Systemdesign med fokus på säkerhet i ASP.NET

System design in ASP.NET with focus on security

Hassan Dundar
Showan Ramazani

EXAMENSARBETE
Informatik C
Nr: C02/2006

2006

HÖGSKOLAN
Dalarna

EXAMENSARBETE, C-nivå i Informatik

Program	Reg nr	Omfattning
IT-design och systemarkitektur, 120p	C02/2006	10p
Namn		
Showan Ramazani IT3 Hassan Dundar IT3	Juni 2006	
	Handledare: Pär Douhan Examinator: Owen Eriksson	
Företag/Institution	Handledare vid företaget	
I am Dead	Patrik Andersson	
Titel		
Systemdesign med fokus på säkerhet i ASP.NET		
Nyckelord		
ASP.NET, Databaser		

Sammanfattning

Affärsidén bakom detta arbete är baserad på behovet av att lämna dokument och meddelanden i olika format bakom sig, om livet plötsligt rycks bort eller om en person bara vill samla allt efter sig på en säker plats.

Ett stort problem med *I am Deads* befintliga webbplats är att den i nuläget bara är ett skal och inte innehåller några underliggande funktioner. För att kunna hantera eventuella kunder måste webbplatsen kompletteras med nödvändiga tillägg och interaktiva funktioner. Då mycket av informationen som kommer att finnas på webbplatsen är personligt, måste den vara så säker som möjligt för att minimera risken för intrång och förlorad information.

Syftet med detta arbete är förändring av företagets system genom systemutveckling.
För att få bästa tänkbara säkerhet runt webbplatsen kommer vi göra en jämförelse mellan olika tekniker för att få fram den teknik som passar målet bäst.

Metoden som vi använt heter Objekt Orienterad Analys och Design (OOAD), och består av en analys, design och implementations fas.

Vi har även gjort en jämförelse mellan ASP.NET och JAVA och analyserat dessa två, för att komma fram till vilken vi skulle ha till arbetet. Dessutom har vi gjort olika jämförelser mellan olika säkerhetstekniker och analyserat dessa, för att få fram en som skulle passa projektet.

Högskolan Dalarna 2006
Examensarbete Informatik 10p C
Hassan Dundar IT3 Showan Ramazani IT3

DALARNA
University College

DEGREE PROJECT in Information Systems

Course	Reg number	Extent
IT Design and System Architechture, 120p	Nr C02/2006	15 ects
Names Showan Ramazani IT3 Hassan Dundar IT3	Month/Year June 2006	
	Supervisor: Pär Douhan Examiner: Owen Eriksson	
Company/Department I am Dead	Supervisor at the Company/Department Patrik Andersson	
Title System design in ASP.NET with focus on security		
Keywords ASP.NET. Database		

Summary

The business concept for this work/report is based on peoples need to save documents and different kinds of messages, if their lives would suddenly end, or if they just need a safe place to save important documents. The business concept is called "I *am Dead*", and offers different kinds of services to potential clients or users. Clients or users should be able to leave for example orally or written messages and different kinds of documents, which is then saved on the company's server. If something would happen to them, their families and relatives will be able to get these messages or documents.

The main problem with *I am Deads* present website is that it is only a shell and does not contain any real working functions. To be able to deal with clients or users, the website needs to be complete with necessary and interactive functions as well. As a lot of the information provided by the users of the website will be personal, a good security is required to prevent unauthorized people to get hold of it.

The purpose of this report is to change the companies system through system development. We will analyse different security techniques, so that we can use the best alternative for the website. We have also compared ASP.NET and Java and analysed both of them, in order to understand which one we could use for our project. A comparison of different security protocols has also been conducted in order to understand which one would suit the project best.

The method that we have used is called Object Oriented Analyse and Design (OOAD), and consists of the phases, analyse, design and implementation

Högskolan Dalarna 2006
Examensarbete Informatik 10p C
Hassan Dundar IT3 Showan Ramazani IT3

Innehållsförteckning

Bilageförteckning

Högskolan Dalarna 2006
Examensarbete Informatik 10p C
Hassan Dundar IT3 Showan Ramazani IT3

1 Inledning

1.1 Bakgrund

Ett av de känsligaste ämnen som finns runt livets slut är döden. Diskussion runt detta ämne har funnits sen urminnes tider och en fråga som återkommer gång på gång är vad som händer efter livets slut. Olika religioner och samfund har egna förklaringar och svar på denna fråga.

Även om frågan runt vad som händer efter livets slut för den enskilde individen är stor, är många mer bekymrade över de personer de lämnar bakom sig dvs. nära och kära. Det är därför det finns många olika tjänster runt detta, t ex livförsäkringar, testamenten osv. Dessa saker finns för att ens ägodelar tillfaller ens nära och kära när en person dör. Affärsidén bakom detta arbete är baserad på behovet av att lämna dokument och meddelanden i olika format bakom sig, om livet plötsligt rycks bort eller för att ha alla viktiga dokument mm på ett enda ställe.

Designen av webbplatsen för Affärsidén *”I am Dead”* är klar och det siktas på att realisera idén inom kort. Idén går ut på att tillhandahålla olika tjänster till potentiella användare. Ett par exempel på dessa tjänster är möjligheten att göra en lista över personer på hemsidan och kunna spara meddelanden och filer på företagets server, som sen personerna på listan får tillgång till om och när användaren tillåter det. Meddelandena kan vara skriftliga eller muntliga, där muntliga skapas genom att ringa in till företagets telefon och sedan spara meddelandet på sitt personliga utrymme på webbplatsen genom att följa instruktionerna.

Eftersom mycket av den information som kommer att lagra på webbplatsen är väldigt personligt, behövs det en hög säkerhet runt inloggning och dataförvaring. Därför vill *I am Dead* att säkerheten skall genomsyra hela proceduren så att användarna kan känna att deras information och filer är så säkra som möjligt.

1.2 Problemställning

Ett stort problem med *I am Deads* befintliga webbplats är att den i nuläget bara är ett skal och inte innehåller några underliggande funktioner. För att kunna hantera eventuella kunder måste webbplatsen kompletteras med nödvändiga tillägg och interaktiva funktioner. Då mycket av informationen som kommer att finnas på webbplatsen är personligt måste den vara så säker som möjligt för att minimera risken för intrång och förlorad information.

1.3 Syfte

Syftet med detta arbete är förändring av företagets system genom systemutveckling.
För att få bästa tänkbara säkerhet runt webbplatsen kommer vi göra en jämförelse mellan olika tekniker för att få fram den teknik som passar målet bäst.

1.4 Mål

Målet med detta arbete är att skapa en normaliserad databas, samt funktioner för att hantera registrering och inloggning till webbplatsen. Det skall även finnas funktioner för att ladda upp filer, översikt över användarens uppgifter, översikt över de dokument som lämnats och möjligheten att ändra uppgifter. En GUI för alla funktioner skall också skapas. En lösning till problemet med att ta emot, identifiera och spara ett meddelande från en telefonväxel till företagets server skall också diskuteras fram. Dessa funktioner skall komplettera *I am deads* nuvarande webbplats för att möjliggöra hantering av riktiga kunder.

Målet med jämförelsen mellan de olika teknikerna är att ta reda på vilken teknik som passar uppgifterna bäst.

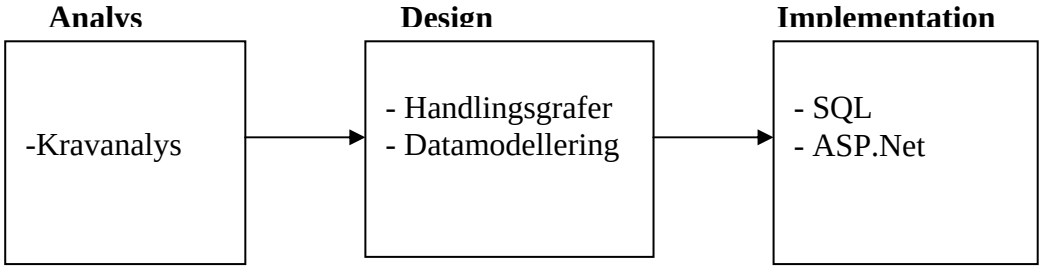
1.5 Avgränsning

Eftersom *I am Dead* redan har en uppbyggt webbplats, kommer vi inte att göra några större förändringar på den. Vi behöver inte heller installera någon form av telefonväxel, utan bara komma med olika förslag till detta. Jämförelsen kommer bara att ske på de två största teknikerna för webbapplikationer, JAVA och ASP.NET. Vi kommer inte att förvalta databasen eller systemen efter arbetets slut.

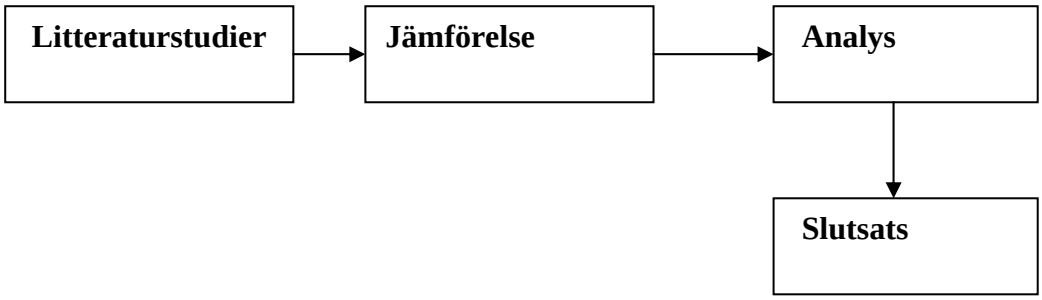
1.6 Metodöversikt

Vi kommer att använda oss av litteraturstudier för att samla in information som är relevant för studien. Det mesta av litteratursökningen kommer att genomföras på Internet med söktmotorer och övrig litteratur hittas genom databasen på högskolan Dalarnas Bibliotek i Borlänge. Förutom självaste informationsinsamlingen kommer vi att ha regelbunden kontakt genom möten och telefonsamtal med uppdragsgivaren för att diskutera och bestämma vart studien skall leda.

För att bygga upp de nödvändiga applikationerna och funktioner kommer vi använda oss av OOAD – Objekt Orienterad Analys och Design. Nedan är ett flödesschema på de faser av arbetet och de metoder och tekniker som kommer att användas i respektive fas.



Nedan är ett flödesschema på den teoretiska delen av arbetet:



2. Metod

I denna sektion beskrivs de olika faserna i metoden och vad som kommer att användas i respektive fas.

2.1 Analys

Den främsta uppgiften i denna fas är att förstå och beskriva det bakomliggande problemet. Det gäller att utforska problemområdet och användarnas förståelse av det. Under denna process behöver man inte bry sig om existerande system eller tänka på tekniska problem när det gäller lagring och överföring av data från andra system[1]

2.1.1 Kravanalys

Här fokuseras det på hur målsystemet kommer att användas. Syftet är att definiera kraven på systemets funktioner och gränssnitt. I denna del fokuserar vi alltså på användarens arbete och specificerar detaljerade krav som sedan används för att modellera klasser, händelser och strukturer i modellen av problemområdet.

Dessa delar kommer att utföras genom att intervjua personen bakom idén och göra en lista på de krav som ställs på de nya funktionerna.(Se bilaga 3)

2.2 Design

I designfasen utgår man från de erfarenheter som har införskaffats från analysfasen och bygger vidare på dem. Nästa steg är att fundera på hur ett program skall implementeras på en dator och hur det skall byggas up i stora drag. Ett par exempel på vad som bestäms i denna fas är vilken hårdvara som behövs för att köra programmet, hur kommunikationen med andra datorer skall ske, hur data skall lagras i databaser, vilka datastrukturer och algoritmer som programmet skall använda osv.

2.2.1 Handlingsgrafer

Vi kommer att använda oss av handlingsgrafer för att få en översikt över olika processer runt funktionerna. (Se bilaga 4)

2.2.2 Datamodellering

Genom datamodellering skall en översikt över databasen och de olika tabellerna skapas för att sedan användas i implementationsfasen. (Se bilaga 5)

[1] Objektorienterad analys och design, sid 65

2.3 Implementation

I denna fas utförs programmeringen. Om analysen och designen har gjorts på rätt sätt så har man förstått problemet och vet i grova drag hur detta ska lösas. Under implementationen brukar brister i de föregående faserna bli påtagliga. Detta kan leda till att man behöver gå tillbaka och göra om vissa delar av analys och design faserna.

2.3.1 SQL

Uppbyggnaden av databasen kommer att ske i My-SQL miljö då vi har erfarenhet i att arbete i denna miljö.

2.3.2 ASP.NET

Utvecklingen av de olika funktionerna kommer huvudsakligen att ske i ASP.NET eftersom vi har arbetat med det förut och tycker det passar syftet bra. (Se bilaga 6)

3. Teoretiskt ramverk

3.1 Säkerhet

Ordet kryptering härstammar från det grekiska språket och betyder ”hemlig skrift”. Behovet av att hålla information hemligt har funnits länge och användes redan under romarnas tid av kejsaren Julius Cesar. På den tiden var metoderna enkla, men uppfyllde sitt syfte. Under 1900- talet utvecklades krypteringen väldigt mycket på grund av första och andra världskriget som uppstod. Ofta när ordet kryptering nämns i datorsammanhang vill man uppnå konfidentalitet, dvs. hålla information hemlig. Det finns även flera andra områden t ex när digitala signaturer ska skapas. Digitala signaturer är digitala underskrifter för dataintegritet. Principen bakom kryptering är att med hjälp av en krypteringsalgoritm förvränga hemlig information. En normal text sägs vara i klartext, medan en förvrängd text kallas kryptotext.

World Wide Web eller WWW som det kallas är själva Internet där det går att surfa omkring på olika sidor runt om i världen. Internet är ett öppet datanät, som består av olika sammankopplade nätverk, servrar, routrar och annan extern datakommunikationsutrustning. I början av Internets historia prioriterades betydelsen av ett öppet nätverk, och vid planeringen kunde dagens dataskyddsproblem inte förutses. Detta kan lösas genom att helt enkelt skydda den information som ska vara skyddad från obehöriga, och kryptering är svaret på problemet.[2]

Kryptering är en teknologi som utvecklades för att dölja information för obehöriga personer, och har varit särskilt användbar i olika krigssituationer, där man velat skicka meddelanden utan att fienden ska kunna förstå de. Detta spred sig sakta men säkert till den civila världen och Internet. Eftersom Internet är ett öppet och stort nät så finns det alltid personer som vill komma åt information som inte tillhör dem. Krypteringen kan förhindra detta genom att försvåra för obehöriga att ta del av hemlig information. Krypteringsteknologi kan också användas för identifikation i nätverkssammanhang - att avgöra vem som kommunicerar med vem, och dessutom går det att skapa en digital signatur, en sorts namnteckning eller stämpel på ett dokument. Med dagens teknik går det även att se om ett meddelande t ex ändrats på vägen till en mottagare. Konfidentialitet - att dölja information, identifikation och integritet är grundstenar för nätverkssäkerhet. Utan denna grundläggande säkerhet blir det svårt för företag och personer att använda Internet för t ex handel. [3]

Utifrån detta utvecklades olika krypteringsprotokoll, där de två mest kända är **SSL** och **S-HTTP**. SSL används av de flesta företag och anses vara standard kryptering, men även S- HTTP används i samhället. Båda protokollen har blivit godkända av IETF (Internet Engineering Task Force).[4]

3.1.1 Krypterings protokoll

[2] http://www.if.fi/web/fi/private.nsf/0/E9BEB53EB8C51B3BC1256FAA003C5BA0
[3] http://www.intranetica.com/intranetica/kds/krypto.shtml
[4] http://www.webopedia.com/TERM/S/SSL.html

SSL

SSL står för Secure Sockets Layer och gör det möjligt att kryptera förbindelsen mellan användarens webbläsarprogram och webbservern. Krypteringen skyddar datatrafiken på så sätt att utomstående inte kan komma åt viktiga uppgifter genom att följa upp förbindelsen. SSL-förbindelsen börjar med en s.k. handskakning, varigenom det bildas en skyddad förbindelse mellan kundens webbläsarprogram och webbservern. I samband med handskakningen avtalar kundprogramvaran och servern om förbindelsespecifika engångskrypteringsnycklar. De används under sessionen för att kryptera och tolka informationen. Utöver krypteringen av förbindelsen möjliggör SSL-protokollet också verifiering av serviceleverantören med hjälp av certifikat. Verifieringen sker på så sätt att servern har ett eget servercertifikat, på basis av vilket användaren kan säkerställa att han/hon verkligen kommunicerar med rätt webbserver.

SSL använder ett krypteringssystem som använder två nycklar för att kryptera data, en öppen som alla känner till och en hemlig, privat nyckel som är bara känd för mottagaren av meddelandet. Både Netscape Navigator och Internet Explorer stödjer SSL, och det går att se på webbläsarens URL (Uniform Resource Locator) om sidan är SSL krypterad eftersom det då börjar med https: (Se bild 3.1.1.2) istället för http**,** som då visar att det inte är krypterad (Se bild 3.1.1.1).[5] Dessutom syns ett hänglås i webbläsarens nedre högra hörn som visar SSL skyddet (Se bild 3.1.1.3). Klickar man på hänglåset kommer det upp ett certifikat som anger till vem certifikatet är utfärdat på. Stämmer adressen där med adressen i webbläsarens verktygsfält betyder det att det är ett äkta intyg (Se bild 3.1.1.4).

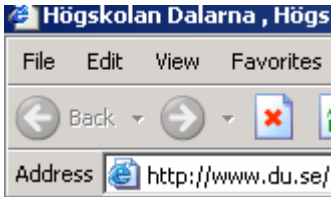


Bild 3.1.1.1 Ej SSL skyddad webbplats

Bild 3.1.1.2 SSL skyddat webbplats

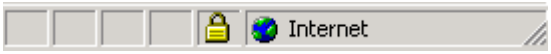


Bild 3.1.1.3 Hänglåset är ännu ett synlig tecken på SSL skyddet.

[5] http://www.webopedia.com/TERM/S/SSL.html

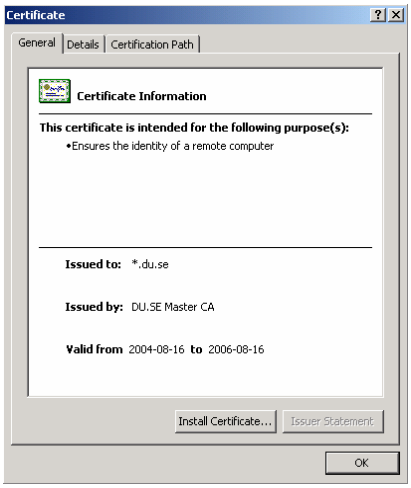


Bild 3.1.1.4 Exempel på SSL-certificat som blir synligt när man klickar på hänglåset i bild3

Vad betyder 128-bits SSL-kryptering?

Det finns olika versioner av SSL-systemet. Den siffra som anges i krypteringssystemets namn anger krypteringsnyckelns längd vid överföring av den krypterade informationen. Ju längre nyckel, desto svårare är det att knäcka krypteringen, vilket innebär att 128-bits kryptering är många gånger säkrare än 40-bits kryptering. En oskyddad förbindelse kan jämföras med att sända ett postkort per post. Alla som hanterar kortet kan också se vad som står på det. 40- och 56-bits kryptering motsvarar ungefär användningen av kuvert, medan 128-bits kryptering är som att låsa in innehållet i ett kassaskåp, som levereras till mottagaren under bevakning. I praktiken stöder de nyaste versionerna av alla de vanligaste webbläsarna 128-bits kryptering. Webbläsarprogrammet borde ställas in på högsta möjliga skyddsnivå.

Varför är inte alla förbindelser SSL-krypterade?

Mottagningen av SSL-krypterade meddelanden och krypteringen av den information som överförs kräver mycket kapacitet av serverutrustningen. Den tid som går åt för dessa åtgärder syns också i svarstiden för tjänsten/sidan.[6]

SSL och HTTP

SSL är implementerat mellan TCP/IP och applikationsprotokollet. SSL kan exempelvis utnyttjas av andra protokoll såsom HTTP, FTP, TELNET med flera. Den mest kända användningen av SSL är med HTTP och kallas HTTPS (SSL + HTTP). Användandet av HTTPS fungerar på samma sätt som HTTP, fast via en annan port, SSL-porten (port 443), istället för den vanliga HTTP-porten (port 80). Detta innebär att en webbserver som stöder SSL, måste lyssna på både port 80 och 443. I en bläddrare, som Netscape Communicator, visas prefixet https:// istället för http://, då HTTPS används.

HTTPS ska dock inte förväxlas med S-HTTP, som är en annan standard utvecklad av E. Rescorla och A. Schiffman, 1994. HTTPS är ett protokoll som lagts ovanpå HTTP och kan

[6] http://www.if.fi/web/fi/private.nsf/0/E9BEB53EB8C51B3BC1256FAA003C5BA0

därför bara användas med HTTP. Om SSL syftar till att säkra all överföring mellan två parter, syftar S-HTTP till att säkra enstaka meddelanden. S-HTTP är mycket flexibelt och stöder många olika krypteringsalgoritmer. S-HTTP har idag dock inte samma stöd som SSL/HTTPS. En session som utnyttjar SSL, börjar med en s.k. handskakningsprocedur. Denna innebär att servern och klienten kommer överens om vilken krypteringsalgoritm som ska användas och nycklar utbyts samt en del annan information såsom eventuella digitala certifikat. Efter handskakningen kan överföring ske på ett säkert sätt.
Det har dock diskuterats i termer av att denna kryptering inte innebär någon större försvåring för "professionella" bedragare, utan snarare förenklar deras arbete. Dekrypteringen av paketen har i och med den stora kapacitetsökningen hos persondatorerna blivit relativt enkel.[7]
S-HTTP
Protokollet som WWW (World Wide Web), klienter och servrar använder för att kommunicera med är http (Hypertext Transer Protokoll). HTTP är det ideala för öppen kommunikation, men det erbjuder inte autentisering och krypterings möjligheter. S-HTTP utvecklades för att kunna användas tillsammans med HTTP för att göra det möjligt för klienter och servrar att möjliggöra privata och säkra transaktioner. S-HTTP blev aldrig riktigt accepterat av utvecklare som Microsoft och Netscape. Istället blev en annan krypterings protokoll populär vid namn SSL (Secure Sockets Layer). SSL erbjuder samma autentisering och krypterings funktioner som https, men SSL har utökade funktioner som att kryptera all data som passerar mellan klient och server, inklusive data på IP nivån. S-HTTP krypterar bara http nivå meddelande. S-http används fortfarande eftersom det erbjuder olika krypterings algoritmer. Meddelanden kan skyddas av olika digitala signaturer, autentisering och som sagt, kryptering. Vid första kontakt, så får både sändaren och mottagaren kontakt och sätter upp en säker linje för att kryptera och hantera säkra meddelanden.[8]

TLS
SSL tillverkades av Netscape 1994 som ett säkerhetsprotokoll. 1997 blev en ” Open source ” version av Netscapes patentskyddade version tillverkad som nu heter ”OpenSSL”.[9] OpenSSL projektet är en gemensam ansträngning för att utveckla en robust/stadig, kommersiell version med alla funktioner och ett Open Source verktyg för att kunna implementera SSL v2 och v3 och TLS protokollet v1 med en fulländad krypterings bibliotek. Projektet sköts av frivilliga runtom i världen som använder Internet för att kommunicera, planera och utveckla OpenSSL projektet, dess verktyg och all dokumentation. Det är gratis att ladda ner.[10] Netscape släppte utvecklingen fritt ett par år efter att de tillverkade SSL, och IETF tog över utvecklingen. TLS som står för Transport Layer Security ska ersätta SSL. TLS består av två protokoll, en handskakning och ett inspelningsprotokoll. Handsskakningsprotokollet gör en hemlig nyckel som används av protokollet för att dekryptera ett meddelande. TLS är också tillverkat så att det ska vara applikationsoberoende.[11]

Fortfarande används namnet SSL på det nya protokollet TLS. TLS krypterar alla meddelanden som skickas mellan klient browsern och servern. Det som händer när en förbindelse upprättas mellan klient och server är att handskakningsprotokollet upprättar den här förbindelsen. Eftersom detta tar tid för att vara säker på att allting går rätt till så förklarar

[7] http://www.cs.umu.se/kurser/TDBD07/VT00/rapp/rapp1.htm
[8] http://www.linktionary.com/s/shttp.html
[9] http://www.nyphp.org/content/presentations/SSL/tls.html
[10] http://www.openssl.org/
[11] http://www.networkworld.com/details/789.html

det här varför skyddade sidor på Internet med SSL/TLS protokoll är lite långsammare än oskyddade. [12]

Som sagt så ska TLS ersätta SSL, men trots det kommer SSL finnas kvar, däremot har det nya protokollet lite fler funktioner än det gamla. Förbättrade funktioner över SSL är t ex att det nu finns nyckelvärdepar för att autentisera ett meddelande (key-Hashing for message Authentification HMAC, jämfört med Message Authentification Code MAC för SSL). På detta sätt går det att försäkra sig att ett meddelande inte förändras. Förbättrad pseudorandom funktion (PRF) som finns definierad i HMAC, som går ut på att de två värdparen som finns ska skydda meddelandets innehåll. Även om den ena av värdparen skulle vara synlig så håller sig information osynlig tills det att det andra värdparet blir synligt. Protokollet har också en funktion som gör att den kan se om det mottagna meddelandet har förändrats på vägen. SSL har också detta, men det som gör det säkrare med TLS är att den baserar detta skydd på HMAC och PRF funktionerna. Dessutom finns en funktion som gör att den måste ange vilket certifikat som ska användas mellan klient och server för att säkra en förbindelse. Den sista funktionen gör att den upptäcker problem som kan upplevas från båda delarna i en session och säger även till när vissa varningar borde skickas ut. [13]

3.1.2 För och nackdelar med SSL och S-HTTP

SSL
En av nackdelarna med SSL är att den är konstruerad så att protokollet helt förlitar sig på förbindelseorienterade protokoll. Därför går det inte att köra SSL över förbindelselösa protokoll som UDP. SSL är också enbart ett kommunikationsprotokoll vilket betyder att protokollet måste förlita sig på andra program för vissa funktioner. Ett exempel är att SSL inte använder egna krypteringsmetoder utan måste förlita sig på redan existerande algoritmer, som t ex RSA (namngett efter dess skapares efternamn, Ron Rivest, Adi Shamar, Len Adleman). Fördelarna är att SSL stöds av nästan alla browsers, som t ex Netscape och Internet Explorer. SSL protokollet förser också browsern med säkerhet, förtroende, autentisering och meddelandeintegritet.

S-HTTP
Säkerhetstjänster i S-HTTP är inte möjliga att använda i FTP, och används endast i applikationsprotokollet dvs. HTTP nivån och är bara vändbara i HTTP, alltså inte möjligt att använda det i t ex FTP. Det binder också säkerheten mer och mer till applikationsnivån. Dessutom måste säkerheten modifieras varenda gång som HTTP ändras. Eftersom S-HTTP bygger på krypteringsprotokoll som finns, så kan den som knäcker krypteringen även komma åt S-HTTP protokollet. Den största nackdelen är att den inte är lika användbar som SSL, utan S-HTTP kan endast användas för att kryptera vanlig http trafik.

Fördelarna med S-HTTP är den stödjer en mängd olika krypteringsprotokoll och meddelanden som skickas kan skyddas med hjälp av digitala signaturer, autenticifikation och kryptering. Precis som med SSL och TLS har även S-HTTP en funktion där handskakning sker mellan klient och server för att säkerhetsställa bådas identitet. [14]

[12] http://www.nyphp.org/content/presentations/SSL/tls.html
[13] http://publib.boulder.ibm.com/infocenter/cicsts/v3r1/index.jsp?topic=/com.ibm.cics.ts.doc/dfhe4/SSL/transportlayersecurity1.0protocol.htm
[14] http://www.linktionary.com/s/shttp.html

3.1.3 Kryptoalgoritmer[15]

Grunden i kryptografi är det kodsystem som används. Det finns flera olika metoder för kryptografi. För varje metod finns det ett antal olika sätt att räkna, det som kallas algoritmer. En sådan formel är ofta skyddad av patent och kräver licens för användning. Metoderna skiljer sig också åt t ex när det gäller hastighet. Ju mer komplicerad och svårgenomtränglig ett sätt att kryptera information är, ju mer tid går det åt för datorn som ska kryptera. Ska ett meddelande i en e-post krypteras är det gott om tid, men när det gäller att kryptera nätverkstrafik i hög hastighet mellan två datorer är tidsåtgången ofta kritisk. Det finns två krypterings sätt, den ena kallas rak kryptering och den andra kallas asymmetrisk kryptering.

Rak kryptering
För rak kryptering används samma nyckel för kryptering och dekryptering. Det gör att både sändare och mottagare har samma nyckel. Rak kryptering används ofta vid kryptering av en session där hastigheten är viktig. Ett problem med nyckeln är att den inte kan överföras mellan avsändare och mottagare okrypterad över ett nätverk. Ett problem är överföring av nyckeln, som är gemensam. Den kan inte överföras mellan avsändare och mottagare okrypterad över ett nätverk. Den måste överföras på ett säkert sätt.

Asymmetrisk kryptering
Asymmetrisk kryptering baseras på två olika nycklar. Den ena kallas hemlig (ofta också "privat" från engelska "private key") och den andra öppen (ofta också "publik" från engelska "public key"). Kryptering med den ena nyckeln kan endast dekrypteras med den andra. Nycklarna hänger alltså mycket intimt ihop. Det gör att den ena nyckeln, som kallas den öppna nyckeln, kan göras fritt tillgänglig. Eftersom nycklarna är beroende av varandra går det att vara säker på att information som kan dekrypteras med den ena nyckeln är krypterad med den andra. Omvänt kan man också vara säker på att om information krypteras med en nyckel, kan den bara dekrypteras med den andra. Symmetrisk kryptering används ofta vid identifiering, till exempel digitala signaturer. Det används också för säkra meddelanden. Nyckelns längd är avgörande

När krypteringen utförs av datorer kan den också knäckas med datorer. Därför är nyckelns längd avgörande. Ju längre nyckellängd, desto längre tid tar det att knäcka. Det handlar givetvis också om algoritmen, men ofta används en känd algoritm och då är nyckelns kvalitet och längd avgörande faktorer. Idag används 40-bitars symmetrisk kryptering för SSL-sessioner i webben, krypterade sessioner som används till exempel för Internetbanker. Inom USA används 128 bitars kryptering, men exportreglerna begränsar nyckelns längd vid export av amerikanska programvaror. 40 bitars kryptering tar inte lång tid att knäcka och är inte ett bra skydd som ger konfidentialitet för din dataöverföring. För identifiering används asymmetrisk kryptering, med ett nyckelpar som hör ihop. För sådana finns inte samma exporthinder, vilket gör att det blir möjligt att använda nycklar med 512 eller 1024 bitar. Vilken nivå av kryptering behöver jag till min webbplats?

Den bästa säkerhetspraxis är att installera ett unikt certifikat på varje server och välja äkta 128 bitars kryptering eller bättre genom att köpa ett SSL-certifikat med SGC-funktion. Ett unikt certifikat upprätthåller skyddet för dina privata nycklar och ett certifikat med SGC-funktion

[15] http://www.intranetica.com/intranetica/kds/krypto.shtml

säkerställer att varje webbplatsbesökare, oavsett vilken webbläsare eller vilket operativsystem de använder, kopplar upp sig med den högsta krypteringsnivån deras system klarar. Nivån av skydd som krävs bör baseras på din informations värde och dina kunders uppfattning. Du behöver 128 bitars kryptering eller bättre om du behandlar betalningar, delar konfidentiella uppgifter eller samlar in personligen identifierbara uppgifter såsom personnummer eller skatteuppgifter, postadress eller födelsedatum. Du behöver 128 bitars kryptering eller bättre om dina kunder känner oro kring sekretessen för de uppgifter de sänder till dig. [16]

3.1.4 SSL certifikat

För att kunna använda ett SSL (Secure Socket Layer) protokoll behövs det ett certifikat som anger att den sida som används är skyddad från obehöriga om viktiga uppgifter är involverade. Du behöver ett SSL certifikat om:

- Du har en online-butik eller tar emot beställningar och accepterar kreditkort online
- Dina affärspartners loggar in på konfidentiell information på ett extranät.
- Du har kontor som delar konfidentiell information i ett intranät
- du behandlar känsliga data såsom adress, födelsedatum, licens eller ID-nummer
- Du behöver efterleva krav på sekretess och säkerhet
- Du värdesätter privatlivet och förväntar dig att andra skall lita på dig

Ett SSL-certifikat består av en publik nyckel och en privat nyckel. Den publika nyckeln används för att kryptera information och den privata nyckeln används för att avkryptera den. När en Webbläsare pekar mot en säkrad domän eller adress som t ex www.aftonbladet.se, ser SSL till att visa att denna sida är säkrad och innehållet korrekt genom en handskakning med servern och klienten och ser till att börja kryptera informationen mellan de två och en unik sessionsnyckel. Om en webbsida är SSL säkrat så visas ett hänglås i Internet Explorers nedre högra hörn. För att vara helt säkert på om certifikatet är äkta kan en användare klicka på hänglåset för att se vem det är utfärdat till. Vid ett äkta certifikat stämmer adressen i certifikatet med adressen högst upp i adressfältet. De kan påbörja en säker session som garanterar meddelandesekretess och meddelandeintegritet.[17]

När behöver du använda ett SSL certifikat?
Att säkra överföringen av finansiell information inom e-handel är för tillfället den största användningen av SSL certifikat. Med ökningen av identitetsstölder, har skydd av personlig identifierbar information blivit allt viktigare. Denna kategori inkluderar identitet, socialskyddssignum och e-post adresser.
Så om du behandlar finansiell information på din webbplats, är det helt klart att ett SSL certifikat är ett krav. Om du behandlar känslig kunddata, är det värt att seriöst överväga användning av ett SSL certifikat, speciellt om kundsekretess står högt på din prioritetslista.

Varför använda ett SSL certifikat?
Det finns två större orsaker till varför du borde använda ett SSL certifikat:

[16] http://www.verisign.se/products-services/security-services/ssl/ssl-information-center/faq/ssl-encryption.html#01000003
[17] http://www.verisign.se/products-services/security-services/ssl/ssl-information-center/how-ssl-security-works/index.html

- För att bevisa att ditt företags (eller din servers) identitet online och genom att göra det skapar du samtidigt en trovärdighet och säkerhet för dina kunder att använda din webbplats
- Säkerhet av data inlämnad till din webbplats (eller mellan servers) genom användning av krypteringen Skulle någon information hamna fel, är den oanvändbar utan den unika nyckeln som används för avkryptering

3.1.5 Lösenord

Olika företag har olika sorters säkerhetssystem och olika hemsidor på Internet har även de någon sorts säkerhet i form av lösenord. Detta är ett första led av säkerhetssystem som företag kan använda sig av. En hemsida kan använda lösenord för att förhindra att obehöriga får tillgång till information som de inte skall ha tillgång till. För att fel personer inte ska få reda på ett lösenord så skall det bytas med jämna mellanrum och att inte användas någonting lättgissat. Ett bra lösenord ska ha en blandning av tecken och lösenord (se tabell 3.1.4.1), följande ingredienser är bra att ha:

- Vara minst 7 tecken lång
- Bestå av tecken ur följande klasser, Bokstäver (ABCabc), Nummer (12345), Andra symboler (#&%)
- Innehålla minst en symbol i mitten av lösenordet
- Skilja sig rejält från t ex tidigare lösenord
- Inte bestå av personnamn
- Inte bestå av vardagliga fraser

Exempel på ett bra lösenord: T22#BoL4u[18]

	Lösenord	Möjliga kombinationer
Bokstäver (2)	Ab	676
Bokstäver (4)	Abcd	456 976
Bokstäver (7)	Abcdefg	8 miljarder
Bokstäver och nummer (7)	A1b2c3d	78 miljarder
Bokstäver, nummer och Symboler (6)	A1@b2%	98 miljarder
Bokstäver, nummer och Symboler (7)	A1@b2%c	6 700 miljarder

Tabell 3.1.4.1. Kombinationen av tecken och längd i lösenord kan ge många kombinationer.

3.1.6 Dataintrång[19]

För nästan 25 år sen började lagar att dyka upp som handlade om IT säkerhet. Detta berodde på att vissa beteenden började dyka upp och det blev tvunget att straffa dessa beteenden för att i grund och botten skydda individen. Sådana beteenden som en person kan få straff för är

[18] Mitrovic, P.2002, s 102-103
[19] Maiwald, E, Sieglein W.2002, s 24-30

brott som begås med hjälp av datorer och t ex användas för förtal eller hets mot folkgrupp. Dataintrången började bli fler och fler och för att stoppa detta var blev det helt enkelt tvunget att straffa dessa beteenden. Dataintrång kan inte bara ske genom att t ex en hacker tar sig in i en annan persons dator eller i ett företag, utan detta kan även ske fysiskt genom att själva personen bryter sig in i ett serverrum där det kan tänkas att känslig data förvaras.

När det pratas om dataintrång så kommer man också in på personuppgiftslagen som är till för att skydda individen mot obehöriga. Det är datainspektionen som ser till att övervaka att lagen efterföljs. Den 24 oktober 1998 trädde **personuppgiftslagen** (SFS 1998:204) i kraft. Lagen bygger på ett EU-direktiv och ska hindra att den personliga integriteten kränks genom behandling av personuppgifter. [20] Sveriges lag inom detta område är baserat på Europaparlamentets direktiv om skydd för enskilda personer med avseende på behandling av personuppgifter. EU: s medlemsstater ska enligt direktivet skydda levande fysiska personers grundläggande fri och rättigheter i samband med behandling av personuppgifter. Lagen är till för att skydda personer mot att deras personliga integritet kränks när personuppgifter behandlas. Lagen gäller all form av behandling, som t ex insamling, bearbetning, lagring och spridning. Men för att lagen ska gälla måste det vara möjligt att kunna knyta uppgifterna till en viss person. Om ändå uppgifterna ska granskas och användas på något sätt måste personen ge sitt samtycke till detta, och vid väldigt känsliga uppgifter, i t.ex. samband med något brott, gäller väldigt stränga regler. Anledningen till detta är att skydda individens uppgifter för obehöriga personer som vill på ett eller annat sätt sprida information som de inte har rätt till.

Lagen gäller inte bara uppgifter, men också ljudinspelningar och bilder som kan knytas till en viss person. I ett företag är det den person som är ansvarig för personalens uppgifter och t ex kundernas uppgifter som lagrats, som kan bli straffad i domstol om uppgifterna används på ett felaktigt sätt. Normalt är det ett företag i sin helhet som ansvarar och kan straffas, alltså företaget själv och inte någon specifik person. Företaget som har en affärsidé där kunderna behöver lämna personlig information, måste ange på något sätt att deras uppgifter kommer att behandlas korrekt och att kunderna lämnar sitt samtycke till att deras uppgifter lagras i företaget servrar under en tid eller till dess att kunden inte längre vill ha sina uppgifter hos företaget. Företaget eller personuppgiftsansvarig ska beroende på känsligheten på uppgifterna om deras kunder, skydda informationen i servrarna med hjälp av t.ex. kryptering. Det är också viktigt att förstå att ett företags utrustning, som t ex servrarna, behöver ibland någon form av reparation eller underhåll, och att det då bör göras på sådant sätt att obehöriga inte kan komma åt informationen. Reparation eller service av utrustning bör göras på sådant sätt att uppgifterna inte blir tillgängliga för obehöriga.

[20] http://www.datainspektionen.se/lagar/pul.shtml

3.2 ASP.NET och Java teknologierna

3.2.1 ASP.NET

ASP.NET (Active Server Pages .NET) är en utvecklingsteknologi för webben från Microsoft och är en del av *.Net ramverket*. ASP.Net tillåter användare att bygga dynamiska applikationer och webbtjänster genom att använda programmeringsspråk såsom VB.NET och C#. Den nyaste versionen är i nuläget ASP.NET 2.0.

Bakgrund[21]
När webben var ung så var det flesta av innehållet statiskt. Webbutvecklare var tvungna att ändra sina sidor ofta och manuellt. För att kunna skapa webbsidor som var dynamiska och som kunde uppdateras automatiskt, skapades det en mängd teknologier inklusive Active Server Pages (ASP) från Microsoft. ASP kördes på server sidan och dess resultat skickade till webbläsaren, vilket ledde till att servern kunde generera dynamiska webbsidor beroende på användarens val.

ASP kom till oktober 1996 som en uppgradering till Internet Information Server (IIS) 2.0. I de första tre versionerna av ASP användes scriptspråket VBScript som standardspråk Att använda ett scriptspråk medför vissa nackdelar som t ex att koden *tolkas* istället för att kompileras, och att använda VBScript som standardspråk inte uppskattades av alla. Tolkningsmodellen för koden minskade dessutom prestandan.

År 2000 introducerade Microsoft den nya .Net ramverket och tillsammans med det introducerades den nya versionen av ASP: ASP.NET 1.0 (som tidigare hette ASP+). De senaste åren har ASP.NET uppdaterats till 1.1 och nyligen till 2.0. I ASP.NET är användningen inte begränsad till scriptspråk, utan nu gick det att använda följande språk:

- C# - C-sharp
- J# - Java sharp?
- VB.NET – Visual Basic.NET

Hur ASP.NET fungerar
När en webbläsare gör en förfrågan om en sida till en webbserver, så kontrollerar webbservern (IIS) först om förfrågan är för en HTML-sida. Om det är det, så blir förfrågan uppfylld genom att hämta filerna från servern och skicka det till klienten (webbläsaren). Om klientens förfrågan rör en ASP.NET sida, skickar IIS den till ASP.NET Runtime som processar applikationen och returnerar resultatet till klienten (Se bild 3.2.1).

[21] http://www.ondotnet.com/pub/a/dotnet/2005/09/19/what-is-asp-net.html?page=4

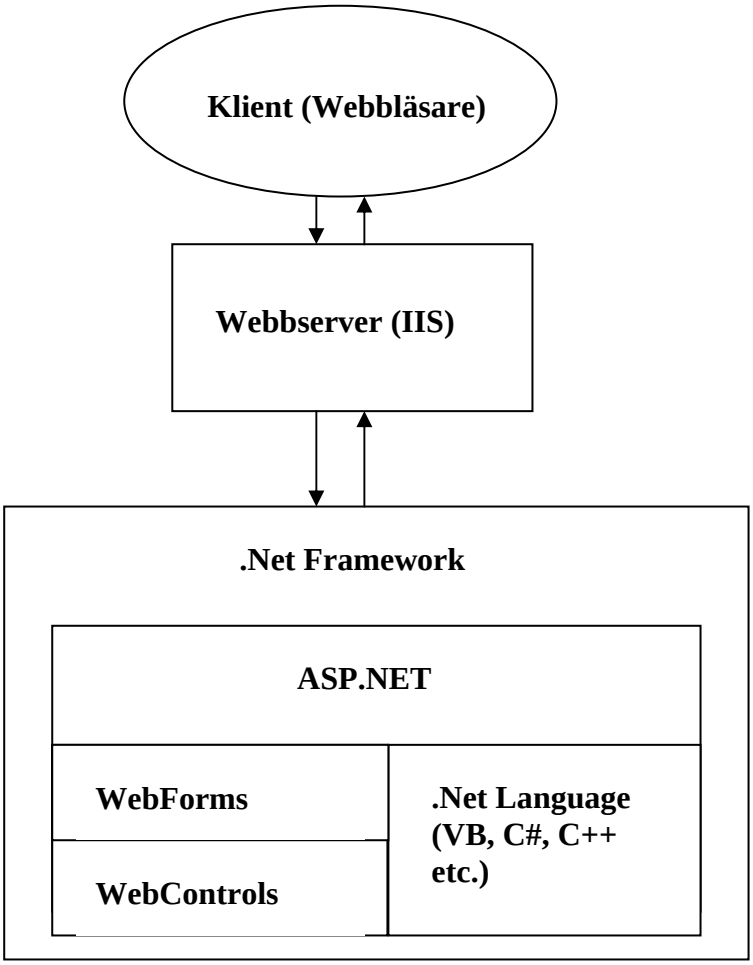


Bild 3.2.1. Beskrivningsmodell av ASP.NET

ASP.NET använder *.aspx* som filändelse. Detta så att ASP.NET kan köras bredvid den klassiska ASP, som använder filändelsen *.asp,* på samma server.

Ett medfött problem med http protokollet är att den är tillståndslös, dvs. en förfrågan från en användare sparas inte i minnet när den väl är utfört. En ny förfrågan processas helt enkelt som en helt ny förfrågan utan hänsyn till vilken användare det är och vad som har begärts före den. Detta gör det svårt att utveckla webbapplikationer då utvecklare gärna vill att servern skall veta det gamla tillståndet på applikationen. Under åren har flera olika verktyg för detta ändamål uppfunnits, som t ex cookies och användning av query strings för överföring av information mellan klient och server.

I gamla ASP behövdes det sidvis med kod för att kunna bevara tillståndet av sidan efter det att användaren har skickat tillbaks ett värde till servern. I ASP.NET så sköts alla dessa uppgifter av dess ***runtime*.**

Vad som behövs för att köra ASP.NET
ASP.NET stöder följande operativsystem:

- Microsoft Windows 2000 Professional and Server
- Microsoft Windows XP Professional
- Microsoft Windows Server 2003

För att kunna köra ASP.NET på en dator, behövs det installeras IIS på den. För att få tillgång till ASP.NET runtime måste .NET ramverket installeras på datorn.

Säkerhet
.NET ramverket är underlaget för alla .NET utvecklings teknologier (exempelvis ASP.NET). De två största komponenterna i .NET ramverket inkluderar följande:

- **The common language runtime**
  Detta är motorn som exekverar koden för alla applikationer som baseras på .NET ramverket. Dess funktioner inkluderar hantering av kod (ladda och exekvera kod), hantering av minne för applikationer och objekt och upprätthålla säkerhetsregler gällande vilka resurser som kan kommas åt av koden.

- **Klassbibliotek**
  Biblioteken tillhandahåller funktionalitet för uppgifter såsom design av användargränssnitt, säkerhetshantering, nätverkskommunikation osv. De tre största klassgrupperna är ADO.NET (ActiveX Data Object) för data manipulation, ASP.NET för webb applikationer och XML(Extensible Markup Language) webb services, och Windows Forms för att bygga Windows baserad klient applikationer.

Dessa komponenter var designade från början att inbegripa en bred och flexibel mängd olika säkerhetsmöjligheter som kan implementeras av utvecklare, administratörer och användare. Lite kort så består säkerhetsfunktionerna av följande:

- **Rollbaserad säkerhet**
  Tillhandahåller en enhetlig modell för autentisering och verifierar användare baserade på identitet och roller. Autentisering innebär att undersöka referenser, såsom användarnamn och lösenord, och fastställa att användaren verkligen är den han eller hon säger sig vara. När det är fastställt vem det är så kan applikationskoden fastställa vilken roll en användare har, och vilka operationer hon/han får göra. Ramverket stöder också de vanliga aututencieringsprotokollen, såsom Kerberos, Digest, SSL/TLS klient certifikat.

- **Teckenbaserad och kod säkerhet**
  Administratörer kan bestämma vilka resurser som kan kommas åt med vilken sorts kod(ex. kod som finns i en katalog, kod som kommer från Internet eller Intranät, kod som bär på en särskild hash värde eller kod som bär på en särskild autentiserings signatur eller signerad med en viss nyckel). Det här lagret av säkerhet ger administratörer bra kontroll över de maskiner som finns i deras domän och skyddar dem mot potentiell farlig kod som det inte går att lita helt på.

- **Kryptografi**
  Ramverket i .NET innehåller funktioner för kryptering, digitala signaturer, hashing och framkallning av slumpmässiga nummer. Algoritmer som stöds inkluderar symetrisk kryptering, asymmetrisk kryptering, XML digitala signaturer, och hashvärden.

Högskolan Dalarna 2006
Examensarbete Informatik 10p C
Hassan Dundar IT3 Showan Ramazani IT3

Säker inloggning med ASP.NET
Lösenorden behöver skyddas på klientsidan, mellan klient och server, samt på serversidan. Att skydda lösenorden på klientsidan och mellan klient och server är oftast rätt så enkelt att lösa med html/http-baserade lösningar: Se till att textboxen som användaren använder för lösenord har typen password (så att bokstäverna inte syns när de skrivs in), använd post-metoden (så lösenordet inte syns i QueryString) och kryptera överförningen mellan klient och server (https). För att lösenord skall kunna användas för autentisering/inloggning måste lösenordet sparas någonstans. Det måste kunna gå att jämföra det lösenord som en användare har skrivit in med det korrekta lösenordet när en inloggning skall ske. En viktig säkerhetsåtgärd är att skydda det korrekta lösenordet som finns sparat på servern. Servern är både en skyddad plats och en oskyddad. Personalen är den näst största risken idag, och endast Internet utgör en större risk. Personal kan hämnas på personer eller företaget och ställa till det.

Lösenord i ASP.NET
I ASP.NET finns det ett bra inbyggt stöd för hantering av inloggning och därmed också för hantering av lösenord. Ett standardsätt att spara lösenorden i ASP.NET är att infoga en user-tagg i web.config-filen. I denna tagg skall användarnamn och lösenord skrivas in på följande sätt: <user name="olle" password="bil" />. Problemet med detta är att alla som får tag på web.config-filen omedelbart kan se vilket lösenord en användare har. Men för att använda det inbyggda sättet att spara ett lösenord måste det sparas i web.config-filen. I web.config filen går det att specificera i vilket format lösenordet skall skrivas. I exemplet ovan är lösenordet inskrivet i klartext. Det finns dock två andra tillåtna format: sha1 och md5. Detta är namn på två olika Hash-algoritmer som finns implementerade som metoder i .NET framework.

En Hash-metod är en metod som kastar om och byter ut tecken i en teckenföljd och på så sätt göra teckenföljden oläsbar. I en säker Hash-metod skall det vara mycket svårt att reversera processen eller att gissa sig till hur den orginala teckenföljden såg ut baserat på den Hashade teckenföljden. För att kunna se sitt lösenord finns det en inbyggt funktionalitet för detta i .NET framework i klassen FormsAuthentication som finns i namespacet System.Web.Security. Metoden används på följande sätt: För att inloggningen skall fungera måste ASP.NET få veta att lösenordet är Hashat. Detta skrivs in i Credentials-taggen. Detta behövs för att inloggningen skall fungera. Nu kommer inloggningsmetoderna att använda samma Hash-metod som den som användes när lösenordet gjordes oläsbart. En jämförelse sker alltså mellan en Hashad version av det korrekta lösenordet och en Hashad version av inmatat lösenord istället för att jämföra lösenorden direkt. Självklart går det aldrig att garantera att ingen kommer åt lösenordet.

Lösenord i databasen
I dagens applikationer i den stora, verkliga världen används oftast databaser för att spara lösenord. Tillvägagångssättet för att skydda lösenorden i databasen är detsamma som tidigare. Det Hashade lösenordet sparas i databasen (trots det något långa men mycket beskrivande metodnamnet går det att använda metoden till att Hasha ett lösenord som inte skall sparas i web.config-filen). Vid inloggningen görs en jämförelse mellan ett Hashat värde av det lösenord som användaren har skrivit in.

En nackdel med Hashade lösenord
En nackdel med Hashade lösenord är att det inte går att erbjuda en service som många efterfrågar idag när det går att logga in på så många olika webbplatser på Internet. Det är viktigt att tänka på att det inte är säkert att skicka lösenordet till en användare per e-post. Om

det är viktigt med hög säkerhet bör lösenordet i så fall krypteras så att endast mottagaren kan läsa lösenordet. Eftersom det inte går att få fram lösenordet (endast det hashade värdet har sparats i databasen), måste ett nytt lösenord automatgenereras och skickas, antingen med e-post (krypterat) eller med vanlig post till den användare som glömt bort sitt gamla lösenord. Nackdelen för användaren är att han/hon får ett lösenord som kan vara svårt att minnas. Men fördelen är att användarna kan då byta sina lösenord på detta sätt.

Hashade lösenord i andra språk
Principerna för lösenordshantering är desamma för alla språk. Detta innebär att det går lika bra att hasha lösenorden i VBScript, Java, C, Perl och alla andra språk som finns tillgängliga. Om man väljer att hasha lösenorden är det viktigt med vilken metod som används. Sha1 och MD5 bedöms idag vara mycket säkra och finns implementerade i många olika språk. Det finns andra hash-metoder men dessa är antigen relativt färska (dvs inte tillräckligt beprövade) eller mindre säkra än Sha1 och MD5. Sha1 ger en längre hash sträng än MD5.[22]

Olika inloggningsformer
I ASP.NET finns det nya funktioner som gör att en hemsida kan göras mycket säkrare med inloggning och utloggning. Det finns tre olika former av autentifikation eller verifiering, dessa är: Forms, Passport och Windows. Forms baserade autentifikation sparar användarens uppgifter i en extern källa som som en databas eller i en XML fil. Windows baserade applikationer använder själva hemsidan som användaren är på för att tillåta inloggning eller inte, Passport är ett sätt som Hotmail använder sig av, där användaren använder sig av deras Microsoft Passport konto. Form baserade säkerhet är bättre eftersom det ger mer säkerhet än t ex Passport, eftersom data kan sparas på en databas t ex, och är bättre än Windows eftersom det bäst lämpar sig för intranät. Användarinformation sparas bäst i en SQL server databas, om det kommer bli en stor informationsmängd. Det skulle också gå att lagra information i en XML fil eller i själva web.config filen. Vad som händer är att en cookie skapas för att göra det enklare nästa gång användaren loggar in igen, men cookien har även en funktion när användaren loggar ut.[23]
Cookien som nämndes tidigare gör det alltså enklare för en användare att logga in på samma sida igen, eftersom dess uppgifter är sparade i cookien. När användaren trycker på OK knappen och skickar iväg en förfrågan för att logga in på sidan, så får antingen användaren ett ”ja” eller ett ”nej”. Om förfrågan är ok och användaren är identifierad så skickar systemet ut en cookie som innehåller uppgifterna eller nyckel för att kontrollera identiteten med cookie som lagrats. Förfrågan som görs av användare är auktoriserade och autenticierade av en ASP.NET händelse hanterare som använder det säkerhetesvalidition som förfrågan behöver hanteras i. [24]

Som nämnts tidigare så finns det flera olika inloggningssätt där olika säkerhets tänkanden finns, bla windows, passport och forms. Forms autentifikation är en av de viktigaste funktionerna i nya ASP.NET, eftersom en administratör kan specificera vilka filer som kan nås på en hemsida och av vilka personer, samt tillåta login och logout. Det går till på så sätt att när det kommer en förfrågan får en användare om att hämta en viss information från en viss hemsida så tittar ASP.NET först på om användarens autentifikations cookie är giltigt eller

[22] **http://www.pellesoft.se/area/articles/article.aspx?artid=652**
[23] **http://www.dotnethero.com/hero/authentication/formsbased.aspx?nmx=5_4**
[24] **http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconthecookieauthenticationprovider.asp**

inte. Om detta är giltigt så är användaren verifierad och godkänt och tillåts att komma åt informationen. Om inte cookien är rätt så skickas personen automatiskt till en login sida och om lösenordet och namnet är rätt så skapas en rätt autentifikations cookie som säger att användaren har rätt att gå in på denna sida. Om inte lösenordet är rätt så skickas användaren tillbaka till login sidan igen (se bild 3.2.1.1)[25]. Cookien ger användaren tillåtelse att surfa omkring på den skyddade sidan utan att för vissa delar, behöva uppge sin identitet igen i from av lösenord och namn. Cookien kan också kontrolleras så att den existerar en viss tid, så att säkerheten höjs, eftersom när tiden går ut måste användaren logga in igen.

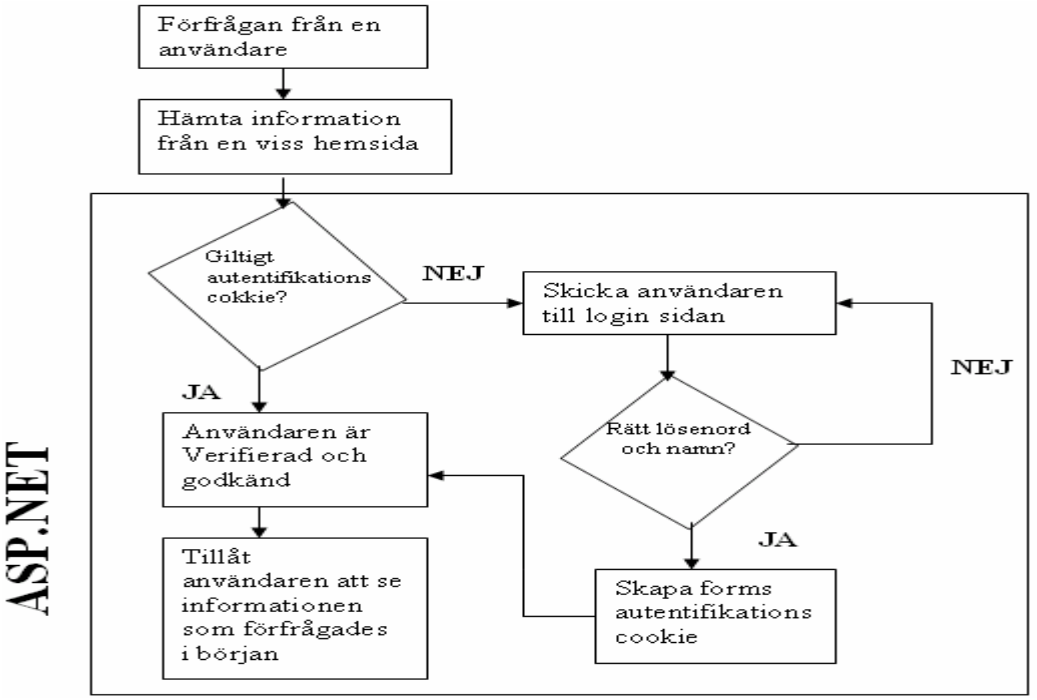


Bild 3.2.1.1 . Överblick över inloggning med forms säkerhet i ASP.NET

Som nämnts tidigare så kan information också sparas i antingen en XML fil eller i web.config filen. I teorin låter detta bra, men i verkligheten när det oftast är stora mängder information som måste sparas, då blir det oftast någon form av databas som informationen sparas i.

De flesta sidor som använder sig utav formbaserad autentifikations säkerhet, tillåter också sina inloggade användare att logga ut efter att de är klara med sin uppgift Metoden för att logga ut i FormsAuthentication gör motsatsen, den loggar ut en verifierad användare. Det fungerar på så sätt att den skickar tillbaka en Set cookie som sätter cookiens original värde vid inloggningen till en null sträng och sätter cookiens utgångsdatum till ett redan passerat datum, som då effektivt förstör autentifikations cookien. Detta får till följd att nästa gång användaren kommer tillbaka till sidan måste de logga in igen, därmed hög säkerhet.

Nackdel

En nackdel med forms baserade autentifikationer är att den bara skyddar ASP.NET filer och

[25] **http://www.awprofessional.com/articles/article.asp?p=351414&rl=1**

liknande filer som är registrerade till ASP.NET. Filer tillhörande HTML eller HTM skyddar den inte. [26]

3.2.2 Java

Java som är ett objektorienterat programspråk skapades 1991 av den kanadensisk-amerikanske programmeraren James Gosling som jobbade på Sun Microsystems. Han fick senare sällskap av flera andra och tillsammans började de utveckla det nya programmeringsspråket. Ett första namn på detta var Oak, pga ek trädet som vid den tiden fanns utanför en av gruppens medlemmars arbetsrum, men senare ändrades detta pga att namnet redan var upptagen, och döptes till java. Ursprungligen var tekniken avsedd för konsumentelektronik, som t ex i brödrostar. [27]
1995 började Java att användas på Internet då Netscape började använda det i sin browser. Med detta kunde Java användas för att skapa interaktiva sidor på Internet. Program skrivna i Java är plattformsoberoende. Det innebär att de fungerar lika bra på en Unix arbetsstation som på en PC, såvida de inte är specifikt skrivna för en sorts plattform. Det enda kravet för att de ska kunna köras är att användaren har en webbläsare som kan hantera Java-kod (att den har en javatolk). En applet är en säker tillämpning, ryktena om att de skulle sprida virus är överdrivna. Den kan inte komma åt användarens hårddisk för att läsa någon fil utan att systemet tillåter den. Men visst finns risken det ska inte förnekas. Du kan avaktivera stödet för java i din browser om du är osäker. Sun har släppt tre större versioner:
Java 1.0.2, Java 1.1.5 och nu senast Java 2. [28]

Java språket, Java virtuella maskin och Java plattformen
När Java nämns brukar det handla om tre saker:

- Java objektorienterat programmeringsspråk
- Java virtuella maskin
- Java plattformen

Java språket
Java som är ett av flera objektorienterade programmeringsspråk bygger på tidigare språk som C++ och C. När Java utvecklades ville man självklart att användarna skulle känna igen sig från tidigare språk, men att Javan fått utökade funktioner. Java är i nuläget det mest använda programmeringsspråket och används både på Internet på hemsidor med animeringar av olika slag, och i mobiler där korta spel kan utvecklas och spelas med Java tekniken. Det som har gjort Java så populär är att det finns tillgång till gratis verktyg för programmet. Java så populärt är att det finns tillgång till gratis verktyg för programmet. Java är också plattformsoberoende, som betyder att den kan köras på olika plattformar som Linux, Windows el Mac t e x.

Java virtuella maskin
Det specifika med Java-tillämpningar (applets) är att de inte är skrivna för en viss plattform, utan för en virtuell maskin. Den virtuella maskinen kan installeras på Unix, Windows eller Powermac och kör Java-tillämpningen som en tillämpning för den maskin som den virtuella maskinen är installerad på. Den virtuella maskinen har dessutom flera likheter med ett

[26] **http://msdn.microsoft.com/msdnmag/issues/02/05/ASPSec2/**
[27] http://www.ne.se/jsp/search/article.jsp?i_art_id=215694&i_word=java
[28] http://www.tiger.se/dok/java_text.html

operativsystem och begränsar den därmed inte till att köra Java-tillämpningar.[29]

Java plattformen
Java har tre plattformar, standard versionen (Standard Edition), Enterprise Edition (EE) och Mikro versionen (Micro Edition).Varje plattform har en beskrivning av vilken språkversion de ska använda och vilket standardbibliotek och till sist vilken virtuell maskin som ska kunna användas för att exekvera en kod. Enterprise versionen (EE) är än förbättrad version av standard versionen (SE). Mobiler och andra små liknade apparater har gjort att Java Mikro versionen (ME), skiljer sig från de andra två.[30]

Säkerhet
Bakom Java plattformen finns en dynamisks och omfattande säkerhetsarkitektur. Säkerhetsfunktioner, som t ex kryptografi, PKI och autentisering finns inbyggda i plattformen. Säkerhetsmodellen för Java är baserad på en skräddarsydd modell där Java program kan fungera utan potentiella risker för systemen eller användare.

Nedan beskrivs de olika säkerhetsfunktionerna lite närmare.

Plattformssäkerhet
Inbyggda funktioner för säkerhet runt språk stöds av Java Kompilatorn och virtuella maskinen:

- Automatisk datahantering.
- Bytecode verifikation.
- Säker laddning av klasser

Fördelar

- Tillhandahåller en säker plattform för utveckling och för att köra applikationer på.
- Automatisk datahantering leder till mer robust kod och minskar ner på fel på minnet och sårbarhet.
- Bytecode verifikation ser till att fientlig kod inte förstör exekveringen. Klassladdare ser till att opålitlig kod inte lägger sig i andra Java program som körs.

Kryptografi
Omfattande API med stöd för ett stort utbud av kryptografiska tjänster såsom digitala signaturer, chiffer (symmetrisk, asymmetrisk etc.), koder för verifiering av meddelande, nyckel generatorer och nyckellager.

- Stöd för en hel del standard algoritmer.

Fördelar

- Ger en förlängbar, full funktionell API för att bygga säkra applikationer.
- Algoritm och implementations oberoende
- Leverantör baserad arkitektur

Autentisering

[29] http://domino.idg.se/cs/artikel.nsf/0/f6e365be7ecdd4e8c12564de0056e926?OpenDocument
[30] http://www.onjava.com/pub/a/onjava/2006/03/08/what-is-java.html , Internetartikel

Abstrakt autentisering av API:er som kan lägga in ett brett utbud av mekanismer.

- En omfattande policy och permissions API:er som tillåter utvecklaren att skapa och administrera applikationer som kräver särskild tillgång till säkerhetskänsliga resurser.

Fördelar
Möjliggör singel signering av flera autentiseringsmekanismer och särskild tillgång till resurser som är resurser baserade på användarens identitet och kod. I nyare versioner finns det stöd för timestamp (en sorts tidskod) som gör det enklare att placera ut signerad kod genom att undvika att behöva signera om kod då signerarens certifikat går ut.

Säker Kommunikation
Java innehåller API:er och implementationer för följande kommunikationsprotokoll:

- Transport Layer Security (TLS)
- Secure Sockets Layer (SSL)
- Simple Authentication and Security Layer (SASL)
- Support för HTTPS över SSL/TLS

Fördelar
Autentiserar datorer som kommunicerar över ett opålitligt nätverk och skyddar integriteten och sekretessen på det data som skickas mellan dem.

Public Key Infrastructure (PKI)
Verktyg för att hantera nycklar och certifikat samt omfattande API:er med support för en mängd funktioner och algoritmer.

Fördelar
Underlättar utvecklingen och utplaceringen av komplexa PKI applikationer. I dagens läge så stöds OCSP (Online Certifikate Status Protocol) vilket ger en mer anpassningsbar och läglig metod till applikationer för att kontrollera giltighetstiden på certifikat.

Olika säkerhets protokoll: *JSSE, JAAS och JCE*
De säkerhetsfunktioner som erbjuds av Java, är avsedda för en mängd olika åskådare däribland slutanvändaren och tillverkaren. För användare finns det inbyggd säkerhet som förhindrar att farliga program där virus kan finnas körs medan användaren har privata filer och information på samma ställe. I javans version 1.2 kan säkerhetsfunktioner tas fram när användaren önskar detta. Utvecklare kan använda API:s (Application-programming interfaces), som är applikations programmerings interface som kan användas för att säkra program. Strukturen i API:en gör det möjligt för t ex administratörer att definiera och sedan integrera säkerhet för att kunna kontrollera tillgången till olika platser. Detta kan vara att säkerhetställa identiteter och godkänna dessa, kryptografi och olika slags implementationer av linjer som bör följas.

Det finns tre viktiga protokoll att titta på i Javan när det gäller säkerhet, dessa är JSSE, JAAS och JCE. JCE står för Java cryptology Extension och tillåter kryptering, nyckel generering, samarbete, verifiering av meddelande, samt verifiering av kod. För att kunna ha använda Internet säkert så har Javans JSSE (Java secure socket Extension) hjälp av krypteringsprotokollen SSL och TLS i sitt program. Detta gör det då möjligt att använda http,

Telnet, NNTP och FTP och användarna kan försäkra sig om att data som passerar mellan klient och server är säkrad mot obehöriga. [31] Det tredje och sista protokollet heter JAAS (Java Authentication and Authorization service) och står för identifiering och tillåtelse av användare när de begär någon form av hemlig information från en skyddad sida. Protokollet är också en oberoende applikation, eftersom den inte är beroende av underliggande applikationer eller säkerhetsprotokoll. Dessutom kan ny teknik lätt läggas till utan att behöva ändra i JAAS protokollet.[32]

[31] **http://yunus.hacettepe.edu.tr/~tonta/courses/fall2002/kut655/13-infrastructure-frame-twitchell.htm**
[32] **http://java.sun.com/products/jaas/overview.html**

3.4 Alternativa ljudinspelningsmetoder

För att kunna spela in telefonsamtal krävs det tillstånd, och i den svenska lagstiftningen står det så här angående detta, ” 8 **§** Varje svensk medborgare och svensk juridisk person har rätt att framställa och sprida tekniska upptagningar. För rätt att offentligt förevisa en film, ett videogram eller någon annan teknisk upptagning med rörliga bilder kan det dock krävas granskning och godkännande i enlighet med 1 kap. 3 § andra stycket. Lag (*1998:1439*).” Det här ger rätten till individen att t ex spela in ett samtal om denne vill. Det står vidare, ”Med tekniska upptagningar avses i denna grundlag upptagningar som innehåller text, bild eller ljud och som kan läsas, avlyssnas eller på annat sätt uppfattas endast med tekniskt hjälpmedel ”.[33] Det är alltså tills vidare lagligt att spela in telefonsamtal i Sverige om ena parten är medveten om det. Det är viktigt att förstå att det är speciella regler för myndigheter, men detta tas inte upp här alls. Tekniken finns redan här för att kunna spela in samtal och det finns en mängd olika tekniska lösningar på detta. Frågan är bara hur mycket en person är beredd att betala för att få ett bra system för inspelning av telefonsamtal. Det finns flera företag som har specialiserat sig på att utveckla teknik för att spela in samtal från antingen telefoner, mobiltelefoner eller från en mikrofon till antingen en pc eller annan inspelningsbar källa. Företagen heter Vidicode, Retell och Multisun. Videcode (Nederländerna) har utvecklat en serie produkter för inspelning av telefon- och faxsamtal. Vissa kan lagra stora mängder samtal och även anslutas till nätverk så att inspelningar kan lagras i en central Arkiv databas. För att öka säkerheten mot obehörig avlyssning kan i vissa säkerhetskort användas. Samtal under inspelning kan avlyssnas av behöriga via nätverket, Call Monitoring. För att spela in ett telefonsamtal behövs en adapter som tar emot samtalen och någon sorts programvara som kan användas för att t ex spara samtalen på datorn. Multisun som är Taiwanesiskt har utvecklat teknik som gör det möjligt att spela in från mobiltelefoner till antingen en pc eller en bandspelare t ex.[34]

Voice Mail
Uppfinnaren till röst mail var Gordon Mahews, som var en framgångsrik företagare som vid sin död den 22 februari 2002 hade 35 amerikanska och utländska patenter. Det var mot slutet av 1970 talet som Gordon började utveckla denna teknik. Samtidigt startade ett företag som hette VMX (voice message express) röst meddelandes express. Han fick patent på detta och sålde företaget till 3M. 1989 lämnade han företaget och företaget såldes vidare, men fortfarande finns det företag än idag som använder VMX system för att möta behovet på röst mail.
I början på 1990 talet när detta presenterades för världen så var det inte många företag som kunde utnyttja detta. VMX systemet var väldigt dyrt och därför var det bara de största företagen som hade råd. Detta jämnade ut sig i början av 1990 talet då röst mail blev tillgänglig för den stora massan som hade pc. Detta var tack vare företaget Dialogic

Communications Corporation grundades 1982 och blev snabbt det ledande företaget inom tillverkningen av mjukvara för röst mail. [35]

[33] http://www.notisum.se/Rnp/SLS/LAG/19911469.htm
[34] http://www.telero.se/inspelning.html
[35] http://www.800voicemailstore.com/voice-mail-history.htm

Voice mail eller röst mail på svenska låter en person att skicka ljudmeddelanden istället för vanliga skriftliga e-mail. Det enda som behövs göras är att tala in i mikrofonen och skicka iväg röst mailet. Ljuden blir i mp3 format vilket gör att filen blir liten och ljudkvalitén bättre. Inga känsloikoner behövs längre eftersom det går att utrycka allt med rösten. Längden på en röst mail är obegränsad och behöver ingen annan e-mail klient. [36]

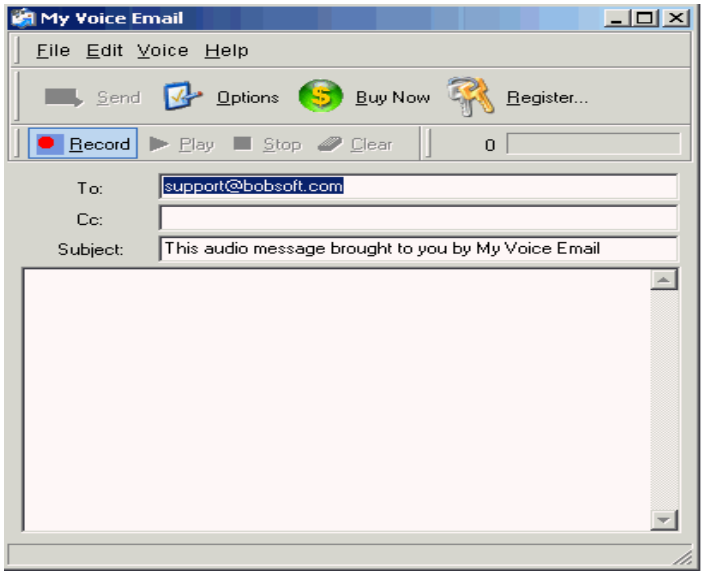


Bild 5.1 exempel på hur ett voicemail gränssnitt ser ut

IP Telefoni [37]

Fram till för några år sedan fanns det egentligen bara två olika slags sätt att ringa på: fast telefoni och mobiltelefoni**.** Den fasta telefonin har funnits i drygt hundra år och tekniken kallas PSTN (Public Switched Telephone Network) eller POTS (Plain Old Telephone Service). Mobiltelefonin har vid det här laget funnits i några decennier. Idag är den fasta telefonin mer eller mindre på väg bort från marknaden, precis som analoga kameror och analoga TV-sändningar. Anledningen är att man idag vill kunna göra mer än att bara prata i telefon: skicka sms och mms, skicka datafiler och bilder, lagra meddelanden på nya sätt, kanske surfa på Internet eller hämta information av olika slag. Den nya telefonmarknaden svämmar för tillfället över av nya sätt att ringa på och det är inte bara mobiltelefoni som blir allt mer etablerad, utan även den telefoni som går under samlingsnamnet IP telefoni. IP telefoni kallas också bla Internet telefoni, bredbandstelefoni, VoIP mm.

IP och digital paketering

IP är ett slags adresseringssystem som används i nätverk för att kunna överföra data, t.ex. skicka meddelanden, till rätt adress. Det samverkar med andra kommunikationsprotokoll (standarder), t.ex. TCP, för att kontrollera att datan skickas i rätt form. Först användes tekniken till att skicka textfiler digitalt, men under 90-talet blev multimediafiler - ljud, filmer, videosnuttar - allt vanligare. Det gick även att skicka telefonsamtal på detta sätt. Överföringen sker antingen analogt eller digitalt. Analogt innebär en kontinuerlig ström i en ledning. När du pratar är det därför upptaget, eftersom hela ledningen används. Digital överföring innebär att informationen (via TCP-protokollet) delas upp i små paket, som därefter skickas med längre

[36] http://www.flashpeak.com/mve/
[37] http://www.jonasweb.nu/sidor/datorn/mobilt/iptelefoni.html

avbrott eller mellanrum. Detta gör att andra data kan skickas i de mellanrum som uppstår och överföringskapaciteten ökar.

Den fasta telefonin var länge helt analog, men under 80- och 90-talen byttes analoga telestationer successivt ut mot digitala. Anledningen var naturligtvis att utnyttja telenätet bättre utan att behöva dra alltför många nya kablar. Idag är den analoga tekniken bara kvar mellan telestationen och abonnentens telefon. Med IP-telefoni blir dock även denna del digital. IP telefoni fungerar så att när en person pratar omvandlas röstens vibrationer till digitala signaler med en liten manick i telefonen. Den analoga informationen (luftens vibrationer) omvandlas till ettor och nollor, som utgör bitar i de datapaket som skickas. De minsta paketen utgörs av åtta ettor eller nollor och kallas för bytes. De små paketen skickas sedan i skurar (snarare än en jämn ström) allt eftersom det finns lediga platser. Adresseringen sker via IP-protokollet. I bärbara telefoner och mobiltelefoner skickas signalerna via radio. När signalen når basenheten övergår den till en elektrisk ström.

VoIP
Ibland går det att se uttrycket VoIP eller ”Voice över IP”. Då handlar det om telefoni över nätverk, som både kan ske via Internet eller via interna nätverk, t.ex. på ett företag. VoIP förekommer nästan inte på konsumentmarknaden utan är istället en teknik för telefonväxlar**,** stora företagsnätverk och liknande. Det skall också sägas att uttrycket "VoIP" ibland används synonymt med IP-telefoni, internettelefoni eller bredbandstelefoni. Anledningen till detta är att begreppen ibland överlappar och det finns sällan någon exakt definition. Fördelen är att det ger billigare samtal jämfört med att ringa via mobilen eller på vanlig telefon. Nackdelen är att det sällan används på konsumentmarknaden. Kräver telefonväxel och är ganska dyrt att installera.

Internettelefoni
Internettelefoni innebär att ringa via Internet, oftast via sin dator, med en särskild programvara installerad. Mottagaren måste ha samma program på sin dator. Programmen kallas allmänt för mjukvarutelefoner. Via ett headset går det sedan att prata med varandra och samtalen kostar inget förutom kostnaden för bredbandet. Exempel på sådana lösningar är MSN och Skype. MSN innebär kommunikation mellan dator till dator, medan Skype möjliggör alla tre formerna av kommunikation:

- Dator till dator
- Dator till telefon
- Telefon till telefon

Med "telefon" menas här såväl fast som mobil telefon. Utrustningen som krävs är normalt tillgång till en dator, ett någorlunda bra ljudkort, bredbandsuppkoppling och ett headset, som kopplas till ljudkortet. Det finns också särskilda telefoner som det går att ansluta till datorn via usb-porten med. De går därför ofta under namnet USB-telefoner. En del av telefonerna är istället trådlösa, där basenheten ansluts till datorn. Fördelen med internettelefoni är att eftersom det inte finns någon uppkopplingsavgift, minutavgift eller abonnemangskostnad ger tekniken billiga samtal som oftast inte kostar något alls. Det är enkelt att komma igång med Internettelefoni. Nackdelarna är att det kräver bredbandsuppkoppling och en dator med en särskild mjukvara.

Bredbandstelefoni

Idag erbjuder såväl Bredbandsbolaget, ComHem och andra leverantörer IP-telefoni via bredbandsmodemet till sina kunder. Den vanliga telefonen kopplas då, med RJ 11-plugg, till bredbandsmodemet och samtalen förmedlas via modemet och Internet. Eftersom det finns många olika slags telefoner kan min användning av uttrycket "vanlig telefon" vara förvirrande. Vad jag menar är t.ex. Dect-telefoner. DECT står för "Digital European Cordless Telephone" och är en EU-standard för digitala trådlösa telefoner. En nackdel är att telefonen inte fungerar då modemet är avstängt, t.ex. vid strömavbrott. Fördelarna med bredbandstelefoni är att det ger billiga samtal, kräver ingen dator, enkelt att komma igång och det går att använda sin vanliga, fasta telefon. Nackdelarna är att det kräver bredbandsuppkoppling och det går inte att ringa vid strömavbrott.

4. Analys

Syftet med detta arbete är förändring av företagets system genom systemutveckling.
För att få bästa tänkbara säkerhet runt webbplatsen kommer vi göra en jämförelse mellan olika tekniker för att få fram den teknik som passar målet bäst.

SSL, TLS och S-HTTP
De tekniker som kommer att analyseras är SSL, TLS och S-HTTP. Alla tre är säkerhetsprotokoll med olika slags funktioner och krypterings möjligheter. Det mest välkända protokollet är SSL (Secure Sockets Layer) och används flitigt idag av många företag och institutioner runt om i världen. Detta protokoll har mer eller mindre ersatts av ett annat, nämligen TLS (Transport Layer Security), trots detta används SSL fortfarande. TLS är det nya, förbättrade krypteringsprotokollet som ersätter SSL. S-HTTP är ett annat protokoll, men som inte är lika känt eller lika använt som SSL eller TLS. Här krypteras bara enstaka meddelanden, och inte under en hel session mellan klient och server. Även om S-HTTP från början skulle bli ett starkt krypteringsprotokoll så glömdes detta bort med tiden, eftersom SSL blev mer och mer accepterad och detta gjorde att föregående protokoll hamnade i dess skugga.

Eftersom SSL fortfarande används ganska flitigt som krypterings protokoll, så kan detta användas på klient sidan när en användare ska koppla upp sig mot I am Deads hemsida. För att detta ska fungera, måste ett certifikat köpas och installeras så att förbindelsen mellan klient och servern blir skyddad. Ett certfikat ökar inte bara säkerheten genom att tillåta kryptering, utan gör också användaren lugn eftersom denne kan se att certifikatet är äkta om han/hon klickar på hänglåset som ska synas i Internet Explorers nedre högra hörn. För att sedan skydda servern finns det möjlighet att använda hashad lösenord. Detta ger ett bra skydd mot obehöriga, men nackdelen är att ett lösenord gäller bara en gång och därefter måste ett nytt lösenord matas in för varje inloggning.

Även om införande av ett SSL protokoll skulle hjälpa ett bra tag framöver, är det viktigt att tänka på framtiden. Eftersom det alltid kommer att finnas personer som vill komma åt obehörig information, kommer alltid krypteringstekniken att gå framåt. Därför är det bra att överväga att installera TLS protokollet eftersom det är en uppdaterad och kraftigare version, även om SSL grund funktionerna finns kvar.

ASP.NET och Java
ASP.NET är en förbättrad och uppgraderad version av ASP. För att kunna köra ASP.NET på en dator, behövs det installeras IIS på den. För att få tillgång till ASP.NET runtime måste .NET ramverket installeras på datorn. Nya ASP.NET stödjer operativsystem som Microsoft Windows XP, 2000 Professional och server och Windows server 2003. NET ramverket är underlaget för alla NET teknologier t ex ASP.NET. De två största komponenterna i NET ramverket är CLR (Common Language runtime) och klassbiblioteket. CLR är motorn som exekverar koden för alla applikationer som baseras på ramverket. Klassbiblioteket tillhandahåller funktionalitet för uppgifter som design av användargränssnitt, säkerhetshantering, nätverkskommunikation osv. ASP.NET har en rollbaserad säkerhet vilket går ut på att det finns funktioner som autentiserar och verifierar användare baserade på identitet och roller. Användarnamn och lösenord undersöks så att användaren verkligen är den han/hon utger sig för att vara. Efter detta kan en applikations kod fastställa vilken roll han/hon

har och vilka operationer denne får utföra. ASP.NET stödjer också de vanligaste krypteringsprotokollen som SSL och TLS.
Trots många fördelar med ASP.NET, finns det några nackdelar. Det går inte köra NET på alla plattformar, den är med andra ord plattforms beroende. Ett annat problem var att html och JavaScript som skickades tillbaka till klientens webbläsare inte alltid validerades mot W3C (World Wide Web).

Java språket är ett programmeringsspråk som ska passa dagens behov av ny och bättre teknik. Det bygger på C++ som är ett annat programmeringsspråk. Java är plattformsoberoende, vilket betyder att den kan köras på plattformar som Linux, Windows el Mac. Säkerheten är bra i Java, eftersom det finns funktioner för kryptografi, PKI (public key infrastructure) och autentisiering. Några fördelar med Java är att automatisk datahantering leder till en mer robust kod och minskar ner fel på minnet och sårbarhet. Bytecode verifikation ser till att fientlig kod inte förstör exekveringen. Klassladdare ser till att opålitlig kod inte lägger sig i andra Java program som körs. Java stödjer också krypteringsprotokoll som SSL, TLS, Simple Authentication and Security Layer (SASL) och en funktion för HTTPS över SSL/TLS.

Både ASP.NET och Java stöder de vanligaste krypteringsprotokollen som SSL och TLS. Javan är inte plattformberoende som ASP.NET är och detta gör att Javan kan köras på flera plattformar som Mac, Linux el PC.

Inspelningsmetoder
Voicemail låter en person skicka röstmeddelanden istället för att skicka textbaserade meddelanden. Detta skulle kunna vara ett alternativ för användare att skicka in sina meddelanden. Antingen lägger de upp det själva på sin sida, eller så skickar de in det till en server som lagrar dessa och som sedan en i personalen lägger upp på respektive användares sida, då skulle det krävas att användaren på något sätt är identifierad så arbetet för personalen underlättas. Det finns gratis program som lätt kan laddas ner från Internet, alternativt ha en liten, gratis program som lätt kan laddas ner från *I am Deads* hemsida. Har användaren ingen dator så finns det möjlighet att via vanlig telefon eller IP telefoni, ringa in sin röst meddelande. Med IP telefoni menas att en person ringer via sitt bredbandsuttag. Samtalet går då över Internet, och är ett billigare sätt att ringa, men kräver dock någon form av bredbands uppkoppling.

5 Slutsatser

Säkerhetsjämförelsen som gjordes var mellan SSL (Secure Socket Layer), TLS (Transport Layer Security) och S-http (Secure-HTTP). S-HTTP är ett fungerande säkerhetsprotokoll, men används inte lika mycket som de andra två. Det mesta använda protokollet är SSL och anses vara en standard för säkerhet över hela världen. SSL har nu uppdaterats med nyare funktioner och säkrare protokoll och kallas TLS. Trots att TLS är ny på marknaden så används fortfarande SSL av många företag. Till projektet kommer SSL protokollet passa bra, eftersom det är en standard som anses vara mycket säkert och effektiv mot obehöriga världen runt. Även om TLS är en lite kraftigare version av SSL, anser vi att SSL skulle duga bra. Däremot kan TLS användas om arbetsgivaren anser att detta skulle passa bättre utifrån den teori som skrivits och analyserats.

Eftersom ASP.NET redan var något som vår grupp kände till och kunde hantera bra, ansågs det att detta skulle passa bra att bygga projektet i. Dessutom anser vi att Javan inte har lika bra funktioner som ASP.NET, och därför inte passade vårt ändamål.

Det finns flera olika sätt för en användare att ringa in ett röstmeddelande till hemsidan eller lämna dokument av olika slag till företaget *I am Dead* . Voicemail är det första alternativet och går ut på att istället för att skicka ett textbaserat meddelande, skicka ett röstbaserat meddelande. Det finns gratisprogram ute på Internet, och skulle nu inte användaren vilja leta, eller inte kan leta, t ex dåliga datakunskaper, kan det på hemsidan finnas ett enkelt program som lätt kan laddas ner till användarens dator. Denna kan sen i sin tur antingen skicka meddelanden direkt på sin sida, genom att lägga upp filer själv, eller att skicka de till företagets server för lagring. Därefter går det att tänka sig att en i personalen går igenom alla lagrade röstmeddelanden (utgår från att identifiering finns för varje person), och lägger upp meddelande en efter en på varje persons sida. Detta är inte så praktisk och svårt, eftersom det måste finnas en identitet på varje meddelande. Enklast vore nog att användaren själv la upp de dokument/röstmeddelande han/hon vill ha direkt på sidan.

IP telefoni är det andra alternativet, och går ut på att användaren via en vanlig telefon ringer via sitt bredbandsuttag, in sitt meddelande till företagets server och därefter sker processen som i ovanstående stycke. Detta skulle kunna kompletteras med att användaren lämnar personlig information om sig själv i samband med röstmeddelandet så att identifiering kan ske, och någon ur personalen kan sedan lägga upp det på hemsidan vid senare tillfälle. Eftersom datorn eller PC:n är något som finns i större delen av svenska befolkningens hem, så låter alternativ ett som en bra lösning. Det krävs inte mycket mer än en mikrofon och ett program för att kunna lägga upp en fil på sitt krypin på hemsidan. För den lilla majoritet som har svårt för en pc (bl.a. äldre personer) så är ett alternativ för dem att ringa in via en vanlig telefon, antingen via sitt bredbandsuttag (IP telefoni), eller via den normala telefonlinjen. Det enda som krävs av företaget *I am Dead* är att det installeras någon form av telefonväxel som kan hantera alla inkommande samtal och en server som sparar och lagrar både informationen om användaren och meddelanden som lämnas.

För att kunna bygga de nödvändiga applikationer och funktioner för hemsidan så använde vi oss av programmeringsspråket ASP.NET och av metoden OOAD (Objekt Orienterad Analys och Design). Vi hade också två andra metoder, RUP och UML, men ansåg att OOAD skulle

passa vårt projekt bäst. Dessutom har vi genom litteraturstudier, genom att läsa i böcker och söka på Internet, kunnat hitta övrig information som krävdes för arbetet som t ex olika säkerhets tekniker.

I början av projektet satte vi upp vissa mål:

- Skapa en normaliserad databas
- Bygga funktioner för att hantera registrering och inloggning till webbplatsen
- Bygga funktioner för översikt an användaren uppgifter och sådant som lämnats, ladda upp filer och kunna ändra uppgifter.
- Skapa en GUI för alla funktioner

Vi har byggt upp fungerande funktioner för de att hantera de olika kraven som ställdes på dem (se bilaga 6). Genom detta får arbetsgivaren fått en bra grund att stå på när det gäller vidareutveckling av affärsidén och hemsidan. Dessutom har vi jämfört och analyserat olika säkerhets lösningar som vi ansett att hemsidan behöver. Detta eftersom privat information om användare kommer sparas och därför måste säkerheten från det att användaren loggar in, till dess att de loggar ut, vara hög.

6 Källförteckning

Internetkällor

ASP.NET
Security with Microsoft .NET: An Overview (09-05-06)
http://download.microsoft.com/download/a/f/4/af49b391-086e-4aa2-a84b-ef6d916b2f08/security_net.doc

Microsofts webbsida för utvecklare i olika Microsoft produkter (07-05-06)
http://msdn.microsoft.com

Webbsidan för ASP.NET och allt runt det (07-05-06)
http://www.asp.net/

Artikel om ASP.NET (12-05-06)
http://www.ondotnet.com/pub/a/dotnet/2005/09/19/what-is-asp-net.html?page=4

SSL + Säkerhet
http://www.if.fi/web/fi/private.nsf/0/E9BEB53EB8C51B3BC1256FAA003C5BA0 (060502)
http://www.verisign.com/advisor/check.html.(060502)
http://www.verisign.se/products-services/security-services/ssl/ssl-information-center/how-ssl-security-works/index.html (00502)
http://www.verisign.se/products-services/security-services/ssl/ssl-information-center/faq/ssl-encryption.html#01000003 (060502)
http://www.cs.umu.se/kurser/TDBD07/VT00/rapp/rapp1.htm (060502)
http://www.webopedia.com/TERM/S/SSL.html (060502)
http://www.openssl.org/(060502)

SSl certifikat
http://isp.surfnet.fi/productsprivate.php?Action=showProduct&productid=40&e=75;70
(060503)

Kryptering
http://www.intranetica.com/intranetica/kds/krypto.shtml (060504)

Lösenord
http://www.pellesoft.se/area/articles/article.aspx?artid=652#1 (060506)

Datainspektionen
http://www.datainspektionen.se/lagar/pul.shtml (060507)

S-HTTP
http://www.linktionary.com/s/shttp.html (060508)

TLS
http://www.nyphp.org/content/presentations/SSL/tls.html (060509)
http://www.networkworld.com/details/789.html (060509)
http://publib.boulder.ibm.com/infocenter/cicsts/v3r1/index.jsp?topic=/com.ibm.cics.ts.doc/dfhe4/SSL/transportlayersecurity1.0protocol.htm (060509)

Java
http://java.sun.com/security/index.jsp (14-05-06)
http://developers.sun.com/techtopics/mobility/apis/articles/satsa2/index.html (14-05-06)
http://www.ne.se/jsp/search/article.jsp?i_art_id=215694&i_word=java (15-05-06)
http://www.tiger.se/dok/java_text.html (15-05-06)
http://domino.idg.se/cs/artikel.nsf/0/f6e365be7ecdd4e8c12564de0056e926?OpenDocument (16-05-06)
http://www.onjava.com/pub/a/onjava/2006/03/08/what-is-java.html Artikel Chris Adamson (060520)

Inloggning med ASP-NET
http://www.pellesoft.se/area/articles/article.aspx?artid=652 (29-05-06)
http://www.dotnethero.com/hero/authentication/formsbased.aspx?nmx=5_4 (29-05-06)
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconthecookieauthenticationprovider.asp (29-05-06)
http://www.awprofessional.com/articles/article.asp?p=351414&rl=1 (29-05-06)
http://msdn.microsoft.com/msdnmag/issues/02/05/ASPSec2/ (29-05-06)

Inloggning med Java
http://java.sun.com/products/jaas/overview.html (29-05-06)
http://yunus.hacettepe.edu.tr/~tonta/courses/fall2002/kut655/13-infrastructure-frame-twitchell.htm (29-05-06)

Olika inspelningsmetoder
http://www.jonasweb.nu/sidor/datorn/mobilt/iptelefoni.html (01-06-06)
http://www.cisco.com/univercd/cc/td/doc/product/voice/sw_ap_to/apps_3_0/english/admn_app/getst303/gsmain3.htm (01-06-06)

Litteratur

Andreas Halvarsson, Tommy Morin. *Elektroniska signaturer- E-affärer utan elände med identifiering, signering och kryptering*, Studentlitteratur, 2000.

Predarag Mitrovic. *Handbok i IT-säkerhet 2:a upplagan*. Pagina, 2002 Göteborg.

Eric Maiwald, William Seiglein, *Datasäkerhet i praktiken.* Pagina, 2002 Göteborg.

Lars mathiassen, Andreas Munk-Madsen, Peter Axel Nielsen, Jan STage, *Objektorienterad analys och design.* Studentlitteratur, 2001

Bilaga 1, begreppslista

Uniform Resource Locator (URL)

URL är en form av resurspekare, en adress inklusive protokollet som används för att kommunicera via. Detta kan till exempel vara en webbsida som http://www.google.com (protokollet är http och adressen www.google.com).

Hyper Text Transfer Protocol (HTTP)

Med denna text i adressfältet i webbläsaren talar du om för din webbläsare att den ska anropa en webbserver. Om http inte fylls i, brukar webbläsaren automatiskt lägga till detta

World Wide Web (WWW)

När du skriver www framför domännamnet i din webbläsare anropar du efter en webbsida som finns på en webbserver. www skulle egentligen kunna heta vad som helst. www är dock det vanligaste datornamnet/hostnamnet som används och det brukar visa att domännamnet har en hemsida.

WEBBLÄSARE/WEBBROWSER

Exempel på webbläsare är Internet Explorer eller Netscape. Det är program som laddar ner hemsidan för att ni ska kunna se på olika hemsidor ("surfa på nätet").

HEMSIDA,WEBSITE, SAJT, WEBBPAGE

Alla dessa benämningar är den sida/information du får upp när du fyller i ett domännamn i adressfältet på din webbläsare/webbrowser, t ex www.aftonbladet.se

TCP/IP

Om två människor ska kommunicera måste det tala samma språk. På samma sätt måste två datorer ha ett gemensamt språk, ett protokoll, för att kommunicera i ett nätverk.
Det finns en uppsättning protokoll som kalls TCP/IP. Namnet är lite förvirrande, protokollsamlingen är nämligen döpt efter sina två viktigaste protokoll, TCP och IP , men det finns många fler protokoll i TCP/IP protokollsamlingen.

IP

IP är förkortning för Internet Protokoll. Namnet ger en antydan om IP:s betydelse för Internet. IP protokollet är basen för allt kommunikation på Internet, alla data som skickas på nätet går via IP.

TCP

TCP är ett protokoll som sköter större delen av alla dataöverföringar på Internet. Namnet är förkortning av Transport Control Protokoll och erbjuder pålitliga överföringar och garanterar att data kommer fram i den ordning det skickas.

FTP

File Transfer Protocol. FTP är ett överföringssätt (protokoll) som i sin tur använder TCP/IP. FTP kan sägas vara nätverkens DIR och COPY-kommando.

TELNET

Telnet är ett gammalt och osäkert protokoll och används till att kontakta en annan dator, förutsatt att datorn du tar kontakt till ger dig rättighet att göra det. (En sådan dator kallas vanligen en server.)

XML

Är en uppsättning regler för hur textuella format skall konstrueras, format som låter dig strukturera dina data. XML gör det lätt för en dator att generera data, läsa data och garantera att datastrukturer är entydiga.

Common Language Runtime (CLR)

Är hjärtat i .NET. Det är det som tar din .NET-applikation, kompilerar den till plattformsspecifik processorkod och kör den. Den tillhandahåller en stor mängd funktionalitet som hjälper applikationerna att köras riktigt, så vi tar en titt på en av dem i taget.

Visual Basic

VB är enklaste och snabbaste sättet att göra egna Windowsprogram. VB introducerades 1991, fick sitt "genombrott" 1993 med version 3.0 och har idag (2002) nått version VB NET (ver 7).

DLL

Står för Dynamic Link Library och är ett bibliotek av antal funktioner som kan bli exekverade, eller data som kan användas av andra Windows applikationer. Det finns flera hundra tom tusentals DLL filer som används för olika saker. Tex comdlg32.dll som är en dll fil som innehåller Common dialog boxar som används av Windows applikationer.

Python

Är ett objektorienterat språk som är körbart på nästan alla plattformar.

C++

Är ett annat objektorienterat språk som används mest nuförtiden. Bygger på ett annat programmeringsspråk kallad C.

C

Är ett i IT-sammanhang mycket gammalt språk och härstammar från början av 1970-talet. C är alltså ett subset av C++.

HyperText Markup Language (HTML)

Är ett sidbeskrivningsspråk som används för att skriva de webbsidor som utgör World Wide Web. Som namnet antyder går det med HTML att märka upp en sidas olika delar, t.ex vad som ska vara rubrik, nytt stycke, kursiverad text och liknande.

Graphical User Interface (GUI)

Är en förkortning för begreppet grafiskt gränssnitt

Active Server Pages (ASP)

Är webbsidor innehållande script som körs och producerar ny eller förändrad html-kod genom en mix av text och html-taggar.

Hypertext Preproccessor/ personal home page (PHP)

Det är ett programmeringsspråk som har som huvudfunktion att användas ihop med webbsidor för att kunna skapa dynamiskt och interaktivt innehåll.

Servlets

Är program skrivna i Java som kompileras och körs på en server, till skillnad från Applets som körs i klienten.

Pseudorandom funktion (PRF)

Är en stark krypterings sätt.

Internet Information Server (**IIS**)

Är en webbserver från Microsoft som kan köras under Windows XP Professional, Windows 2000 och Windows NT och nu även Windows server 2003. Den kännetecknas av att den kan exekvera Active Server Pages, även om den inte är ensam om denna egenskap. Det är dock av naturliga skäl den mest kompletta implementationen.

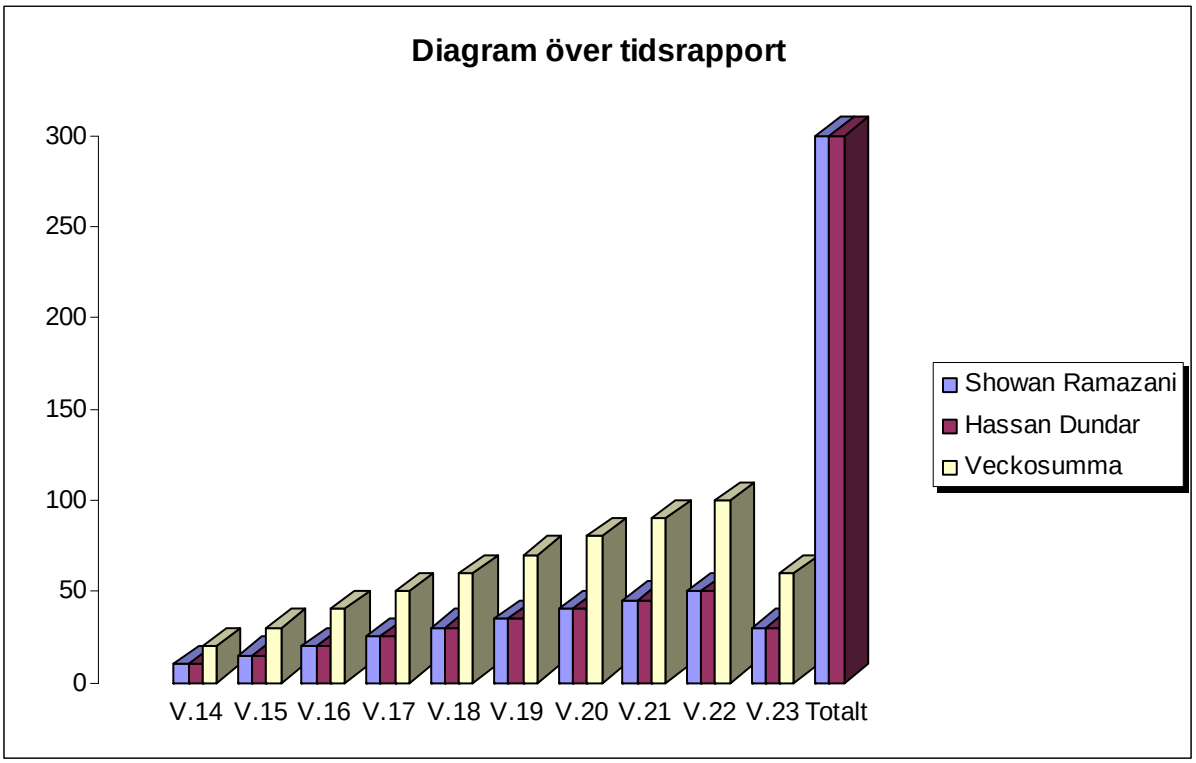
Hashat lösenord

För att skydda information såsom lösenord används ofta en "Hash", dvs en metod som förvränger lösenordet och dessutom genererar unika varianter för olika kombinationer. Vanliga algoritmer för detta är bland andra SHA1 och MD5. Som systemadministratör går det aldrig att se användares lösenord, och även att användarna aldrig kan få tillbaka och se vilket lösenord hon/han har. Istället används då en mekanism där ett nytt lösenord skickas till användaren, varvid denna kan gå in och ändra i efterhand.

Bilaga 2, Tidsrapport

Tidsrapport

Medlemmar	V.14	V.15	V.16	V.17	V.18	V.19	V.20	V.21	V.22	V.23	Totalt
Showan Ramazani	10	15	20	25	30	35	40	40	40	30	300
Hassan Dundar	10	15	20	25	30	35	40	40	40	30	300
Veckosumma	20	30	40	50	60	70	80	90	100	60	



Bilaga 3 - Kravanalys

Kravanalys

Följande krav på funktioner fick vi fram efter diskussion med arbetsgivaren:

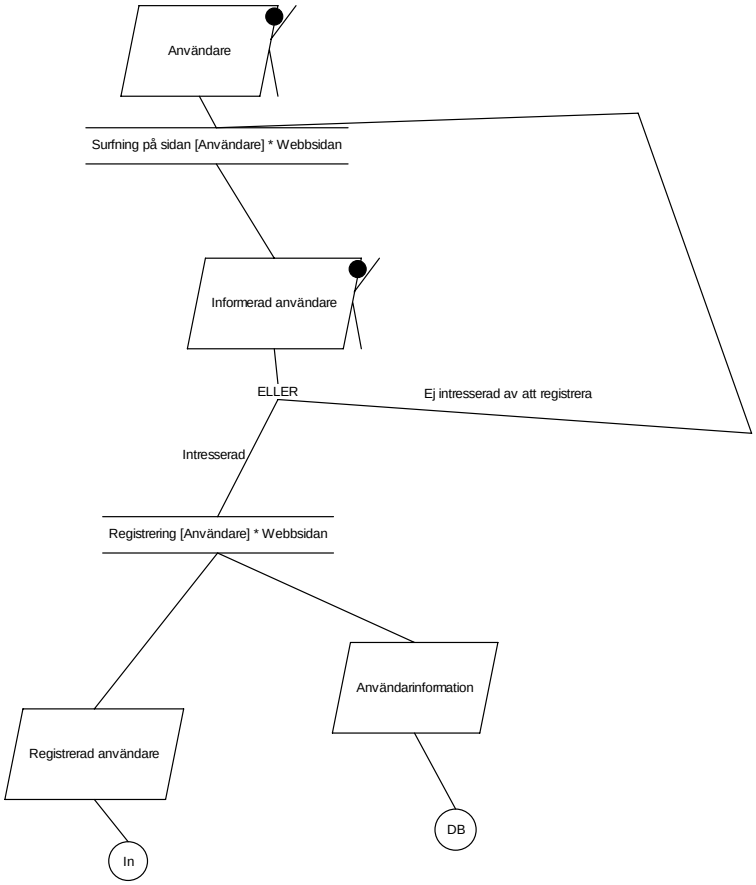
- Funktion för Registrering
- Funktion för Inloggning
- Funktion för Editering av registreringsuppgifter
- Funktion för att spara filer till server
- Funktion för att lista upp all användarens sparade filer
- Funktion för att ladda ner sparade filerna

Bilaga 4- Handlingsgrafer

HANDLINGSGRAF

Serie			**Dokument-Id**
Inloggning			reg
Utfärdare	**Datum**	**Version**	**Sida**
Hassan, Showan	2006-05-22	1	1 (1)

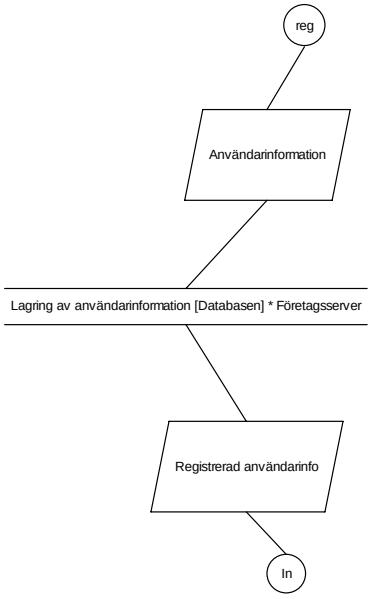
Dokumentnamn: Registrering



HANDLINGSGRAF

Serie			**Dokument-Id**
Inloggning			DB
Utfärdare	**Datum**	**Version**	**Sida**
Hassan, Showan	2006-05-22	1	1 (1)

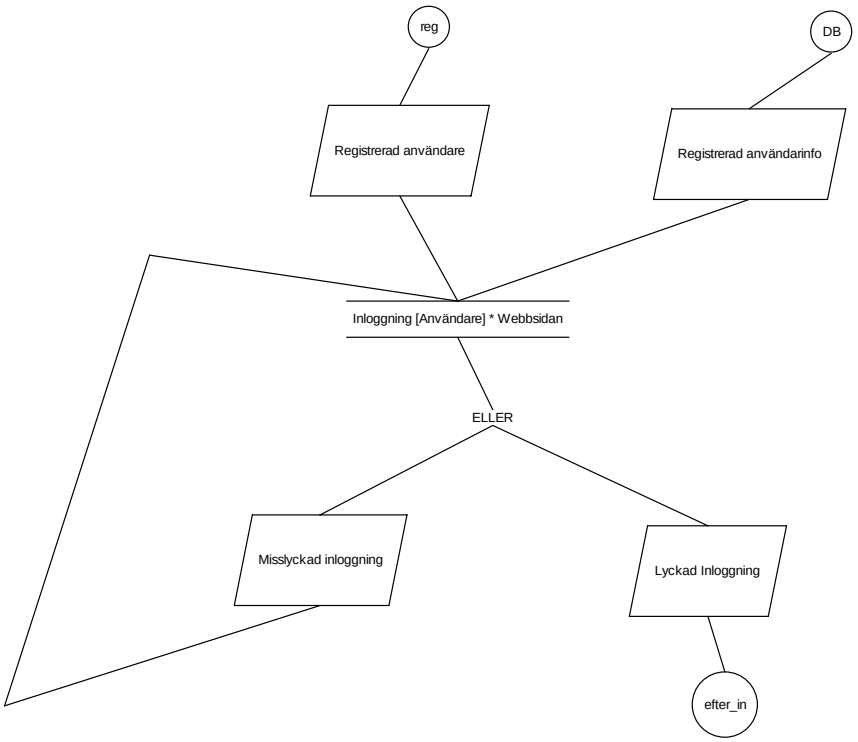
Dokumentnamn: Databas



HANDLINGSGRAF

Serie			**Dokument-Id**
Inloggning			In
Utfärdare	**Datum**	**Version**	**Sida**
Hassan, Showan	2006-05-22	1	1 (1)

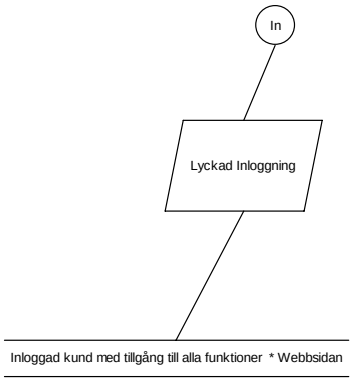
Dokumentnamn: Inloggning



HANDLINGSGRAF

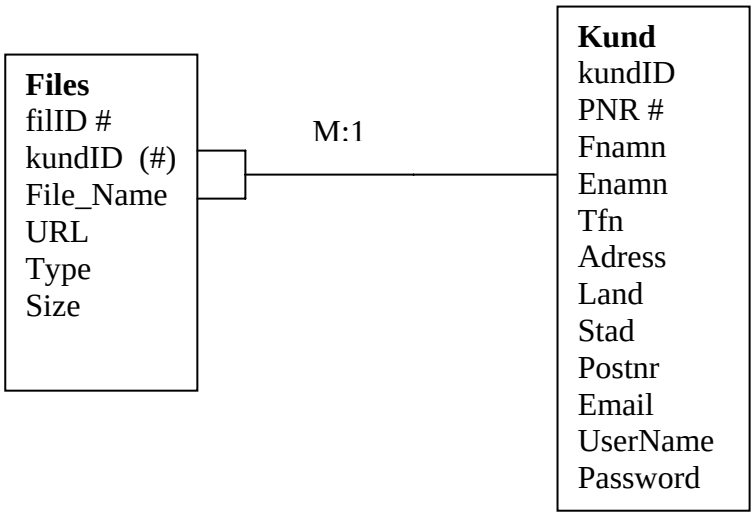
Serie			**Dokument-Id**
Inloggning			efter_in
Utfärdare	**Datum**	**Version**	**Sida**
Hassan, Showan	2006-05-22	1	1 (1)

Dokumentnamn: efter_inloggning



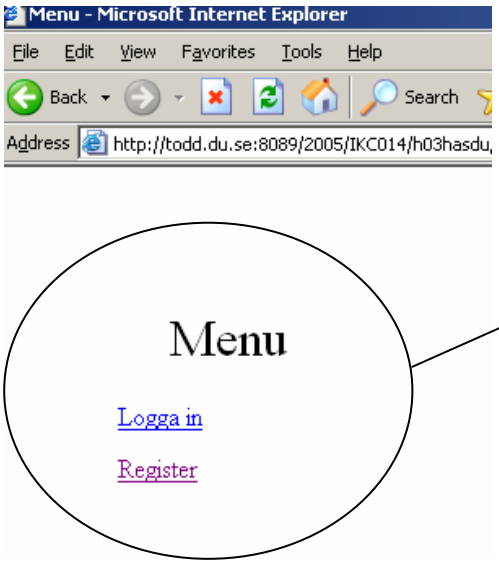
Bilaga 5 – Databasmodell

Datamodell


Files
filID #
kundID (#)
File_Name
URL
Type
Size
M:1
Kund
kundID
PNR #
Fnamn
Enamn
Tfn
Adress
Land
Stad
Postnr
Email
UserName
Password

Bilaga 6- Skärmdumpar

Meny



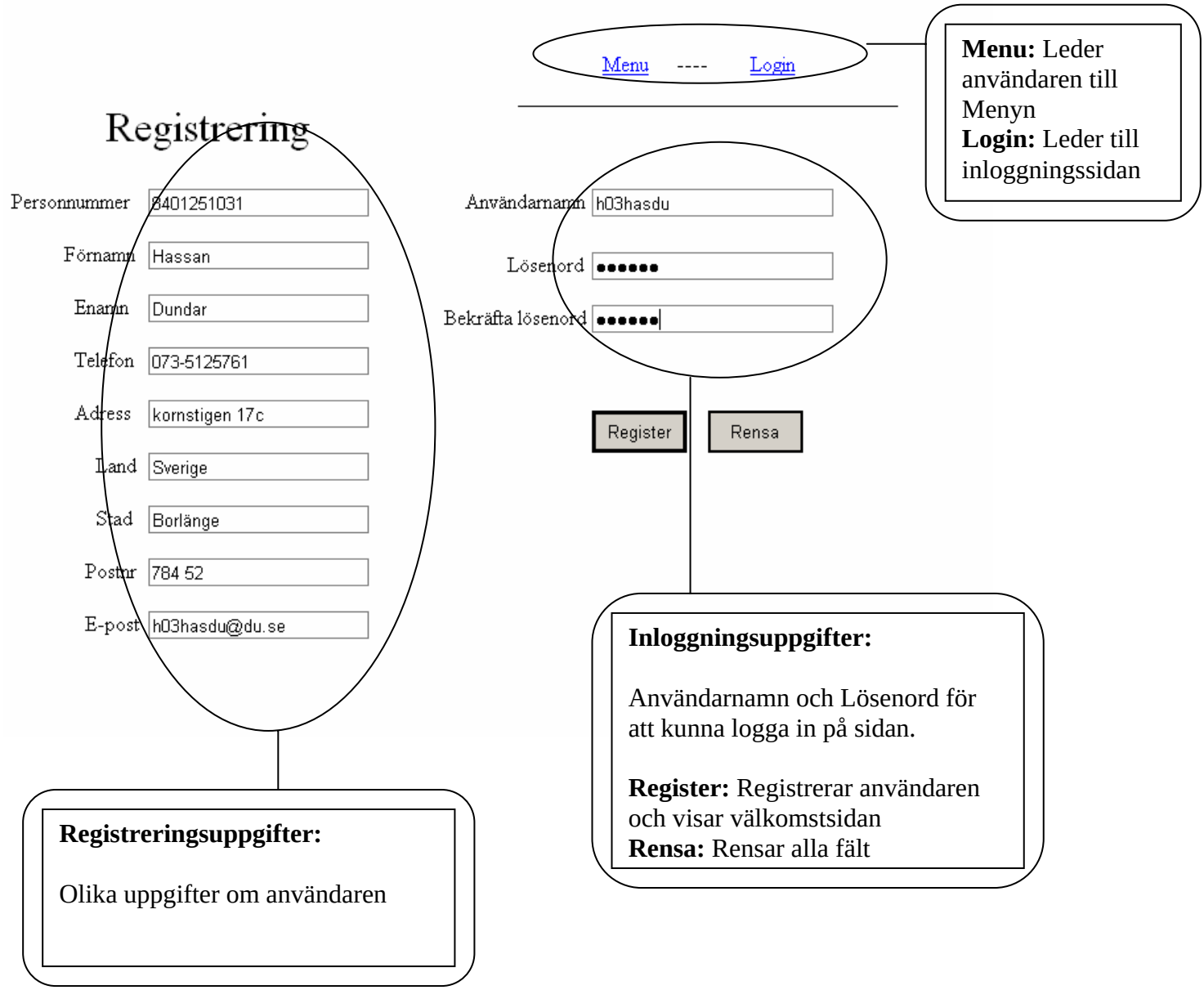
Meny före inloggning:

Logga in: Tar besökaren till login sidan
Register: Leder till registrationssidan där nya användare kan registrera sig.

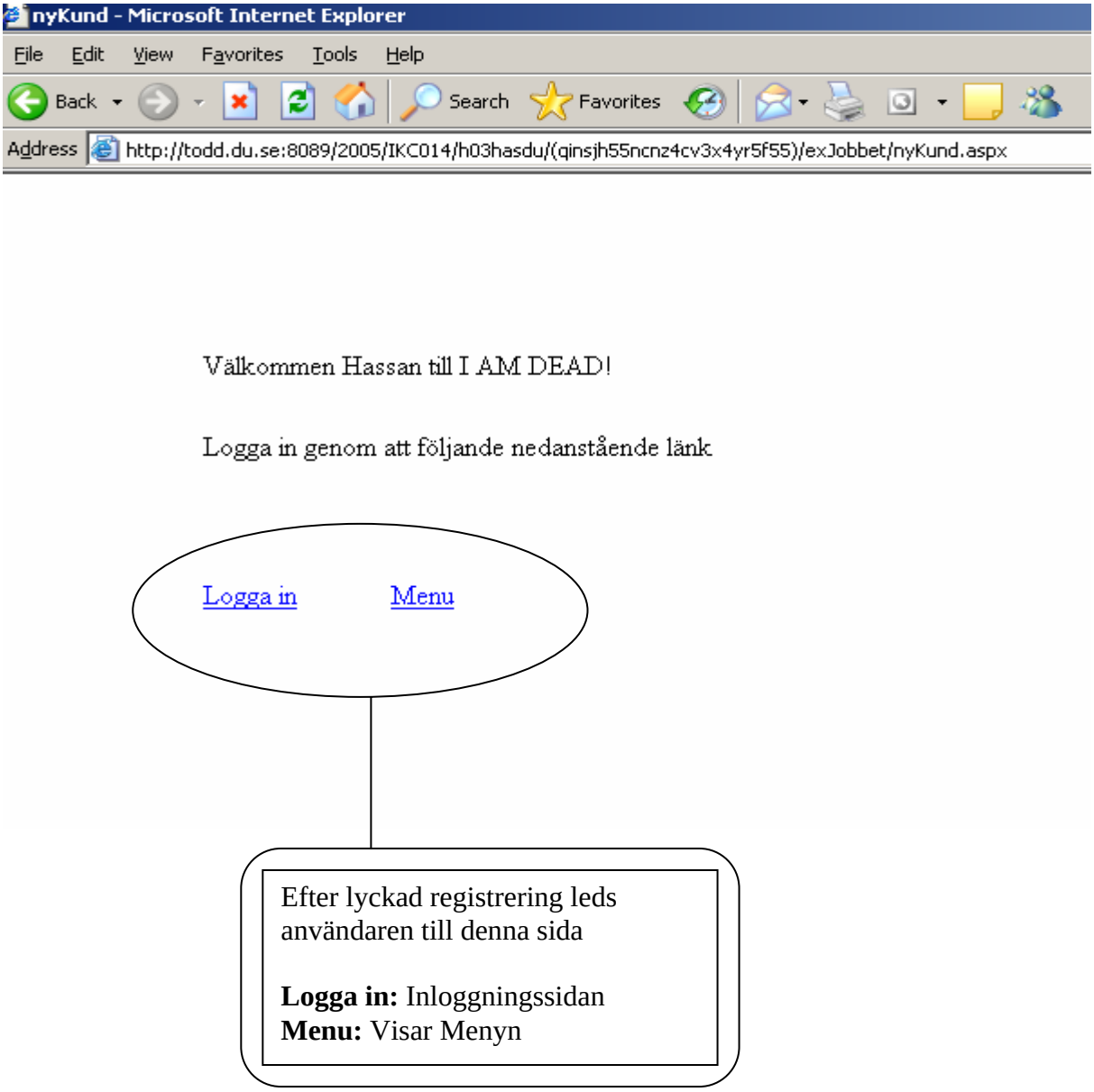
Meny Efter inloggning:

Inställningar: Visar användarens information och möjlighet att ändra dem
Ladda upp filer: Sidan för att ladda upp egna filer till servern
Mina Filer: Visar användarens sparade filer.
Logga ut: Loggar ut från sidan

Registrering

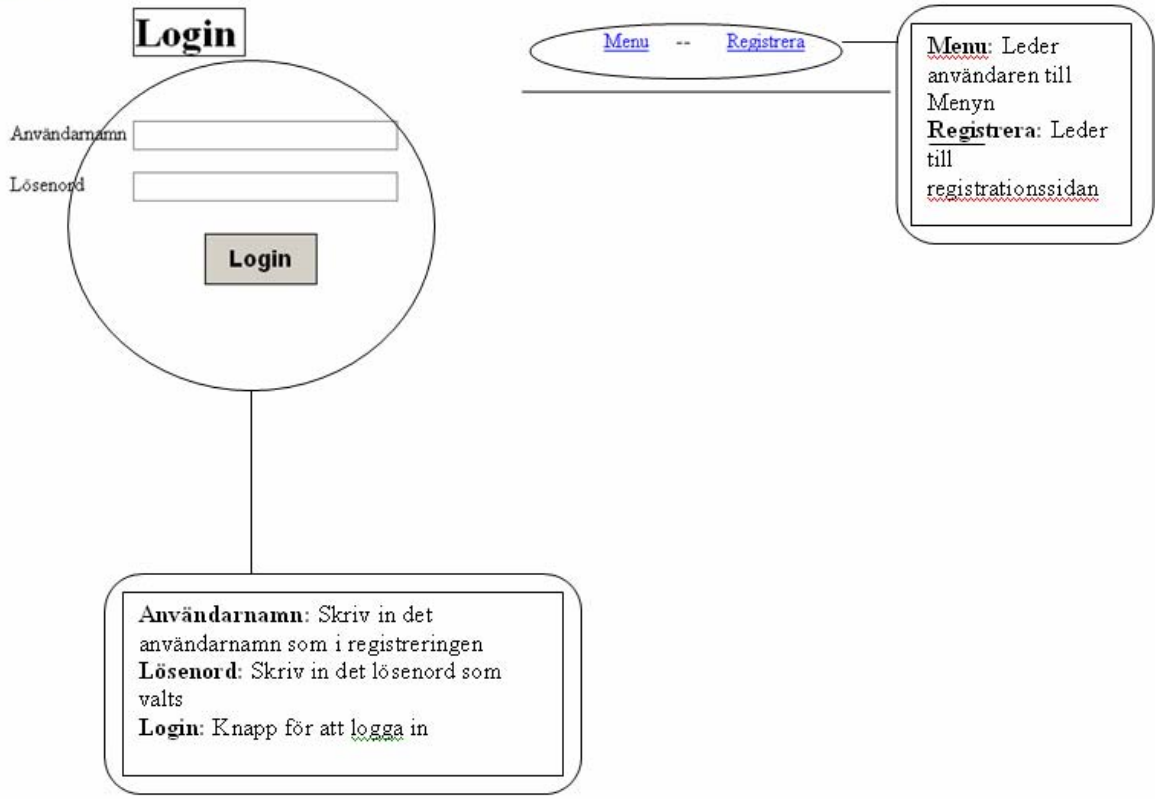


Välkomstsida

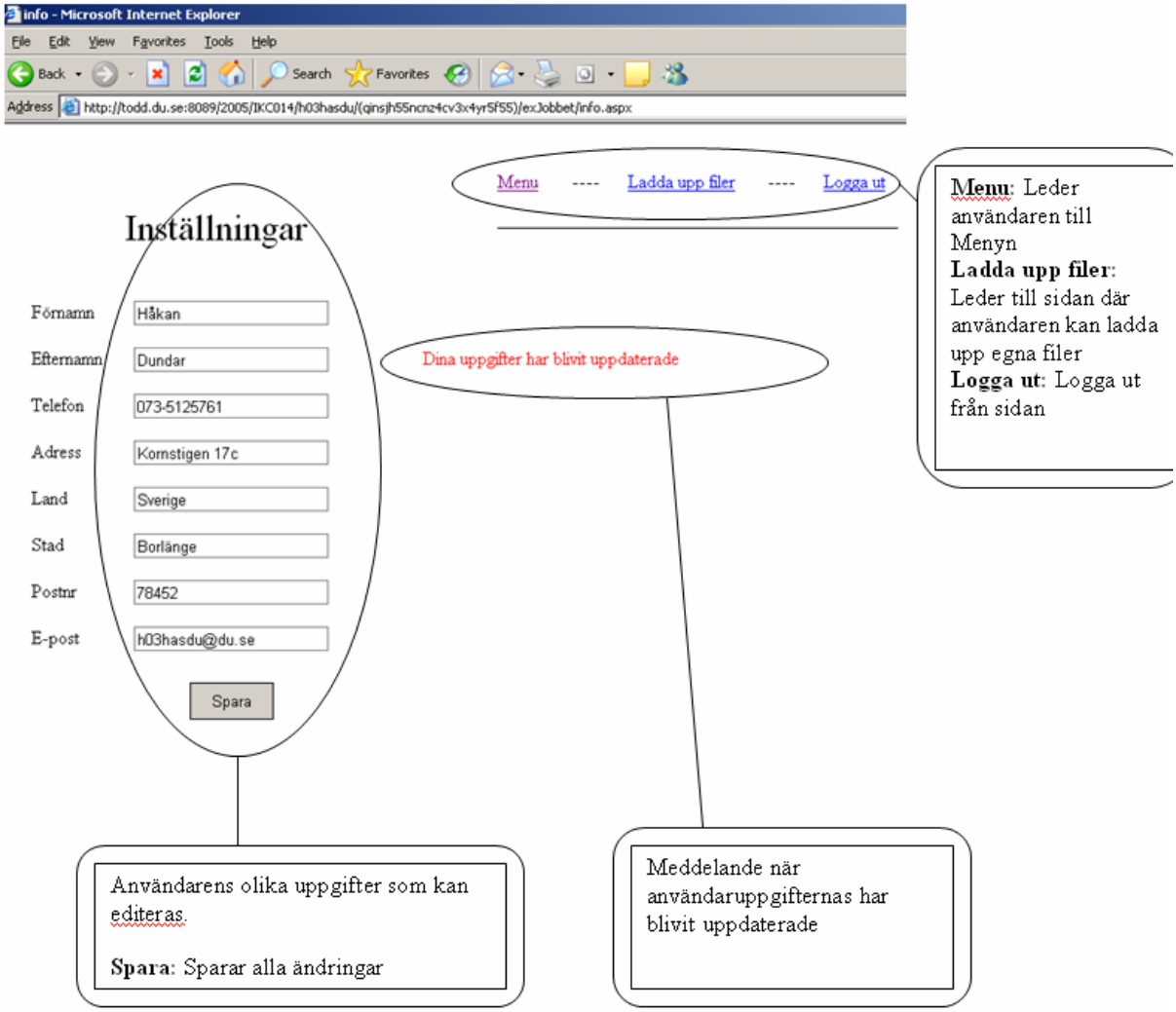


Login

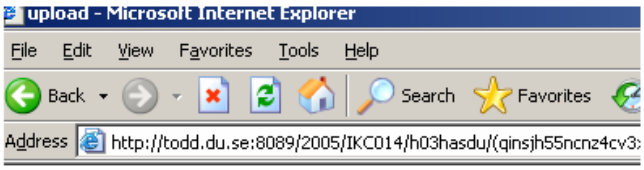

login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://todd.du.se:8089/2005/IKC014/h03hasdu/(ihxuvv3vaoqoemja0232e045)/exJobbet/loginMod.aspx


Login
Användarnamn
Lösenord
Login
Menu -- Registrera
Menu: Leder användaren till Menyn
Registrera: Leder till registrationssidan
Användarnamn: Skriv in det användarnamn som i registreringen
Lösenord: Skriv in det lösenord som valts
Login: Knapp för att logga in

Inställningar



Ladda upp filer:



Ladda upp filer

Välj den fill du vill ladda upp:

\\student\h03hasdu\Windesk\CV.doc Browse...

Mitt CV Namnge filen

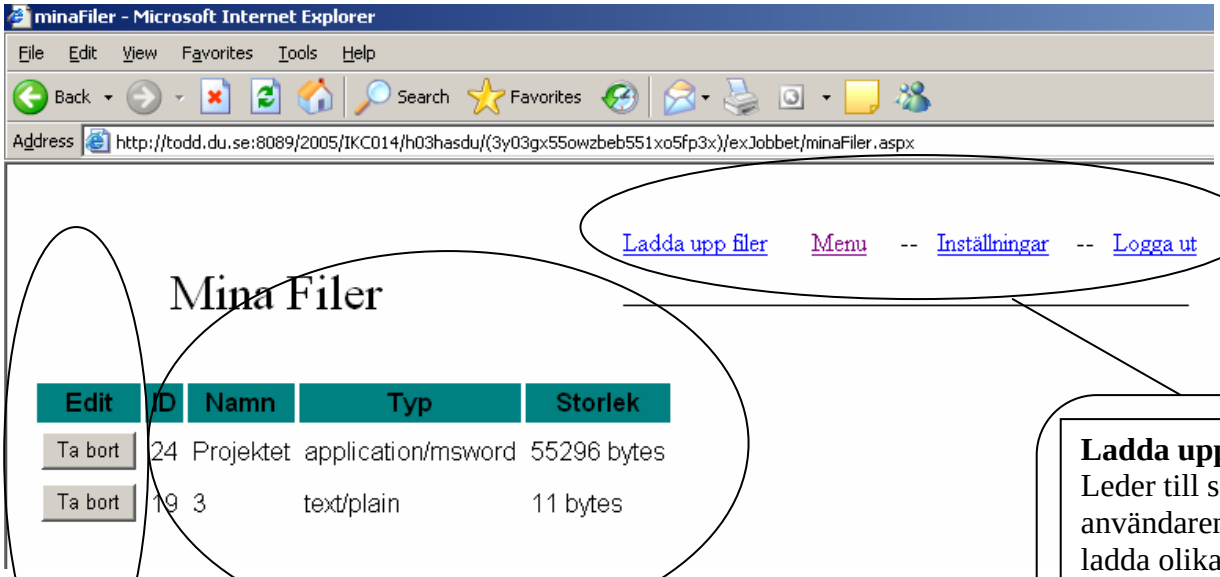
Upload

Ruta för sökvägen till filen som skall laddas upp.

Browse: Bläddra runt på datorn för att hitta filer att ladda upp
Namnge Filen: Namnet du vill ha på filen som skall sparas.
Upload: Knapp för att spara filen och namnet på filen till databasen.

FileName: \\student\h03hasdu\Windesk\CV.doc
File Content: application/msword
File size: 26624
Filen har nu sparats på servern i detta namn: CV.doc

File Name: Namnet på ursprungsfilen och sökvägen till den.
File Content: Typ av fil som har laddats up
File size: Storlek på filen i bytes.



Ta bort: Tryck på denna knapp för att ta bort filen

Dessa fält innehåller information om filen:

ID: Identifikationsnummret för filen
Typ: Vilken typ av fil det är
Storlek: Hur stor filen är (i bytes)

Ladda upp filer: Leder till sidan där användaren kan ladda olika filer
Menu: Leder användaren till Menyn
Inställningar: Leder användaren till sidan där användaren kan ändra sina uppgifter
Logga ut: Loggar ut användaren