRAINT `msg_ibfk_2` FOREIGN KEY (`kontaktid`) REFERENCES `kontaktperson` (`anvnamn`)
ON DELETE CASCADE ON UPDATE NO ACTION;

--
-- Restriktioner för tabell `oppettider`
--
ALTER TABLE `oppettider`
ADD CONSTRAINT `kontraktid_ibfk_1` FOREIGN KEY (`kontraktid`) REFERENCES `kontrakt` (`ID`) ON
DELETE CASCADE,
ADD CONSTRAINT `veckodagar_ibfk_1` FOREIGN KEY (`veckodagarid`) REFERENCES `veckodagar`
(`ID`);

--
-- Restriktioner för tabell `specialtider`
--
ALTER TABLE `specialtider`
```

```
ADD CONSTRAINT `specialtider_ibfk_1` FOREIGN KEY (`kontraktid`) REFERENCES `feltyp` (`ID`),  
ADD CONSTRAINT `specialtider_ibfk_2` FOREIGN KEY (`veckodagarid`) REFERENCES `kontrakt` (`ID`);
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```