

Webbtjänst för projektledare

Ett stöd för ekonomiska uppskattningar

ZANA ALI

EXAMENSARBETE

Webbtjänst för projektledare Ett stöd för ekonomiska uppskattningar

ZANA ALI

Sammanfattning

Denna rapport beskriver utvecklingen av ett system som tillhandahåller tjänster för projektledare som jobbar med EU. Systemet består av en .Net webbtjänst och två klienter. Den ena klienten är en webbapplikation som utnyttjar tjänster från webbtjänsten och den andra är en Windows applikation som jobbar gentemot en databas genom webbtjänsten för lagring av projekt information. Systems applikationer körs på datorer som finns i ett lokalt nätverk som skyddar webbtjänsten mot Internetattacker med brandvägg.

Webbtjänsten svarar på auktoriserade klienters anrop genom att klienten skickar sin identitet (användarenamn och lösenord) till webbtjänsten. Klient identiteten skickas som säkerhet elementer i ett Soap meddelande vilket tas emot och tolkas av webbtjänsten. Den metod som jag valt för att uppnå en säker webbtjänst är WSE 1.0.

Avslutningsvis har jag kommit fram till att webbtjänsten har fyllt de utveckligns krav som ställs på systemet. Vidare har jag åstadkommit den säkerhet som krävs för att klienterna skall kunna skicka sitt användarenamn och sitt lösenord som säkerhet elementer i Soap meddelandet.

Webbtjänst för projektledare

Web services for project leader The support of economic estimations ZANA ALI

Summary

This report describes the development of a system which provides services to the project leader, working with EU projects. The system consist of a .Net Web Services and two clients. The first one is a Web application which makes use of services from Web Services. Another client is a Windows application, which work with database through Web Services to store project information. Systems applications executes on the computers, which works in a local network there Web Services are protected against Internet attacks by firewall.

Web Services answers to authorized client's requests by sending identity (username and password) to web Services. Client identity is send as security element in head Soap message. WES version 1.0, is the method which I chose to reach a safe Web Service.

The result shows that the Web Service has reached the requirements, demanded by the system, and clients sends username and password as security element in the Soap message.

Publisher:	University of Trollhättan/Uddevalla, Department of Technology, Mathematics and Computer Science, Box 957, S-461 29 Trollhättan, SWEDEN Phone: + 46 520 47 50 00 Fax: + 46 520 47 50 99 Web: www.htu.se		
Examiner:	Lektor Hektor Sektor		
Advisor:	Sven Svensson, Företaget AB		
Subject:	Electrical Engineering	Language:	Swedish
Level:	Advanced	Credits:	10 Swedish, 15 ECTS credits
Number:	2004:E000	Date:	January 1, 2004
Keywords	5 – 10 words, separated by commas, describing the contents of the report		

Webbtjänst för projektledare

Innehållsförteckning

Sammanfattning.....	i
Summary.....	ii
Nomenklatur.....	3
1 Inledning.....	4
1.1 Bakgrund.....	4
1.2 Syfte och mål.....	5
1.3 Problemformulering.....	6
1.3 Metod.....	6
1.5 Avgränsningar.....	7
2 Introduktion till webbtjänst.....	8
2.1 XML.....	9
2.2 Soap.....	9
2.3 WSDL.....	10
2.4 UDDI.....	10
3 Kravspecifikation.....	11
4 Systemdesign för både Webbtjänsten och klienter.....	13
4.1 Tanken bakom designen.....	13
4.2 Intressenter och händelser.....	14
4.3 Systemarkitektur.....	15
4.3.1 klienter.....	16
4.3.2 webbtjänsten.....	16
4.3.3 Databas design.....	16
5 Implementering av webbtjänsten och klient.....	18
5.1 Klassdiagram.....	18
5.2 Webbtjänstapplikationer.....	19
5.2.1 Klassen Service1.asmx.cs.....	19
5.2.2 Klassen Databas.cs.....	20
5.2.3 Klassen PasswordProvider.cs.....	20
5.2.4 Detaljerad klassdiagram över metoder som ingår i webbtjänsten.....	21
5.3 klientapplikationer.....	22
5.3.1 Klassen Form1.cs.....	23
5.3.2 klassen WebForm1.aspx.cs.....	23
5.3.3 Detaljerad klassdiagram över metoder som ingår i klass klienten.....	24
6 Webbtjänsters säkerhet över Internet.....	26
6.1 krav på webbtjänstssäkerhet över Internet.....	26
6.2 Nätverksskärhet teknik för webbtjänster.....	27
6.2.1 IPsec och VPN (Virtual Private Networks).....	27
6.2.2 SSL/TLS (Secure Socket Layer and Transport Layer Security).....	28
6.3 Autentisering och auktorisering teknik för webbtjänster.....	29
6.3.1 Kerberos.....	29
6.3.2 Windows autentisering.....	29
6.3.3 HTTP bas säkerhet.....	30
6.4 Säkerhetsillämpning för webbtjänst på applikationsnivå.....	30
6.4.1 Introduktion till webbtjänstssäkerhet.....	30
6.4.2 Webbtjänstssäkerhets princip.....	32
6.4.3 Säkerhetsdesign för webbtjänsten i ett lokalt nät.....	32
6.4.4 Implementering av webbtjänstssäkerhet.....	33

Webbtjänst för projektledare

7	Slutsats	35
7.1	<i>Verifiering av kravspecifikation</i>	35
7.2	<i>Verifiering av mål</i>	36
8	Källförteckning	37

Bilagor

- A Webbtjänsts källkod
- B klienters källkod

Webbtjänst för projektledare

Nomenklatur

<i>DHCP</i>	D ynamic H ost C onfiguration P rotocol kommunikationsprotokoll för Internet som tilldelar klienten en tillfällig IP-adress varje gång denna kopplar upp sig.
<i>HTML</i>	H ypertext M arkup L anguage standard, baserad på S GML, för strukturering av information på bl.a. webbsidor och i e-post.
<i>SQL</i>	S tructured Q uery L anguage standardiserat språk för frågeställning, hantering och manipulation av data i relationsdatabaser.
<i>HTTP</i>	H ypertext T ransfer P rotocol kommunikationsprotokoll för överförngav bl.a. H TML- dokument på Internet.
<i>LAN</i>	L ocal A rea N etwork Lokalt datornät, oftast begränsat till ett eller flera hus som tillhör samma organisation (t.ex. ett företag, en skola).
<i>ASP</i>	A ctive S erver P ages- A ppliaktion S ervice P rovider produkt för dynamisk framställning av webbsidor- programuthyrningsföretag eller annat företag som erbjuder tjänster som innebär att kunden hyr tillgången till program och drift via t.ex. Internet.
<i>IP</i>	I nternet P rotocol en uppsättning regler för kommunikation mellan datorer på Internet.
<i>URL</i>	U niform R esource L ocator, är en webbadress

Webbtjänst för projektledare

1 Inledning

Arbetet handlar om att utveckla en webbtjänst som hjälper projektledare och andra som arbetar med EU-projekt att bl.a. göra ekonomiska uppskattningar. Systemet skall presentera flera funktioner genom ett webbgränssnitt. Systemets komponenter skall innehålla ett antal grundläggande funktioner som hjälper en projektledare att göra uppskattningar över hur mycket ett projekt kostar och vilken samhällspåverkan det har.

Webbtjänsten kopplas till en databas för lagring av information. Information matas in från en administratör/klient till webbtjänsten. Andra klienter till webbtjänsten har bara rätt att läsa informationen.

Arbetet består av två delar, utveckling av en prototyp till en webbtjänst samt en teoretisk del som behandlar säkerheten i webbtjänster. I den första delen utvecklas en webbtjänst som fungerar i ett lokalnät utifrån uppdragsgivarens kravspecifikation. I teoridelen diskuteras säkerhetsproblematiken med webbtjänster och olika tekniska lösningar. Rapporten diskuterar också de design val som ligger bakom prototypen.

1.1 Bakgrund

Idag strävar många organisationer efter att ha ett IT-stöd för sina medarbetare för att underlätta deras arbetsuppgifter som att göra ekonomiska uppskattningar och beslutfattning. IT-stödet kan nås genom att bygga ett system som till exempel systemet är tillgängligt på ett lokalt nät eller över Internet för multianvändare. Detta system skall klara av in/utmatning och lagring av information. Det är således effektivt att bygga systemet med en standard teknik istället för att varje organisation utvecklar en egen teknik för att åstadkomma både informationsutbyte och säkerhets policy.

Informationsutbyte mellan systemet och systemklienterna skall ske med säkerhet så att informationen inte modifieras eller avlyssnas av obehöriga. För att systemadministratörer skall ha tillgång till systemets resurser krävs det autentisering.

En utvecklad standard teknik för informationsutbyte mellan applikationer i ett lokalt nät eller över Internet är webbtjänster (Web Services).

Webbtjänst för projektledare

Webbtjänster är en samling av ett antal standard komponenter som XML, Soap, WSDL och UDDI vilka används för informationsutbyte mellan applikationerna. Dessa applikationer kan skrivas på olika programspråk som java eller C# osv. Plattformar som webbtjänster körs på kan också vara olika, Webbtjänster är med andra ord plattformsoberoende[4].

Webbtjänster med standard komponenter räcker inte för uppnå informationsutbyte med säkerhet mellan applikationerna eftersom webbtjänsternas WSDL dokument är baserad på Soap meddelande och XML-schema och vilket kan läsas av obehöriga. En lösning för att förebygga problemet beträffande applikations och nätverk säkerheten är att använda sig av de olika tekniska mekanismer som webbtjänstsäkerhet (Web Service Security) och SSL/TLS

Webbtjänstsäkerhet är ett förslag till standardformat som håller säkerhetsrelaterad information i Soap meddelande. Den är en specifikation som definierar hur signatur och kryptering algoritmer bifogas till *Head* Soap meddelande och definierar säkerhetssymboler som Kerberos biljett eller andra certifikat och nyklar. Tekniken behandlar flera säkerhetsmekanismer som SSL/TLS (Secure Socket Layer and Transport Layer Security), XML signatur och XML kryptering[4].

SSL/TLS erbjuder säkerhet på transportnivå och möjliggör point-to-point säkerhetsessioner. Den erbjuder dessutom verifiering (authentication), data integritet och konfidentiella.

XML-signatur specificerar en XML syntax för presentering och verifiering av signatur för webbtjänster.

XML-kryptering utvecklar en process för kryptering och dekryptering. XML-Kryptering beskriver information som krypteras av avsändare till systemet, som i sedan dekrypteras av mottagare.

1.2 Syfte och mål

Avsikten med arbetet är att utveckla en webbtjänst som i första hand hjälper projektledare. Systemet skall fungera dels som en självständig enhet och dels som integrerad enhet med en databas. Databasen lagrar all information som ett projekt behöver för att en projektledare skall kunna göra sina ekonomiska uppskattningar för hur mycket ett projekt kostar och vilken samhällspåverkan projektet ger.

Den utvecklade webbtjänsten hanterar många användare i en organisation i ett lokalt nät. Klienterna till webbtjänsten kan antingen vara projektadministratörer som har rätt till att både läsa och skriva information eller andra klienter som bara har rätt till att läsa information.

Webbtjänst för projektledare

Syftet är vidare att undersöka säkerhetstekniker för hur webbtjänsten kan vara tillgängligt över Internet. Här behandlas de teoriska problematikerna och lösningar med webbtjänstsäkerhet för informationsutbyte över Internet. Slutligen är syftet att tillämpa en metod för att uppnå säkerhet i det lokala nätet.

1.3 Problemformulering

Frågorna kan formuleras på följande sätt:

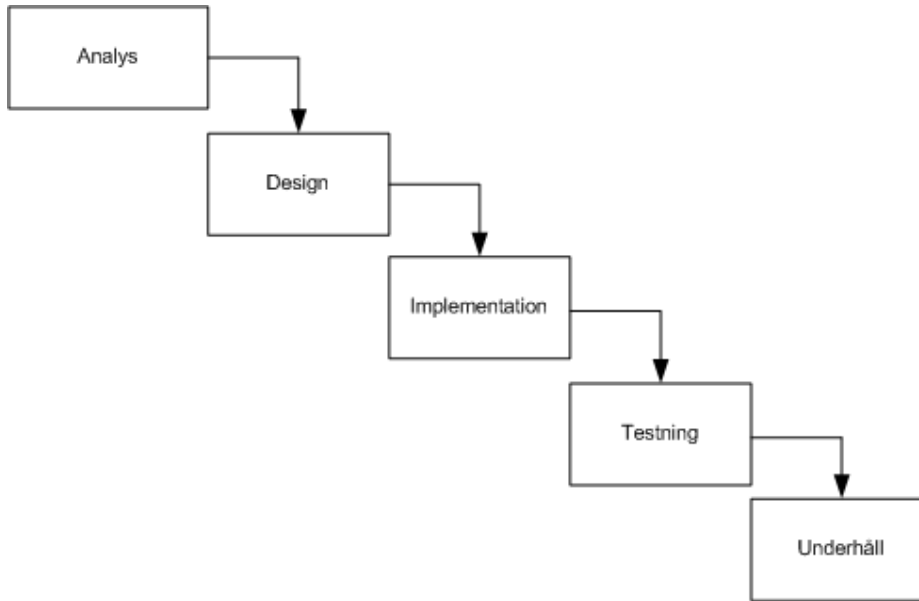
1. Hur kan en projektledare använda sig av en webbtjänst för att göra ekonomiska uppskattningar för hur mycket ett projekt kostar och vilken samhällspåverkan projektet ger?
2. Hur skall webbtjänsten teoretiskt leverera information över Internet på ett säkert sätt?
3. Hur skall webbtjänsten hantera autentisering i det lokala nätet.

1.3 Metod

En metod som används för att utveckla webbtjänsten är vattenfallsmodellen. Modellen består av flera aktiviteter som analys, design, implementering, testing och underhållning se figur 1.3. I rapporten används endast vissa av dessa aktiviteter för att beskriva och analysera utvecklingen av webbtjänsten. Den praktiska delen i projektet börjar med analysfasen där behandlar jag dem krav på systemet som önskas från uppdragsgivaren. Därefter kommer jag till designfasen där design av webbtjänsten och andra komponenter beskrivs. Slutligen i implementeringsfasen visas hur systemet möter kravspecifikationen.

Systemet upprättas i .Net miljö och programmering språk C#. Webbtjänsten körs under Windows XP med .Net ramverk 1.0. och IIS 5. Databas verktyget som används här är MYSQL databas.

Webbtjänst för projektledare



Figur 1.3 : Vattenfallsmodell aktiviteter

För att åstadkomma information som används i rapporten studerats ett antal böcker och artiklar. Föreläsningar från kurser också används för implementering av källkoder.

1.5 Avgränsningar

- Vad som kommer att upprättas i systemet är utvecklat i .Net ramverksmiljö, ASP.net och programmeringsspråket C#.
- Databasen som används är MYSQL.
- Webbtjänsten fungerar bara i en lokal nät.
- Webbtjänsten levererar grundläggande information som är projekt kostnader och intäkter, vidare klienterna kalkylerar informationen med en räntesats.
- Problematik och lösningar om hur webbtjänsten skall fungera över Internet undersöks bara teoretiskt.
- Bara en praktisk implementering utförs för att bygga säkra webbtjänster i det lokala nätverket.

Webbtjänst för projektledare

2 Introduktion till webbtjänst

Webbtjänster är en samling av fyra komponenter XML, Soap, WSDL och UDDI. [5]. De har utvecklats på koncept av distribuerat process för att bygga komponenter vars metoder kan anropas över Internet eller i lokalnät. Metoderna är webbtjänstmetoder som utför en eller flera specifika uppgifter. Kommunikation mellan de anropade applikationerna och Webbmetoderna sker genom att skapa en Proxy med själva webbtjänsten [2].

Det finns fördelar och nackdelar med att använda webbtjänster som ett system för informationsutbyte:

- Webbtjänster använder HTTP över TCP port 80. Fördelen med detta är dels att porten alltid är öppet för webbserver och dels att webbtjänster använder porten som en tunnel och då är det lättare att komma förbi brandvägg.
- Ändring av komponenter har blivit viktig för en eventuell programvara som bygger på en samling av individuella komponenter utan att dessa har negativa effekter på varandra. Webbtjänster erbjuder därmed den flexibilitet att koppling mellan applikationerna och webbtjänsten sker fritt.
- Webbtjänsten kan själv byta tjänster med andra webbtjänster med hjälp av webbtjänst-routing. Den ligger som ett tillägg i Soap meddelande för att addera meddelandets sökväg (path). Med webbtjänst-routings skriver informationen regler och kontext där meddelande skickas från en nod till andra via en eller flera mellanhänder. Dock kan Webbtjänst-Routing inte erbjuda information som krävs av parter som är involverade med meddelandet, till exempel en mellanhänder till ett Soap meddelande behöver nödvändig information för att välja sökväg med villkor. Detta problem kan då lösas med webbtjänst-Referral [4].
- Att använda Soaps protokollspecifikation innebär att Soap hjälper applikationerna att förstå de standarder karven för förfrågningar och svar och hur de tolkas med WSDL dokument som växlas mellan applikationerna. WSDL är ett kontrakt som beskriver datatyper, metoder och parametrar mellan klienterna och webbtjänsten för informationsutbyte och tjänster. Att jobba med de standarderna innebär bara integritet på språk, data typer och transaktion nivå. Men det finns flera områden som måste adresseras till webbtjänster för att öka ytterliggare funktionalitet för att blir en standard teknik. Dessa områden behandlar frågor om säkerhet i Soap meddelande som skickas mellan applikationer och själva webbtjänsten.

Webbtjänst för projektledare

2.1 XML

XML (Extensible Markup Language) är en standard som beskriver data och är en *Extensible Markup Language*. XML gör webbtjänster betydelsefulla genom att applikationer kan kommunicera med varandra om de kan tolka XML oberoende av deras plattform eller programmeringsspråk. XML skiljer sig från HTML med att XML- utvecklare kan själv bygga struktur och definiera egna taggar för lagring av data , vilket kallas för XML dokument.

Ett XML dokument refereras till andra dokument för att kontrollera XML strukturen. De andra dokumenten är DTD (Document Type Definiton) eller XML schema[5].

2.2 Soap

Soap (Simple Object Access Protocol) är det mest vanliga standardprotokollet för kommunikation mellan webbtjänster. Soap utför dataöverföring mellan två system över ett nätverk d.v.s. att två system kan utbyta data med varandra. Klientapplikationer kan genom Soap meddelande kommunicera med webbtjänst. Medelanden som skickas till webbtjänsten bearbetas där. Webbtjänsten använder information som finns i Soap medellandet för att utföra sina funktioner. Webbtjänsten returnerar sedan resultatet via andra Soap meddelande till klienten om det erfordras i det skickade Soap meddelandet.

Soap är en XML baserad kommunikations protokoll som är ett uppsättning av standardiserade XML schema. Scheman definierar format för överföring av XML meddelande över nätverk. Dessa meddelanden inkluderar datatyper och strukturer som webbtjänsten/klienter kan tolka på rätt sätt.

Soap meddelande består av tre delar: *Envelope*, *Head* och *Body*. Envelop packar hela meddelandet d.v.s. *Head* och *Body* elementerna. I *Envelop* kan olika namespaces deklarerar för definition exempelvis av förekommande datatyper i dokumentet.

Head är en valfri del, om den förekommer då kan exempelvis information om säkerhet eller routing osv. bifogas

Body delen innehåller data som transporteras mellan två system. Den innehåller metodanrop och eventuellt parametrar eller retur data från webbtjänsten.

Soap har blivit vald som RPC-protokoll på grund av enkelheten, utbyggbarheten och Interoperabiliteten och den rekommenderas för uppbygganden av webbtjänster[5].

Webbtjänst för projektledare

2.3 WSDL

WSDL (WSDL (web Service Description Language) är ett XML baserat språk och beskriver information som klientapplikationer behöver för att anropa webbtjänster.

WSDL är ett dokument som Webbtjänster använder för att beskriva sina tjänster, metoder eller funktioner för att tillhandahållas för klienter. I WSDL dokumentet definieras meddelande som webbtjänsten mottagit respektive skickat och datatyper som applikationerna mottagit/sänt till webbtjänsten. WSDL dokumentet erbjuder också en specifik teknik där den talar om för klienterna hur de skall kommunicera med webbtjänster över HTTP eller andra protokoll[5].

2.4 UDDI

UDDI är ett register för företag och utvecklare för att publicera, registrera och lokalisera sina webbtjänster över Internet. UDDI är ett XML baserat format och i den kan företag och utvecklare beskriva tjänster och affärsprocesser processer. Detta register kan vara tillgängligt för allmänna eller privata klienter [5].

Webbtjänst för projektledare

3 Kravspecifikation

I detta kapitel beskrivs kravspecifikationen som systemet skall grundas på i ett lokalnät. Kravspecifikationerna har kommit en del från uppdragsgivaren som har förslaget för utveckling det systemet och en del har kommit med överens med examinator.

- **Webbtjänsten skall skapas i ett lokalnät:** Webbtjänsten skall byggas i ett lokalnät och det skall skyddas från Internetattacker med en brandvägg. Klienterna skall också byggas i samma lokalnät. Det menas att webbtjänsten har en central roll där frågor från klients applikationer till databasen går via webbtjänsten.
- **Databasen skall lagra och returnera data från och till klienter genom webbtjänsten oberoende av deras plats:** Databasen skall skapas i en valfri dator. Datorn kan placeras i lokalnätet eller i Internet. Databasen skall endast svara bara på frågorna som kommer från webbtjänsten.
- Webbtjänsten är ett lager mellan klienterna och databasen: systemet skall fungera som tre lager arkitektur.
- **Webbtjänsten tillhandhåller tjänster till behöriga Klienter genom autentisering:** Klienterna d.v.s. projektsamordnare och projektledare skall logga in med ett användarenamn och ett lösenord till webbtjänsten för att kunna hantera projektets information. Klienternas identiteter för inloggningen till systemet skall skickas krypterat mellan säkerhet elementerna i Soap meddelandet.
- **Spara data om projekt:** Projektsamordnare skall lägga in information i en databas. Informationen är projektkostnader i form av löner till projektledare och bidrag till projektanställda. Webbtjänsten hanterar information och skickar vidare dem till databasen. Webbtjänsten returnerar kostnader/intäkter som lagars i databasen för vidare kalkylering som utförs av klienterna. Intäkter från ett projekt är skillnaden mellan samhällsbidrag till de anställda i projektgång och efter projektslut. Kostnaderna för ett projekt är lön till projektledning och anställda och övriga kostnader i form av hyror och material. Kalkylering baseras bara på kostnader och intäkter. I resultatet beräknas den period som behövs för att projektet skall vara tillfredsställande att ge samhällsvinst. I kalkyleringen beräknas en räntesats på 4 % för att vinst perioden skall räknas.

Webbtjänst för projektledare

- **Skriva ut projektrapport:** Den är en viktig funktion som ger projektledare koll på vilka summor som kommer in i ett projekt och hur samhälle påverkas av ett projekt ger efter projektets slut att . Webbtjänsten ger data (projekt information) till klienterna som i sin tur kalkylerar de och presenterar resultatet till användare (projketledare).
- **Systemet skall vara tillgängligt för flera användare:** Genom att skapa webbapplikation som utnyttjar webbtjänsten skall flera användare kunna tillhandshålla tjänster från webbtjänsten. Webbapplikationen skall bara presentera projektrapport.
- **Databasen har stöd för att lagra information oberoende av datatyper:** Det är viktigt att databasen lagrar olika datatyper som strängar, datum, heltal och decimaltal för att siffror av projekt information kan vara till exempel decimaltal.
- **Parametrar som skickas mellan klienter och webbtjänsten skall vara enkla datatyper:** Parametrar som skickas mellan klienterna och webbtjänstsmetoder skall vara enkla datatyper för att inte belasta nätverket med många data. Servermaskinen som webbtjänsten kör på slipper då att göra kalkyleringar eftersom det är klienterna som gör dem. Webbtjänstens metoder kan själva skapa SQL frågor till databasen för hämtning eller lagring av information till och från databasen.
- **Webbtjänsten måste generera fel meddelande till klienter:** Det är viktigt att webbtjänsten skickar meddelande till anropande webbmetoder om det uppstår fel exempelvis att fel parametrar har skickats till en webbmetod.

Webbtjänst för projektledare

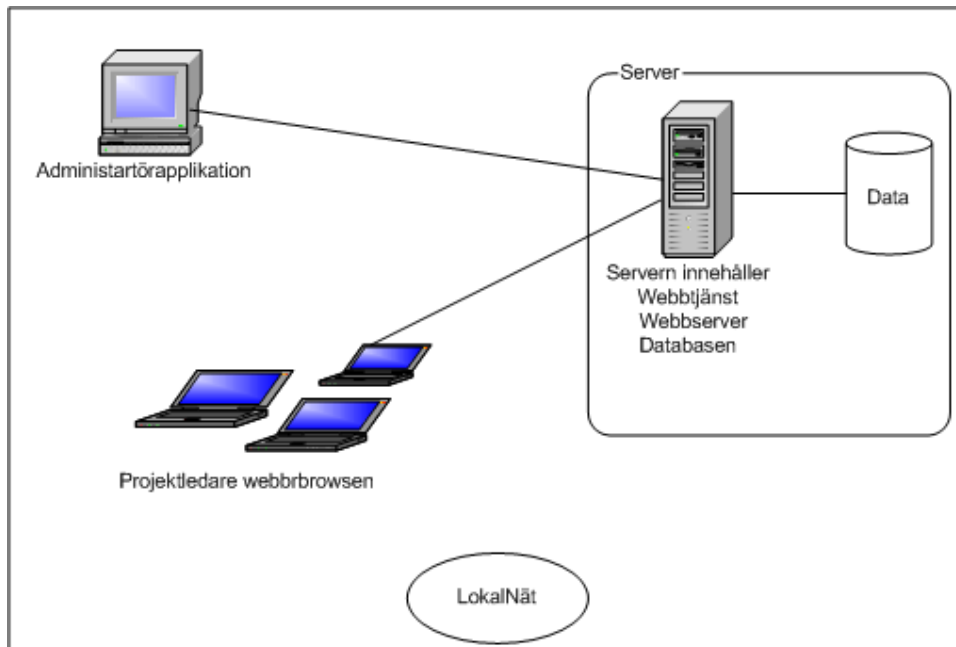
4 Systemdesign för både Webbtjänsten och klienter

I följande kapitel behandlas designsval för utveckling av webbtjänsten och andra subsystem för att bygga systemet som en helhet. Dessutom skall val av tekniska säkerhetsmekanismer för webbtjänsten i lokalnät behandlas i detta kapitel.

4.1 Tanken bakom designen

Systemet utvecklas i en tre lager arkitektur. Klientapplikationernas frågor/svar går via webbtjänsten till databasen och dessa utvecklas och placeras i en dator som går i ett lokalt nät. Klientapplikationerna utvecklas och placeras i annan dator som ingår i samma lokala nätverk.

Tjänsterna från webbtjänsten utnyttjas sedan av klienter vilka kommer att vara oberoende av webbtjänstens placering. figur 4.1.



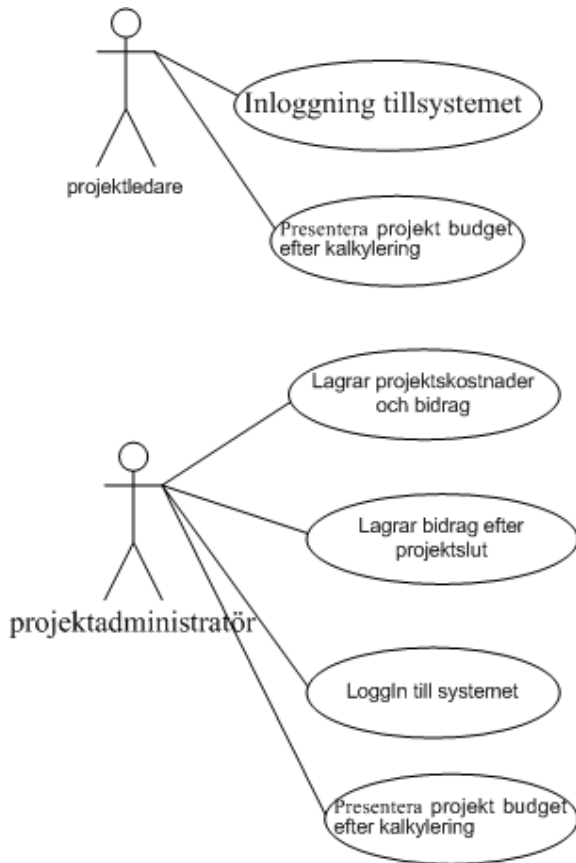
Figur 4.1: Tre lager arkitektur över systemet i ett lokalt nät

Webbtjänst för projektledare

4.2 Intressenter och händelser

Ett Use-Case diagram ger en översiktlig bild över systemet och dess intressenter (se figur 4.2). I bilden visas de inblandade parterna och de händelserna som sker i systemet. Här kan projektledarna göra förfrågningar om kostnader/intäkter i ett projekt. Förfrågningarna går till webbtjänsten som skickas vidare till en databas. Databasen innehåller lagrad information vilket projektadministratör skickat via händelsen "lagra ett projekts kostnader/bidrag" och händelsen "lagra bidrag efter ett projektslut".

Båda intressenterna kan göra inloggning till systemet via händelsen "inloggning till systemet" och presenterar projekts budget via händelsen "presenterar projekt budget efter kalkylering".

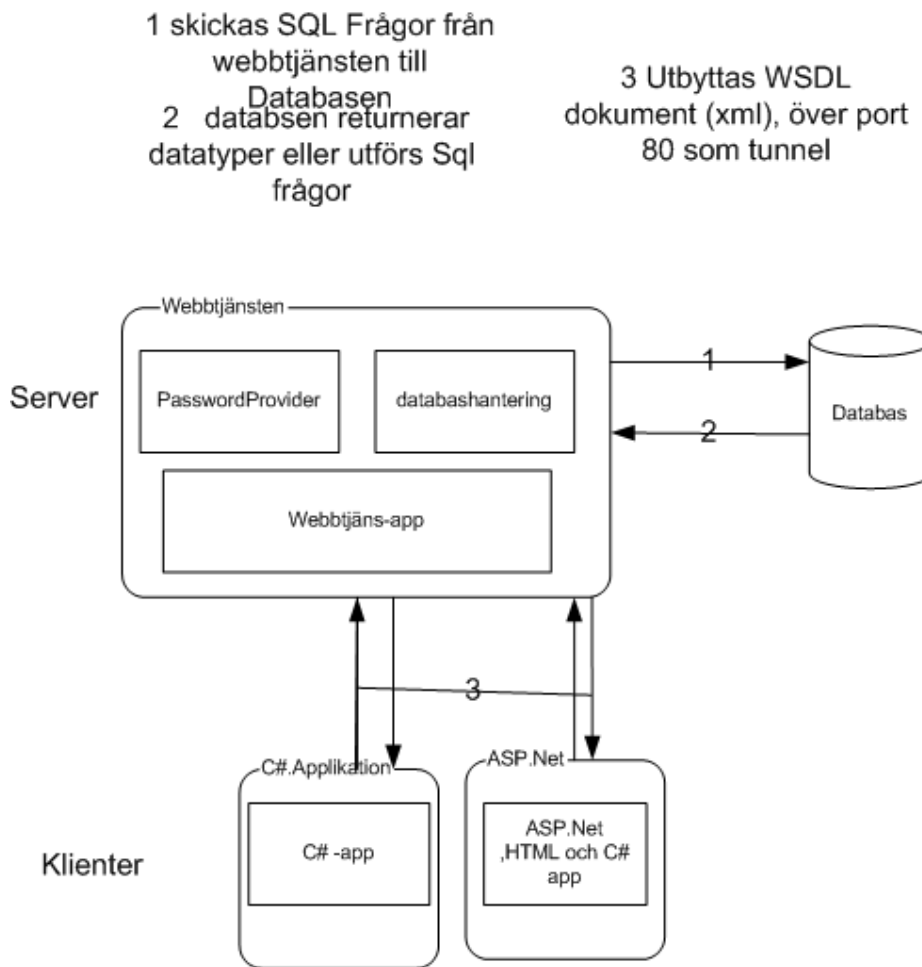


Figur 4.2: Use-Case diagram över systemet med de intressenter och händelserna.

Webbtjänst för projektledare

4.3 Systemarkitektur

Figur 4.3 visar en helbild över hur kommunikation mellan de olika delsystemen sker inom systemet samt kommunikationen mellan systemet och dess intressenter, och slutligen vilka protokoll som används.



Figur 4.3: Helbild av kommunikation mellan webbtjänsten och klienterna och kommunikation protokoll.

Webbtjänst för projektledare

4.3.1 Klienter

Klientapplikationerna som utnyttjar webbtjänsten är olika. Projektadministratörens applikation körs på en dator vilken ger möjlighet för projektadministratören att lagra information om ett projekt och vidare presenterar en rapport om projektbudgeten. De andra applikationerna är de som är webbapplikationer och utnyttjar webbtjänsten för att hämta data som har lagrats av projektadministratören för kalkylering och presentering av budget. Generellt använder klienterna Soap protokollet för kommunikation med webbtjänsten.

Projektadministratörens applikation skapas med språket C# i .Net miljö. Andra klienter skapas med ASP.Net som är webbserver applikation och är tillgängliga för flera användare.

4.3.2 Webbtjänsten

Webbtjänstens applikation har en central roll i systemet. Den har som uppgift att tillhandhålla tjänster för klienterna genom att returnera data om ett projekt till dem. Webbtjänsten har också metoder vilka projektadministratören kan utnyttja för att lagra data i form av projektkostnader och samhällsbidrag till projektanställda och metoder för att lagra summa av samhällsbidrag efter projektets slut. Kommunikationen mellan klienterna och webbtjänsten sker med Soap protokoll.

Webbtjänsten vänder sig till en databas för att hämta olika data beroende på vad som efterfrågas. Kommunikationen mellan webbtjänsten och databasen sker med SQL-frågor.

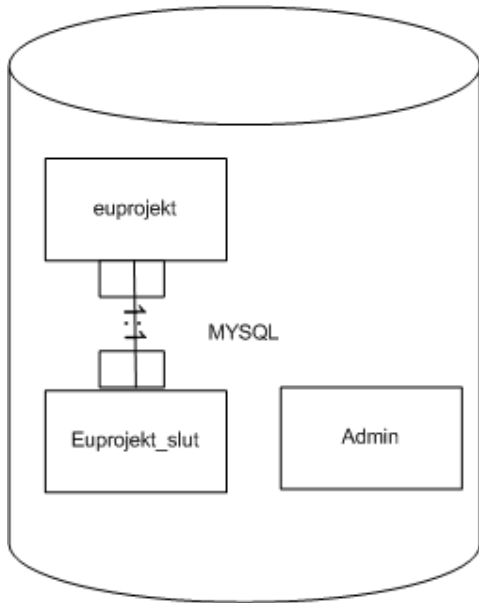
Webbtjänsten har som ansvar att upplysa klienterna om eventuella fel som uppstår och om vad som orsakat dem. Det kan handla om att klienterna skickar fel parametrar till webbtjänsten eller att databasen inte är kontaktbar. Alla dessa felmeddelanden returneras med SytemException och SoapException.

4.3.3 Databas design

MySQL databas är den valda databasen för systemet. Den ger möjligheten att lagra olika datatyper och returnerar data oberoende av förfrågningens plats se (kapitel 3 krav specifikationen).

Databasen består av tre entiteter. se figur 4.3.3. Euprojekt och Euprojekt_slut är entiteter som lagrar information om kostnader i tabellen Euprojekt och intäkter i Euprojekt_slut om varje projekt av projektadministratören. I entiteten Euprojekt lagras en primärnyckel(unikt nummer för sökning) för varje projekt. Primärnyckeln är den samma för entiteten Euprojekt_slut, vilket leder till att en relation mellan de entiteterna skapas som är 1:1. Admin entiteten lagrar användarenamn och lösenord och den har ingen relation till de andra entiteterna.

Webbtjänst för projektledare



Figur 4.3.3: Databas entiteter och deras relation.

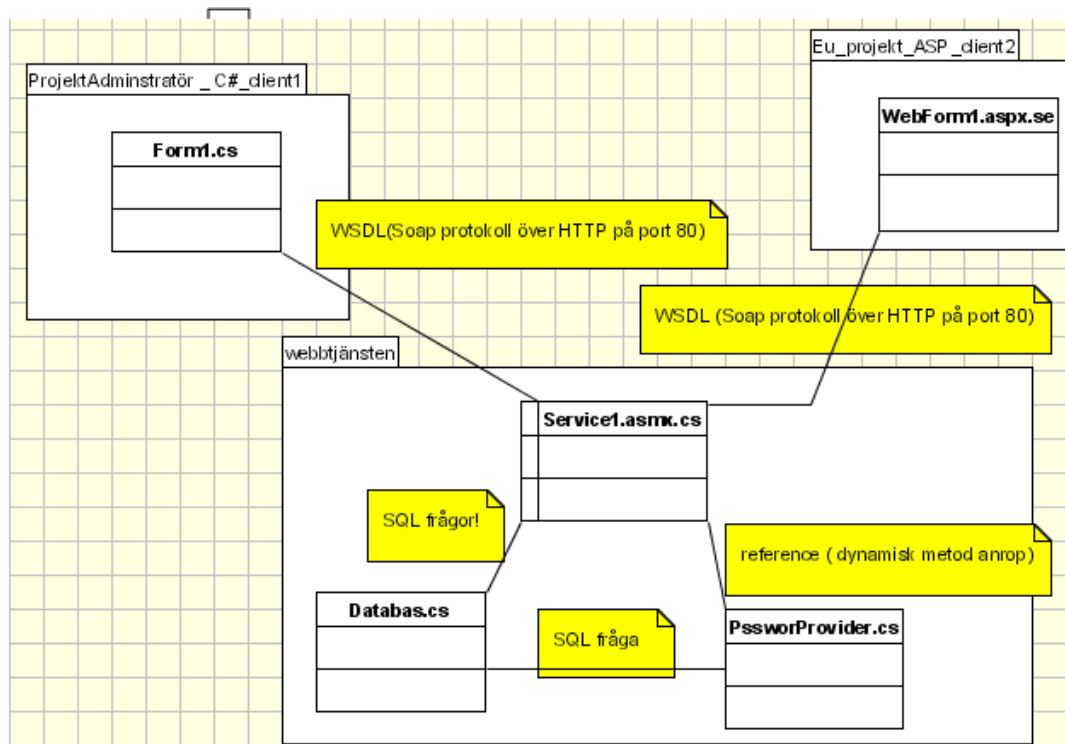
Webbtjänst för projektledare

5 Implementering av webbtjänsten och klient

I följande avsnitt presenteras implementering av de applikationer som bygger själva systemet, databasen samt klienter som står för presentation av projektrapport och lagring av projekts kostnader och intäkter. För varje applikation framgår eventuellt utvecklingsverktyg samt programspråk som använts. Vidare presenterar klassar och metoder som ingår i applikationerna.

5.1 Klassdiagram

I följande klassdiagram visas de klasserna som ingår i systemet samt de klasserna som utgör anropande klienterna.



Figur 5.1 : Klassdiagram över hela systemet

Webbtjänst för projektledare

5.2 Webbtjänstapplikationer

Webbtjänstapplikationer har en central roll i systemet när det gäller att tillhandhålla tjänster för klienterna och förbindelse med databasen. Webbtjänsten har webbmetoder vilka anropas av klienterna, och interna metoder vilka sköter kommunikation och SQL frågorna till databasen för lagring och returnering av data. Webbtjänstapplikationer har utvecklats med Microsoft Visual Studio .Net miljö och programmeringsspråk C#.

5.2.1 Klassen Service1.aspx.cs

Service1 är den klass som innehåller webbmetoder vilka tillhandhåller klienterna följande tjänster:

- Webbtjänstapplikationer skall fungera som ett lager mellan databasen, projektledares/projektadministratörens applikationer. Dessa applikationer kan skicka information till databasen för lagring genom att anropa vissa av de webbmetoderna i klassen Service1 och vilka i sin tur anropar interna metoder för databas kopplingen och SQL frågorna.
- För att projektadministratörs applikation skall lagra information för varje projekt skall applikationen identifiera sig till klassen Service1. Detta sker genom att applikationen anropar klassens metod som tolkar de säkerhets elementer som innehåller användarenamn och lösenord. Detta ger klientapplikationerna rätt till att anropa övriga webbmetoder.
- Klassens huvudmål är att tillhandhålla en tjänst som hjälper klienterna att göra kalkylering och presentering.
- En annan tjänst är att klassen upplyser klienterna när det uppstår något fel när databasen är okontaktbar och när webbmetoder anropas med fel parametrar.

Klassen Service1 innehåller följande metoder:

- validUser() : metoden används för autentisering av klients applikationer.
- GetFirstUsernameToken(): metoden används för returnering av säkerhet symboler.
- insertIntoProject(): metoden används för lagring av projekt kostnader.
- insertIntoProjectFinsh(): metoden används för lagring av bidrag efter projektslut.
- MySoapException(): metoden används för att returnera detaljerad felmeddelande.

Webbtjänst för projektledare

- `getProjectNamn()`: metoden används för att returnera lagrad projektnamn.
- `getProjectId()`: metoden hämtar projektets primärnyckel från databasen.
- `getProjectCost()`: metoden hämtar projektkostnader som löner, bidrag och övriga kostnader från databasen.
- `getProjektsBidrag()`: metoden hämtar bidrags summor efter projektets slut från databasen.

5.2.2 Klassen Databas.cs

Klassen Databas förser webbtjänsten med lagrad data för varje projekt. Klassen upprättar kommunikation med databasen och exekverar SQL-frågorna som kommer från Service1 klassen. Databas klassen returnerar data på begäran av Service1 klassen.

Det är bara Databas-klassen som kommunicerar med databasen. Klienterna belastar inte databasen och vet inte heller var databasen placeras.

Klassen innehåller följande metoder:

- `getNamePass()`: Metoden returnerar en dataset som innehåller användarenamn och lösenord till Service1 klassen.
- `setProjectInfo()`: Metoden används för att utföra en SQL sats som begärs av Service1 klassen.
- `getProjectName()`: Metoden returnerar lagrad projekt namn.
- `getProjektID()`: Metoden hämtar primärnycklar från Euprojekt tabellen i databasen och returnerar dem till Service1 klassen.
- `rowCount()`: Metoden returnerar ett antal rader från Euprojekt tabellen som lagrar projektkostnader till Service1 klassen.
- `getProjectCost()`: Metoden returnerar kostnader från Euprojekt tabellen till Service1 klassen.
- `getProjektBidrag()`: Metoden returnerar bidrags summor från Euprojekt_slut till Sercivce1 klassen..

5.2.3 Klassen PasswordProvider.cs

Klassen PasswordProvider ärver från interface klassen IPasswordProvider som ingår i paketet WSE 1.0 och klassens namn måste registreras i webbtjänstens Konfiguration

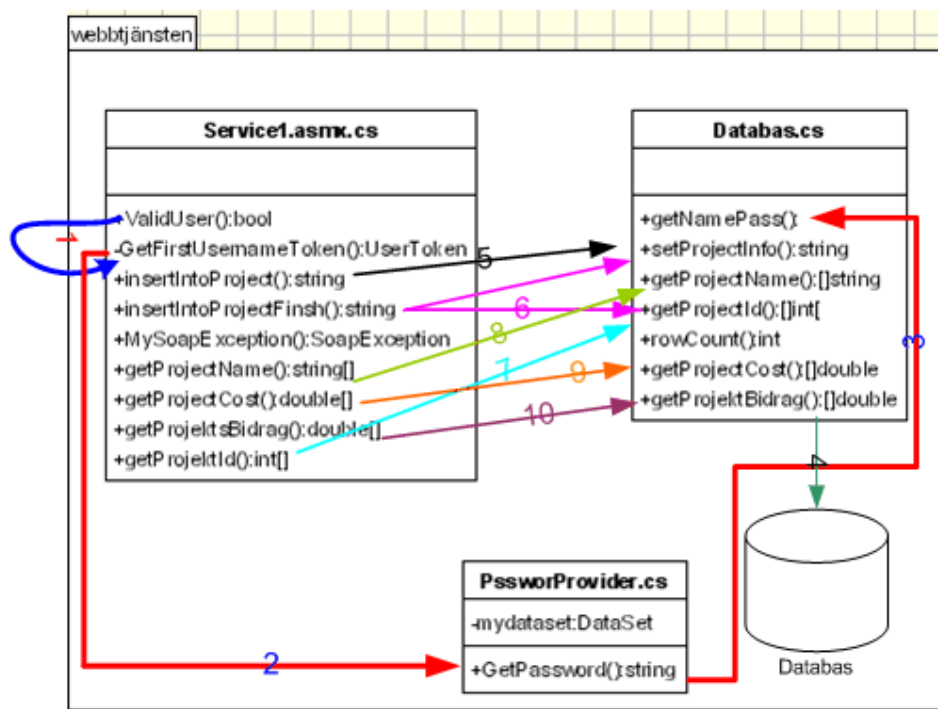
Webbtjänst för projektledare

filen. Klassen har en metod ,GetPssword, vilken anropas av Service1 klassen för att matcha klient identiteten som skickas hit.

För att metoden GetPassword skall användas effektivt av Service1 klassen, returnerar metoden ett lösenord som matchar det användarenamn som skickas av klassen Service1 som parametrar till GetPassword. Det returnerade lösenordet lagras i en tabell i databasen (se kapitel 6).

5.2.4 Detaljerat klassdiagram över metoder som ingår i webbtjänsten

I följande diagram visas de anropade klass metoderna som ingår i webbtjänsten. Vidare beskrivs hur kommunikationen mellan metoderna i en och samma klass och med andra klassar i samma paket sker.



Figur 5.2.4 : Kommunikationer mellan metoder i del systemet webbtjänsten

1: Metoden ValidUser i klassen Service1 anropar intern metod GetFirstUsernameToken för att kontrollera om säkerhet symbolerna har skickats med Soap medellandet från en klient till webbtjänsten.

2: Metoden ValidUser i klassen Service1 anropar metoden GetPassword dynamiskt i klassen PasswordProvider för att klientens lösenord som har skickats som parameter till den anropade metoden skall jämföras mot lösenord som är lagrat i databasen.

3: Metoden GetPassword anropar metoden GetNamePass i klassen Databas genom med klass referens för att hämta lagrade lösenord från databasen.

Webbtjänst för projektledare

4 : Metoden GetNamePass hämtar lagrade lösenord från databasen.

5: Metoden insertIntoProjekt i klassen Service1 är en webbmetod och anropas av en klient för att lagra projekts kostnader. Den anropar i sin tur metoden setProjektInf i Databas klassen med klass referens för att lagra kostnaderna för ett projekt. Metoden har en sträng i form av en parameter och då kan metoden anropas med SQL frågor.

6: Metoden insertIntoProjektFinsh i klassen Service1 är en webbmetod och anropas för att lagra projektets intäkter och vilket i sin tur anropar metoden setProjektInf med klass referens för att lagra intäkter för projektet.

7: Metoden getPerjektId i klassen Service 1 är en webbmetod och anropas för att returnera primärnyckeln från entiteter euprojekt i databasen som hämtas från metoden getProjektId i klassen Databas. Projektets Primärnycklar behövs för att klienterna skall kunna välja ett projekt för rapporters presentering.

8: Metoden getProjektName i klassen Service 1 anropas av klienter för att presentera alla projektnamn som är lagrade för användare. Metoden anropar getProjektName i klassen Databas som i sin tur hämtar alla projektnamn från databasen och skickar dem i en fält av texter.

9: Metoden getProjektCost i klassen Serivce 1 är en webbmetod och anropas för att returnera kostnader för ett projekt. Metoden anropar metoden getProjektCost i klassen Databas för att hämta kostnader för ett projekt som är lagrade i databasen. Kostnaderna returneras i ett fält av decimalt tal till metoden i klassen Service 1 som sin tur returnerar dem i ett WSDL dokument till den anropade klient applikationen.

: Metoden getProjektBidrag i klassen Serivce 1 är en webbmetod och anropas för att returnera intäkter för ett projekt. Metoden anropar metoden getProjektBidrag i klassen Databas för att hämta intäkter för ett projekt som är lagrat i databasen. Intäkterna returneras i ett fält av decimaltal till metoden i klassen Service 1 som sin tur returnerar dem i en WSDL dokumentet till den anropade klient applikationen.

5.3 klientapplikationer

Klienter till webbtjänsten är av två typer, ena är projektadministratörens applikation och andra är projektledarnas applikationer. Båda typerna har utvecklats i verktyget Microsoft Visual Studio .Net miljö och programmeringsspråk C#. Kommunikation mellan webbtjänsten och klientapplikationerna baseras på WSDL dokument.

Webbtjänst för projektledare

5.3.1 Klassen Form1.cs

Klassen ingår i en Windows applikation ProjektAdministratör och används av projektsamordnare för att lagra projekt information för varje projekt som senare används av projektledarna som stöd för att göra sina ekonomiska uppskattningar.

Klassen kalkylerar och presenterar projekt kostnader och intäkter. Kostnaderna är löner till projektanställda och projektledarna, samhällets kostnader i form av bostadsbidrag och socialbidrag som går till projektanställda och övriga kostnader. Intäkter är bidragens mellanskillnad vid projekts slut, avgifter och skatter. Vidare räknar klassen den summa som är kvar efter kalkyleringen av intäkterna och kostnaderna med en räntesats på 4 %. Detta görs för att räkna ut den period som behövs för att betala tillbaka kvarsumman till samhället.

Skälet till att det är klientapplikationen och inte webbtjänsten som utför kalkyleringar för vidare presentation är för att endast de behöriga klienterna skall kunna utnyttja webbtjänsten.

Klassen har följande metoder:

- validDigits(): Metoden kollar siffra eller bokstav.
- addUserNamePass(): adderar användarenamn och lösenord till Soap meddelande och TTL värde sedan kickas till webbtjänsten.
- insertIntoDatabas() metoden används för lagring av kostnader.
- setProjektInt() metoden används för lagring av intäkter.
- HrrorHandel(): returnerar fel meddelande
- calc(): kalkylerar och presenterar projektinformation

5.3.2 klassen WebForm1.aspx.cs

Klassen ingår i en webb baserad applikationsgrupp EuProjekt_ASP_client2 som är utvecklad med ASP.Net och programmeringsspråk C#. Applikationen används av projektledarna.

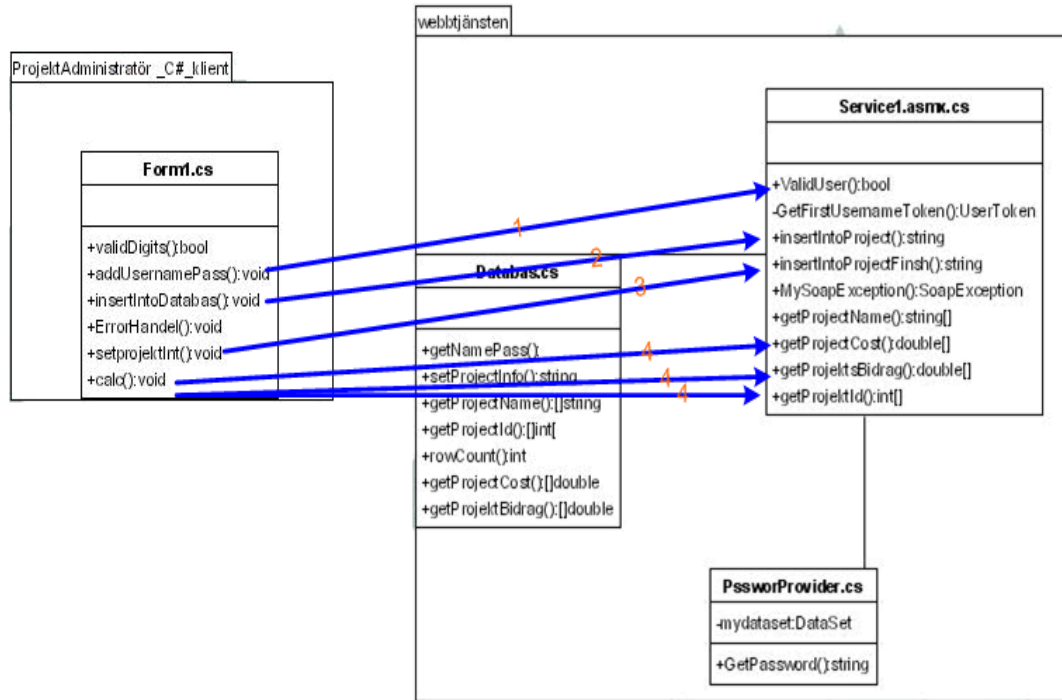
På samma sätt som ProjektAdministratörsklassen får denna klass projektinformationen från webbtjänsten för kalkylering och presentering. Kalkyleringen utförs också på samma sätt som klassen ProjektAdministratör klass se 5.2.1.

Klassen har nästan samma metoder som i 5.2.1 för inloggningen till webbtjänsten och för att få projekts information om kostnader/intäkter. Dock har klassen inte de metoder som kan användas för att lagra projekts information.

Webbtjänst för projektledare

5.3.3 Detaljerad klassdiagram över metoder som ingår i klass klienten

I följande diagram visas metoder som ingår i klass klienten ProjektAdministratör. Vidare beskrivs hur kommunikationen med webbtjänsten sker.



Figur 5.3.3: Kommunikation mellan klients metoder och webbmeter

1 : Genom metoden `addUserName` i klassen `Form1` adderas användarnamn och lösenord som säkerhetslementer i Soap meddelande som adderas med ett TTL värde. Metoden `addUserNamePass` anropar sedan metoden `ValidUser` i klassen `Service 1` med en skapande proxy referens för att skicka WSDL dokumentet som innehåller säkerhets elementer.

2 : Metoden `insertIntoProjekt` i klassen `Form1` anropar webbmeteroden `insertIntoProjekt` i klassen `Service 1` som i sin tur adderar projekt kostnader i databasen.

3 : Med metoden `setProjektInt` i klassen `Form1` utförs flera operationer:

- Klienten väljer ett projektnamn som motsvarar projektets primärnyckel i entiteter `Euprojekt` i databasen.
- Metoden `setProjektFinish` i klassen `Service1` anropas med ett heltal som en parameter som motsvarar den valde primärnyckeln för att lagra projekts intäkter.

Webbtjänst för projektledare

4: Metoden `calc` i klassen `Form1` användes för kalkylering och då utförs följande operationer:

- Ett projekts primärnyckel väljs av klienten.
- Metoden `getProjektCost` i klassen `Service1` anropas med ett heltal som parameter som motsvarar primärnyckeln. Sedan returnerar metoden kostnaderna för just det valde projektet.
- Metoden `getProjektBidrag` i klassen `Service1` anropas med ett heltal som parameter som motsvarar primärnyckeln. Sedan returnera metoden intäkterna för just det valde projektet.
- Kostnader och intäkter presenteras för klienten efter kalkylering.

Webbtjänst för projektledare

6 Webbtjänsters säkerhet över Internet

I följande avsnitt beskrivs de krav som ställs för att skapa säkra webbtjänster. Vidare analysers en del av de tekniska mekanismerna som behandlar säkerhet i webbtjänster på nätverksnivå och applikationsnivå.

6.1 Krav på webbtjänstssäkerhet över Internet

Kravet på säkra webbtjänster är lika stort som det krav som ställs på webb baserad programvara. Det finns olika tänkbara krav för att meddelande skall utbytas mellan webbtjänsten och klientapplikationer på ett säkert sätt [5]:

1. Privatisering (privacy): Det innebär att överföring av information över Internet sker på det sättet att informationen inte kan avlyssnas eller passera tredje part utan samtycke från den som är behörig. Detta kan tillämpas genom[5]:
 - Sändning av en serie av siffror vilka är inloggningsinformation som går genom hashfunktioner.
 - SSL(HTTPS) som används för att kryptera och skicka data genom säkra kanaler.
 - XML kryptering
2. Autentisering (authentication): Innebär att sändare och mottagare av meddelande verifierar varandra. Det finns många tekniska mekanismer som kan tillämpas för autentisering innan mottagares resurser används. Teknikerna är följande :
 - Grund HTTP: denna används med andra säkerhet teknologier som Kerberos, SSL, Digit autentisering och IPsec för begränsning av IP adresser och lösning av problemet (IP spoofing)[5].
 - XML-Signatur och XML-Kreptering.
3. Integritet (integrity): Innebär att information som sänds eller mottages försäkras och inte kan modifieras eller ändras. Detta kan åstadkommas med SSL, XML-Signatur som digital signerar meddelande med ett certifikat som X.509 och XML-Kryptering för kryptering en del eller hela meddelande med public key.

Webbtjänst för projektledare

4. Auktorisering (authorization): Innebär att tillgång till servens/webbtjänsts resurser hanteras och skyddas enligt användares identitet (credential). Med digit autentisering och SSL kan kravet uppfylls.
5. Icke-förnekande (non-repudaiton): Innebär ett lagligt bevis på att ett meddelande har sänts eller mottagits. SSL och Digital signatur gör det möjligt att bygga webbapplikationer som uppfyller detta krav.

6.2 Nätverkssäkerhet teknik för webbtjänster

I det följande kapitel behandlas två tekniska mekanismer för transport av information över nätverk med säkerhet.

6.2.1 IPsec och VPN (Virtual Private Networks)

VPN kan användas för koppling av två nätverk över Internet. VPN använder Internet infrastruktur som redan finns i bruk och därför är tekniken mer ekonomiskt än privat nätverk kryptering. VPN erbjuder också samma tjänst som ett lokalt nätverk erbjuder fasten över ett publiknätverk. Med VPN kan säkra webbtjänster skapas genom att etablera en säker tunnel där data kan passera mellan multipla nätverk.

IPsec (Internet protocol Security) utvecklas av (IETF) och är en av de teknologier som används för att skapa säkra tunnlar, försäkra data privatisering, integritet och verifiering genom användning av publik nyckel kryptering. IPsec använder standard transföring av data packet mellan två nätverk över Internet via IP (Internet Protokoll).

VPN kan användas för tre olika säkerhetskoncepter, verifiering av användares identitet, kryptering av data som sänts över nätverk och kontroll av kollektiva informationer. IPsec protokollet stödjer de här koncepterna genom användning av:

- AH (Authentication Head): bifogar extra information till varje packet för säkerhet.
- ESP (Encapsulating Security Palyload): för kryptering av data med symetriy nyckel för att skydda data från avlyssnare medan data IP packet sänts från en maskin till en annan.
- IKE (Internt Key Exchange): är ett nyckelväxlingsprotokoll som används i IPsec för att bestämma säkerhets begränsningar och verifiering av krypterings nyckel.

För att etablera VPN i ett nätverk krävs det att alla användare av tekniken har samma mjuk/hård vara och därför tekniken är svårt att hanteras [11].

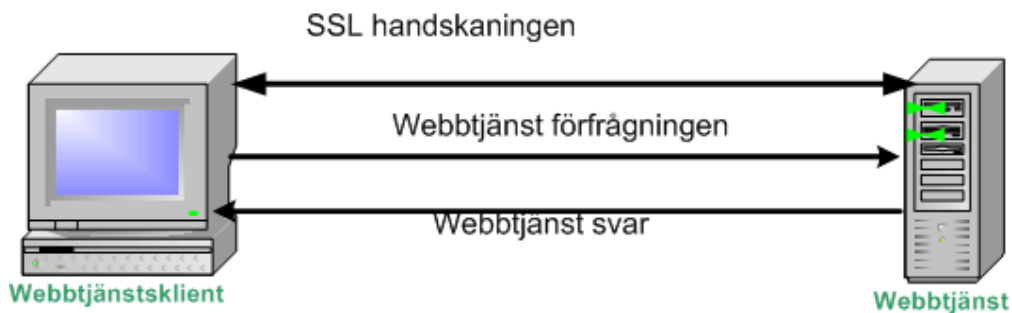
Webbtjänst för projektledare

6.2.2 SSL/TLS (Secure Socket Layer and Transport Layer Security)

SSL/TLS är den mest användbara tekniken som webbserver\webbtjänster och IIS (Internet Information Server från Microsoft) stödjer den[7]. Både webbserver och klienter använder samma infrastruktur när en order går ut till URL med SSL/TLS.

SSL använder digital signering och digital certifikat som skall vara en VeriSign eller en auktoriserad (CAs) över Internet. Samma certifikat skall bindas med webbserver\webbtjänster och med klienter. Klienterna konfigurerar certifikaten för att verifiera sig hos webbserver eller webbtjänsten[7].

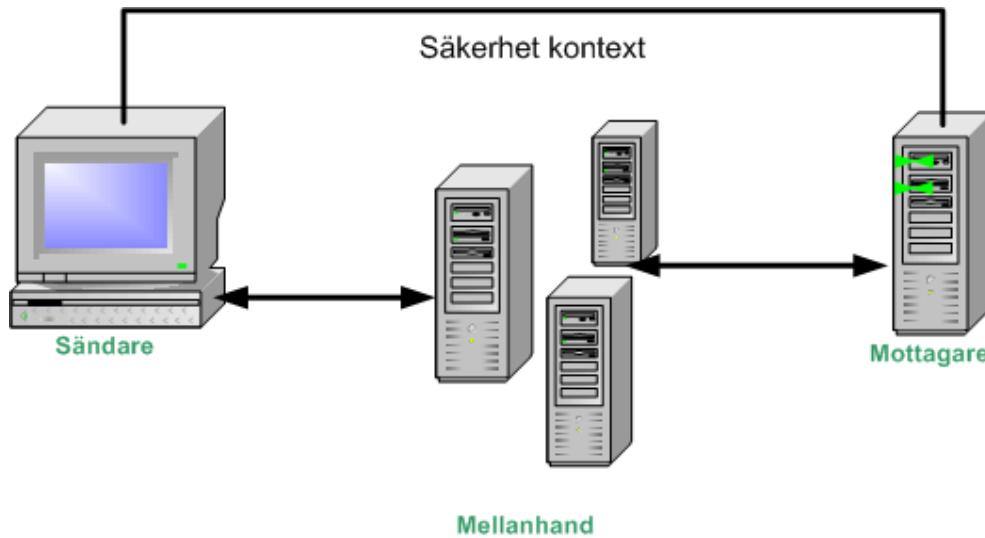
Den första handskningen mellan webbtjänsten och klienten sker med en kryptografisk teknik, vilket gör att klienten får förfoga över de tjänsterna från den webbtjänsten som den vill ha. figur 6.2.1a[7].



Figur 6.2.1a[7]: En bild över handskaning mellan klient och webbtjänst.

Det är en hög säkerhets nivå som skapas med hjälp av SSL mellan två noder, d.v.s. webbtjänster och klienter, över Internet. Dock kan och ingen kontroll tillhandhållas om webbtjänsten bestämmer eller behöver skicka en förfrågning från klient till andra noder för svar (response). Därmed är SSL inte den lämpliga säkerhets lösning för webbtjänster när det handlar om att en eller flera mellanhänder uppstår mellan webbtjänsten och klienten [7]. figur 6.2.1b. En lösning på problemet kan vara att använda sig av HTTPS. HTTPS säkrar då kommunikationen genom att skicka HTTP frågor och svar mellan en end-to-end över en SSL förbindelse.

Webbtjänst för projektledare



figur 6.2.1b säkre kanalen mellan sändare och mottagare[8].

6.3 Autentisering och auktorisering teknik för webbtjänster

Identifiering till en webbtjänst och användning av webbtjänstens resurser som har skyddats kan tillämpas genom flera tekniker som Kerberos, Biometrics, Single Sign-On, HTTP grund säkerhet och Windows authentication. I det följande kapitel behandlas teknikerna Kerberos, Windows authentication och HTTP grund säkerhet.

6.3.1 Kerberos

Kerberos är ett nätverks autentisering protokoll. Det är designat så att den kan erbjuda stränga autentisering för klient/server applikationer genom att använda säker nyckel kryptering. Protokollet erbjuder också privatisering och data integritet efter att både klient och server har identifierat sig till varandra. [12]

Autentisering i Kerberos hanteras i försthand från Kerberos system och i andra hand från biljett TGS (Ticket Granting Service). Hela systemet liknar tekniken där nycklar distribueras. Kerberos system verifierar klientens identitet till en TGS biljett sedan verifierar TGS klientens rätt till en specifik nätverks tjänst.

6.3.2 Windows autentisering

En annan teknik för att skapa säkra webbtjänster är Windows autentisering som innebär verifiering och inte kryptering av data. Tekniken tillämpas då information skickas från klientapplikationer till webbtjänster i klart text. [7]

Webbtjänst för projektledare

6.3.3 HTTP bas säkerhet

Ett annat sätt att skapa säkra webbtjänster är HTTP identifikationsmekanismer som redan finns färdig för HTTP. HTTP med IIS (Microsoft Information Server) erbjuder webbtjänster att verifiera användare innan tillåtelse för användning av webbtjänstens resurser lämnas. Webbtjänsten kontrollerar då användarenamn och lösenord gentemot en databas.

HTTP använder nycklar-kryptering och digit autentisering som krypterar klients identitet (användarenamn och lösenord) med base64-krypterd, vilket kan lätt dekryptera klientens identitet. Denna teknik har stöd av många webbläsare. IIS tillåter klienter access till webbtjänstsresurser om klientens identitet matchar ett godkänt inloggningskonto.

HTTP används med andra teknologier som SSL och Kerberos för att uppnå sträng säkerhet. HTTP kombineras också med IPsec för att begränsa IP adresser till en webbtjänst och lösa problemet med IP spoofing[5].

6.4 Säkerhetstillämpning för webbtjänst på applikationsnivå

Webbtjänster levererar tjänster till klientapplikationer. Detta sker genom utbyte av WSDL dokument som är XML schema baserad på Soap protokollet (se kapitel 2). För att skapa en säker webbtjänst på applikationsnivå, finns det olika teknologer som webbtjänstssäkerhet, XML signatur, XML kryptering, XML Key management Specification (X-KMS) och Security Assertion Markup Language SAML. I följande avsnitt beskrivs och implementera webbtjänstssäkerhets teknologi[5].

6.4.1 Introduktion till webbtjänstssäkerhet

Webbtjänstssäkerhet (WS-Security) är en ny säkerhetsspecifikation för webbtjänster som publicerades av IBM, Microsoft och VeriSign, april 2002. Tekniken är ännu inte en standard men chansen är stor att det kommer att bli det i framtiden. Det grundat på att skapas en standard säkert policy som alla affärsprocesser skall följa[5].

Webbtjänstssäkerhet är en del av ett större ramverk som kallas *Global XML Web Services Architecture (GXA)*. Där ingår även WS-Routing, WS-Coordination, WS-Inspection, WS-Referalsamt WS-Transaction.

Webbtjänstssäkerhet föreslår en mängd elementer som tillägg i Soap. De elementerna är:

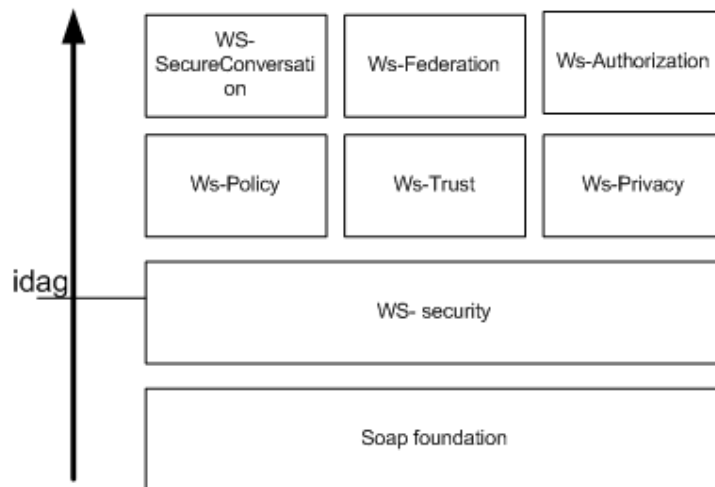
- Security : Definierar säkerhet mekanismen.
- UsernameToken: Definierar information om användarnamn och lösenord.

Webbtjänst för projektledare

- BinarySecurityToken: Definierar binära säkerhetssymboler som Kerberos biljetter och X.509.
- KeyInfo: Definierar key information.
- Signature: Definierar signatur:
- ReferenceList: Definierar innehållet i krypterad meddelande.
- EncryptedData: Definierar krypterad data.

De ovan nämnda elementerna kan användas som tillägg i Soap meddelande för att åstadkomma integritet, konfidentiell och autentisering på webbtjänster. Specifikationerna kan också ge standard mekanism för webbtjänsts applikationer så att säkra och signerade meddelanden kan utbytas [4][3].

Webbtjänstsäkerhet utgör grunden för en mängd framtida säkerhetsspecifikationer och även en mängd specifikationer som för närvarande utvecklas se figur 6.4.1 [9].



Figur 6.4.1: Framtida specifikationer som bygger på webbtjänstsäkerhet

WS-Policy : definierar en säkerhets policy som skall vara kompatibel med standard säkerhets policy.

WS-Trust : beskriver en modell för att etablera en direkt eller 3:e part tillförlitlig relation. Det är viktigt när ett certifikat används i Soap meddelande.

Webbtjänst för projektledare

WS-Privacy : definierar hur webbtjänster formar och utför privatpraktik t.ex. Microsoft kräver webbplats som använder deras Passports tjänster för att gå med på P3P som har inflyttande på WS-Privacy.

WS-Secure Conversation : beskriver hur ändring av meddelande skötas.

WS-Federation : beskriver hur en tillförlitlig relation hanteras när det gäller teknisk autentisering.

WS-Authorization : definierar hantering av autentiserings data och säkerhets policy.

6.4.2 Webbtjänstsäkerhets princip

Webbtjänstsäkerhet erbjuder en mekanism för att överföra säkerhetssymboler från klienter till webbtjänster, t.ex. användarnamn och lösenord samt digitala certifikat. Det krävs ingen speciell typ av säkerhetssymbol, utan denna mekanism är tänkt att vara utbyggbar till att stödja en mängd olika format. Webbtjänstsäkerhet stödjer multipla säkerhetssymbols format till exempel en klient kan ge bevis på sin identitet och bevis på att han/hon har en certifikat som X 509. Det finns även specificerat hur kryptering av binära säkerhetssymboler som Kerberos biljetter och X.509 certifikat skall gå till. Med hjälp av dessa och i vissa fall digital signering kan identiteten för ett meddelande kontrolleras.

Webbtjänstsäkerhet är flexibel och används för att erbjuda stöd för brett och varierat säkerhets tekniker som PKI, Kerberos och SSL. [3].

Webbtjänstsäkerhet erbjuder integritet och konfidentialitet för meddelande och det kan tillhandhållas genom att kryptera eller signera digitalt *Body* eller/och *Head* meddelande. Integritet kan skaffas genom XML signering som kombinera med säkerhetssymbol för att säkra att meddelande inte modifieras under transporting. Med kravet på konfidentiell kan det erhållas genom XML kryptering i kombination med säkerhetssymboler för att hålla Soap meddelande konfidentiellt.

6.4.3 Säkerhetsdesign för webbtjänsten i ett lokalt nät

Webbtjänsten och klienterna skall byggas i ett lokalt nät. Lokalnät som är ett LAN nätverk (Lokal Area Network) där flera datorer och andra resurser är uppkopplade tillsammans. I det lokala nätet skyddas webbtjänsten från Internetattacker med hjälp av hård/mjukvara brandvägg vilken kontrollerar trafiken mellan intranät och Internet

Webbtjänsten skall skyddas i det lokala nätet mot attacker som kommer från interna användare som vill ha tillgång till information som de inte har rätt till. Det finns då flera tekniska sätt för att skydda webbtjänsten. Exempelvis Kerberos för autentisering, IPsec för att bygga säkra kommunikation mellan datorer i lokalnät, SSL som används för

Webbtjänst för projektledare

nätverks kommunikation mellan två datorer, webbtjänstsäkerhet som bygger på applikationsnivå och kombineras med SSL, Kerberos, XML kryptering och XML signatur vilka erbjuder autentisering och data integritet.

Design valet för att bygga en säker webbtjänst för autentisering är webbtjänstssäkerhet där används användarenamns säkerhetsymboler (username token). Där adderas användarenamn och lösenord som säkra elementer till Head Soap meddelandet (se kapitel 6).

6.4.4 Implementering av webbtjänstsäkerhet

Webbtjänsten som har skapats tillhandhåller klienterna tjänster efter autentisering till systemet och detta sker genom att klienterna anropar en webbtjänstsmetod för att logga in till webbtjänsten. Klienternas identitet (användarenamn och lösenord) sänds i klartext vilket kan avlyssnas och modifieras av obehöriga.

För att lösa problemet kan olika teknologier användas (se kapitel 6). Den valda tekniken för att lösa problemet är webbtjänstsäkerhet. Här adderas säkerhets relaterade information till Soap meddelande för autentisering. Den valda metoden är användning av användarenamn - säkerhetsymboler som adderas till Soap meddelande som utbyts mellan webbtjänsten och klienterna. Metoden sänder klientens användarnamn och krypterar lösenord som går genom en hashfunktion till webbtjänsten exempel 6.4.4[13]. Vidare matchar webbtjänsten klientens identitet gentemot databasen som har krypterade lösenord. Om klienten har godkänd identitet då tillåts den att utnyttja systemets resurser.

```
<!-- No Password -->
  <UsernameToken>
<Username>Bob</Username>
  </UsernameToken>

  <!-- Clear Text Password -->
  <UsernameToken>
<Username>Bob</Username>
<Password Type="wsse:PasswordText">Opensezme</Password>
  </UsernameToken>

  <!-- Digest: SHA1 hash of base64-encoded Password -->
  <UsernameToken>
  <Username>Bob</Username>
  <Password Type="wsse:PasswordDigest">
    QSMAKo67+vzYnU9TcMSqOFXy14U=
  </Password>
  </UsernameToken>
```

Exempel 6.4.4: För XML säkerhets elementer med UsernameToken.

Webbtjänst för projektledare

För att uppfylla autentisering kravet i webbtjänstsäkerhet används (Web Services Enhancements – WSE version 1.0). WSE har utvecklats av Microsoft och sitter i toppen på .Net ramverket som stödjer skrivning och konsumering av webbtjänster. WSE används för att skapa säkra .Net XML webbtjänster. WSE erbjuder klasser som används av webbtjänster för att tolka säkerhets elementer i Soap meddelande. Dessa klasser kan också användas av .Net klienter för att lägga säkerhet elementerna till Soap meddelande.

WSE erbjuder en rad olika metoder vilka adderar säkerhet relaterad information till Soap meddelande och de är följande [13]:

- Signerad klient identifiering genom användarenamn och lösenord, signerad med X.509 certifikat och klient binär symboler.
- Digital signatur för Soap meddelande
- Kryptering Soap meddelande
- Webbtjänst routing
- Webbtjänst filtrering

Klient autentisering med WSE implementeras genom signering av användarenamn och krypterad lösenord och som skickas med Soap i Head delen till webbtjänsten. Sedan tolkar webbtjänsten säkerhets element och matchar dem med lagrad lösenord. Detta sker genom att båda webbtjänstapplikationen och klienternas applikationer skapar referenser till WSE klasserna.

När en klient vill kontakta webbtjänsten för utnyttjande av tjänster adderas klientens användarenamn och lösenord som barn till säkerhet elementer i Head Soap meddelade och skickas till webbtjänsten. Sedan läser webbtjänsten de Head säkerhet elementerna och matchar lösenordet med ett lagrat krypterat lösenord genom dynamiskt anrop till klassen PasswordProvider som ärver från interface klassen IPasswordProvider. Klassen PasswordProvider måste registreras i webbtjänstens konfigurations fil.

Metoden i klassen PasswordProvider returnerar ett lösenord vilket används av webbtjänsten för matchning av lösenordet som skickas med Soap meddelande från klienten. För att metoden utnyttjas effektivt hämtar metoden lagrad lösenord från databasen se design kapitlet.

Genom tillämpning av den valda metoden förhindras webbtjänsten att acceptera meddelande som spelas av och skickas tillbaks vid senare tillfälle av någon obehörig (replay-attack). Detta sker genom att addera TTL värde (Time-To-Live) eller sekvensnummer till Soap meddelande som skickas till webbtjänsten.

För källkoden till metoden se appendix A, B.

Webbtjänst för projektledare

7 Slutsats

I följande avsnitt kommer jag att verifiera om de krav som jag ställt på systemet har uppfyllts. Vidare verifierar jag målen med arbetet.

7.1 Verifiering av kravspecifikation

Beträffande de krav som ställts har jag kommit fram till följande:

- Webbtjänsten är tillgängligt i ett lokalnät

Den dator som webbtjänsten körs på finns i ett lokalnätverk där klienternas datorer är i ett lokalt nät. Klienterna utnyttjar webbtjänsten genom att skapa referens med en lokaladress (är en tillfällig IP adress som DHCP protokollet tilldelar). Webbtjänsten skyddas från Internet attacker genom att blockera access till webbservern på port 80.

- Databasen lagrar och returnerar data från och till klienter genom webbtjänsten oberoende av deras plats:

MySQL är den valde databasen som körs på en dator i lokalnätet. Webbtjänsten kan kontakta den genom datorns IP nummer vilket tilldelas av DHCP.

- Webbtjänsten är ett lager mellan klienterna och databasen:

Klienterna anropar webbtjänstens metoder för lagring och returnering av data för och vidare kalkylering och presentering. Det är webbtjänsten som skapar kopplingen och SQL frågorna till databasen.

- Webbtjänsten tillhandhåller tjänster till behöriga Klienter genom autentisering:

Klienten skickar sin identitet d.v.s. användarnamn och krypterat lösenord med Soap protokollet till webbtjänsten. Identiteten placeras i Soap Head delen som säkerhet elementer. Webbtjänsten kontrollerar sedan om lösenordet är ett godkänd lösenord för att klienten skall tillåtas att utnyttja webbtjänstens tjänster.

- Spara data om projekt:

Det är bara projektsamordnare som kan spara data om varje projekt. Genom projektsamordnares applikation skickas information till webbtjänsten. Sedan bearbetar webbtjänsten informationen enligt den strukturen som projekt tabellen har i databasen.

- Klient applikationer framvisar projektrapport:

Klienten anropar en webbtjänstens metod efter verifiering av klientens identitet av webbtjänsten. Webbtjänsten returnerar lagrade information för ett bestämt projekt till den anropande klienten. Klient applikationen kalkylerar och presenterar sedan informationen.

- Webbtjänsten måste generera fel meddelande till klienter:

Webbtjänst för projektledare

Webbtjänsten fångar SoapException och SystemException om exempelvis något felet uppstår i databas kopplingen eller SQL frågor. Vidare skickar Webbtjänsten fel rapport till anropande klienter när en ej godkänd lösenord skickas med säkerhet elementerna och obehöriga klienter försöker utnyttja webbtjänstens resurser.

- Webbtjänsten ger information om kostnader/intäkter för ett projekt som lagrats i databasen:

Webbtjänsten returnerar kostnader och intäkter för ett bestämt projekt till anropande klienter. Kostnaderna och intäkterna är lagrade i databasen och skickas i två samlingar (array av typen double) först till webbtjänsten och vidare till anropande klienter.

- Klienterna kalkylerar kostnader/intäkter och presenterar rapport för varje sparat projekt, se (kapitel 5.2).

7.2 Verifiering av mål

Resultatet har visat att målet med arbetet har uppnåtts. Webbtjänsten tillhandshåller behöriga klienter tjänster som är följande. Webbtjänsten returnerar projekts kostnader och intäkter till klienter för vidare kalkylering och presentering. Istället för att webbtjänsts applikation själv kalkylera lagrad projekts information och returnera rapporten har det lämnats som uppgift för klients applikationer att utföra kalkyleringen. Anledning är att inte belasta nätverket med onödiga dataöverföringar och att bara klients applikationer skulle kunna utnyttja webbtjänsten.

Webbtjänsten fungerar som ett lager mellan klienterna och databasen. Databasen kan belastas med många och samma SQL frågor och för att undvika detta rekommenderas att projekts information från databasen sparas i en XML fil för svar på nästa fråga från klienterna.

Två typer av klient applikationer har utvecklats. Den ena är projektsamordnares applikation som körs på en lokal dator och jobbar gentemot databasen för lagring och läsning genom webbtjänsten. Den andra är webbapplikation som läser data från databasen genom webbtjänsten och är tillgängligt för flera användare.

Systemet fungerar i ett lokalt nätverk och är skyddad mot Internet attacker med en brandvägg som kontrollerar trafiken mellan det lokala nätet och Internet. Brandväggen skyddar inte webbtjänsten från attacker som kommer från interna användare som exempelvis modifierar data som skickas från behörig klient och skickas vid andra tillfälle av obehöriga i samma nätverk. Som lösning på detta problem implementeras WES version 1.0 som tillhandshåller en rad av säkerhetspecifikationer till webbtjänsten. Tekniken används för att behörig identitet skickas med säkerhet till webbtjänsten.

WSE föredras framför andra teknologier som har analyserats i kapitel 6 för att WSE är baserad på applikationsnivå och är därför lätt att implementeras. Dessutom är den inte

Webbtjänst för projektledare

resurs krävande för att WSEs klasser kan laddas ner gratis. En annan fördel med WSE är att man kan skapa samma säkerhets policy oavsett om klient förfrågningarna går via en eller flera mellanhand.

Emellertid finns det en nackdel med WSE och det är att den bara är baserad på .Net applikationer och därför inte kan användas till applikationer som har skrivit med programmeringsspråket Java.

Slutligen bör det påpekas att det finns ytterligare tekniker för att bygga säkra webbtjänster över Internet och som fyller dem fem grundliga säkerhetskrav men varje teknik har sina fördelar och nackdelar se (kapitel 6)

8 Källförteckning

[1] Web Services Security (WS-Security) Version 1.0 April 5, 2002 Autho Microsoft, IBM and VeriSign <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-security.asp>

[2] Programming C# by Jesse Liberty second Edition 2002

[3] Kapil Apshankar WS-Security security for Web Services

<http://www.webservicesarchitect.com/content/articles/apshankar04.asp>.

[4] Brian E. Travis, MAE Ozkan Web Service Implementation Guide 2002

[5] Deitel Web Services A Technical Introduction 2002

[6] Web Services Activity by W3c <http://www.w3c.org/2002/ws/>.

[7] Secure Your Web Services by Geir Olsen

http://www.ftponline.com/wss/2003_03/magazine/columns/security/.

[8] Web Services Security and More: The Global XML Web Services Architecture (GXA) By Joseph M. Chiusano, Booz Allen Hamilton.

<http://www.developer.com/design/article.php/2171031>.

[9] Security in a Web Services World: A Proposed Architecture and Roadmap by IBM and Microsoft 1 April 2002

<http://www.ibm.com/developerworks/webservices/library/ws-secmap/>.

[10] Svenska datatermgrupp

Webbtjänst för projektledare

<http://www.nada.kth.se/dataterm/>

[11] IETF

<http://www.ietf.org/html.charters/ipsec-charter.html/>

[12] MIT

<http://web.mit.edu/kerberos/www/>

[13] WS-Security Authentication and Digital Signatures with Web Services Enhancements by Matt powell 2002

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwse/html/wssecauthwse.asp>

A webbtjänsts källkod

1. Srvicel1.asmx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Xml;
using System.Diagnostics;
using Microsoft.Data.Odbc;
using System.Text;
using System.Globalization;
using System.Threading;
using Microsoft.Web.Services;
using Microsoft.Web.Services.Security;
using System.Security.Cryptography;

namespace projekt_WS
{
    /// <summary>
    /// Summary description for Srvicel1.
    /// </summary>
    public class Srvicel1 : System.Web.Services.WebService
    {
        Databas mydatabas=new projekt_WS.Databas();

        /// <summary>
        ///constuktur for Srvicel1
        /// </summary>
        public Srvicel1()
        {
//CODEGEN: This call is required by the ASP.NET Web Services Designer
```

Webbtjänst för projektledare

```
        InitializeComponent();
    }

    #region Component Designer generated code

    //Required by the Web Services Designer
    private IContainer components = null;

    /// <summary>
    /// Required method for Designer support - do not
modify
editor.
    /// the contents of this method with the code

    /// </summary>
    private void InitializeComponent()
    {
    }

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    protected override void Dispose( bool disposing )
    {
        if(disposing && components != null)
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #endregion

    /// <summary>
    /// varifierar klineter och lagra en datatyp av
bool i cachMinne
    /// </summary>
    ///<param name="sName"> username</param>
    ///<param name="spassword">password</param>
    ///<returns> true if username and password are
valid</returns>

    [WebMethod(Description="metoden anropas för autentisering!!")]
    public bool validUser(string sName, string spassword)
    {
        bool idadmin=false;

        try
        {
            DataSet myDataSet=mydatabas.getNamePass("select *
from Admin");

            foreach(DataRow myRow in
myDataSet.Tables[0].Rows)
            {
                if( sName==myRow[0].ToString()&&
spassword==myRow[1].ToString())
```

Webbtjänst för projektledare

```
        {
            idadmin=true;
        }
    }

}

catch(System.Exception ex)
{

    throw MySoapException("fel uppstår i databasen");

}

Context.Cache.Insert("Autentisering", idadmin, null, System.DateTime.Now.
AddSeconds(280), TimeSpan.Zero);
    return idadmin;
}

/// <summary>
/// metoden används för kontroll av säkerhetsymboler
/// </summary>
[WebMethod(Description="metoden används för kontroll av
säkerhetsymboler")]
public bool validUser()
{
    bool retval = false;

    SoapContext requestContext =
HttpSoapContext.RequestContext;
    // Verifies that a SOAP request was received.
    if (requestContext == null)
        throw new
ApplicationException("Either a non-SOAP request was
" + "received or the WSE is not properly installed
for " + "the Web application hosting the XML Web
service.");

    UsernameToken theToken = GetFirstUsernameToken(
requestContext.Security );

    if ( theToken != null )
    {
        retval = true;

        Context.Cache.Insert("Autentisering", retval, null, System.DateTime.Now.AddSeconds(280), TimeSpan.Zero);

    }

    return retval;
}

/// <summary>
/// metoden kontrollerar symboler
/// </summary>
```

Webbtjänst för projektledare

```
/// <param name="sec">säkerhet symboler</param>
/// <returns> användarenamn symboler</returns>

private UsernameToken GetFirstUsernameToken( Security sec )
{
    UsernameToken retval = null;
    if ( sec.Tokens.Count > 0 )
    {
        // In each WS-Security SOAP header
        // there can be zero or more security tokens.
        foreach ( SecurityToken tok in sec.Tokens )
        {
            // Determines if the current security token
            // is a UsernameToken.
            retval = tok as UsernameToken;
            if ( retval != null )
            {
                // If a UsernameToken was found,
                // returns it.
                return retval;
            }
        }
        return retval;
    }
}

/// <summary>
/// metoden lgras Projekt Information och kontrollerar
/// om användare
/// är admin. detta sker gentemot en cache som håller sig
/// i 280 sekunder.
/// </summary>
/// <param name="ansalldLon">lön till projektanställda</param>
/// <param name="bbidrag">bostadbidrag till projektanställda i
/// projektsgång</param></param>
/// <param name="ledningLon">lön till projektledningen</param>
/// <param name="ovrig">övrig kostnader som tillkommer</param>
/// <param name="sbidrag">sociabidrag till projektantäalda i
/// projektsgång</param>
/// <param name="sProjekt"> namnet på projekt</param>
/// <returns>en sträng för vad som kommer hända </returns>

[WebMethod(Description="metoden för lagring av projekt information")]

public string insertIntoProject(string sProjekt, double ledningLon,
double ansalldLon, double bbidrag, double sbidrag, double ovrig)
{
    if(Context.Cache["Autentisering"]==null)
    {
        //return "Ange ditt AnvändareNamn och Lösenord";
        throw MySoapException("Ange ditt AnvändareNamn
och Losenord");
    }
    else if((bool)Context.Cache["Autentisering"]==
false)
    {

```

Webbtjänst för projektledare

```
        //return "Du har inte Administratörs rätt!";
        throw MySoapException("***Du har inte
Administrators rätt***");
    }
    else
    {
        try
        {
// skapas en reference till Nummerforamt för att lgara decimatal
enligt USA format till MYSQL
            NumberFormatInfo provider = new NumberFormatInfo( );

            provider.NumberDecimalSeparator = ".";

            provider.NumberGroupSeparator = ",";

            provider.NumberGroupSizes = new int[ ] { 3 };
            Thread.CurrentThread.CurrentCulture = new CultureInfo("en-
US");
            object temp1= Convert.ToDouble(ledningLon.ToString(),provider);
            object temp2= Convert.ToDouble(ansalldLon.ToString(),provider);
            object temp3= Convert.ToDouble(bbidrag.ToString(),provider);
            object temp4= Convert.ToDouble(sbidrag.ToString(),provider);
            object temp5= Convert.ToDouble(ovrig.ToString(),provider);
            double dledareLon= double.Parse(temp1.ToString());
            double danstalldLon= double.Parse(temp2.ToString());
            double dbostdbidrag= double.Parse(temp3.ToString());
            double dsocialbidrag= double.Parse(temp4.ToString());
            double dovrigkostnader= double.Parse(temp5.ToString());

            return mydatabas.setProjectInfo(String.Format("insert into euprojekt
values(null, '{0}', '{1}', '{2}', '{3}', '{4}', '{5}')" ,sProjekt,dledareLon,
danstalldLon,dbostdbidrag,dsocialbidrag,dovrigkostnader));

        }
        catch(System.Exception ex)
        {
            throw MySoapException("***kan inte uppdateras
DatabasServern!***");
        }
    }
}

/// <summary>
/// metoden för lgaraing Projekt Information efter projektslut och
///kontrollerar om användare
/// är admin detta sker gentemot en cache som håller sig i 280
second.
/// </summary>
/// <param name="dbidrag1">bostadbidrag till anställda efter
projektslut </param>
///<param name="dbidrag2">socialbidrag till anställda efter
projektslut</param>
///<param name="id">projekt Id är samma som för ett redan
sparat projekt i databasen</param>
```

Webbtjänst för projektledare

```
///<param name="sdata"> datumet när projekt slutar</param>
///<param name="sproject"> samma projektnamn som redan sparad i
databasen</param>
///<returns> en sträng för som hända med metod anropet</returns>

[WebMethod(Description="metoden för lagring av information efter
projektslut")]
public string insertIntoProjectFinsh(int id,string sproject,string
sdata,double dbidrag1,double dbidrag2)
{
    bool tempControllId=true;
    if(Context.Cache["Autentisering"]==null)
    {
        //return "Ange ditt AnvändareNamn och Lösenord";
        throw MySoapException("Ange ditt
AnvandareNamn och Losenord");
    }
    else if((bool)Context.Cache["Autentisering"]==
false)
    {
        //return "Du har inte Administratörs rätt!";
        throw MySoapException("***Du har inte
Administrators rätt***");
    }
    else
    {
        try
        {
            int[]intger=mydatabas.getProjectId("select
* from euprojekt_6;");
            foreach(int tempint in intger)
            {
                if(id==tempint)
                    tempControllId=false;
            }

            if(tempControllId)
            {
                // convertera double into USA format för att stöda MYSQL Databas

                NumberFormatInfo provider = new NumberFormatInfo( );

                provider.NumberDecimalSeparator = ".";

                provider.NumberGroupSeparator = ",";

                provider.NumberGroupSizes = new int[ ] { 3 };

                Thread.CurrentThread.CurrentCulture = new CultureInfo("en-
US");
                object temp1= Convert.ToDouble(dbidrag1.ToString(),provider);
                object temp2= Convert.ToDecimal(dbidrag2.ToString(),provider);
                double dtemp1=double.Parse(temp1.ToString());
                double dtemp2=double.Parse(temp2.ToString());
            }
        }
    }
}
```

Webbtjänst för projektledare

```
        return mydatabas.setProjectInfo(String.Format("insert into
euprojekt_6
values({0}, '{1}', '{2}', '{3}', '{4}')" , id, sproject, sdata, dtemp1, dtemp2))
;
    }
    else
        return "kan inte lagra data två gånger!";
    }
    catch(System.Exception ex)
    {
        throw MySoapException("***kan inte uppdateras
DatabasServern!***");
    }
}

/// <summary>
/// A method for SoapExceptions... useful for sending detailed data
/// </summary>
/// <param name="sError"></param>
/// <returns>en sträng av felmeddelande</returns>
public SoapException MySoapException(string sError)
{
    System.Xml.XmlDocument doc = new
    System.Xml.XmlDocument();

    System.Xml.XmlNode node =
    doc.CreateNode(XmlNodeType.Element,
    SoapException.DetailElementName.Name,
    SoapException.DetailElementName.Namespace);

    System.Xml.XmlNode child =
    doc.CreateNode(XmlNodeType.Element, "Error",
    "http://ds01.gotdns.org/projekt_WS/");
    child.InnerText = sError;

    node.AppendChild(child);

    //(sError, SoapException.ClientFaultCode, Context.Req
    uest.Url.AbsoluteUri, node)
    SoapException se = new
    SoapException(sError, SoapException.ClientFaultCode,
    Context.Request.Url.AbsoluteUri, node);
    return se;
}

/// <summary>
/// metoden hämtar projektsnamn som lagrad i databasen
/// </summary>
/// <returns> en sträng array av projektsnamn</returns>
[WebMethod(Description="metoden hämtar projektsnamn som lagrad i
databasen")]
public string[] getProjectName()
{
    try
```

Webbtjänst för projektledare

```
        {
            string[] myProjectName=
mydatabas.getProjectName();
            return myProjectName;
        }
catch(SoapException ex)
{
    throw MySoapException("***can not Reader
from Project table!***");
}
return null;
}

/// <summary>
/// hämtar projektkostnader som löner, bidrag och övrigakostnader
/// </summary>
/// <param name="projektid">projekt Id för valde projekt</param>
/// <returns>an array av double</returns>

[WebMethod(Description="metoden hämtar projektkostnader")]

public double[] getProjectCost(int projektid)
{
    if(Context.Cache["Autentisering"]==null)
    {
        //return "Ange ditt AnvändareNamn och Lösenord";
        throw MySoapException("Ange ditt
AnvandareNamn och Losenord");
    }
    else if((bool)Context.Cache["Autentisering"]==
false)
    {
        //return "Du har inte Administratörs rätt!";
        throw MySoapException("***Du har inte
Administrators rätt***");
    }
    else
    {
        try
        {
            return
mydatabas.getProjectCost(projektid);
        }
        catch(System.Exception ex)
        {
            throw MySoapException("***kan
inte uppdateras
DatabasServern!***");
        }
    }
}

/// <summary>
/// beräkna projektsbidragskostnader efter projektslut
/// </summary>
/// <param name="projektid">projekt Id för valde projekt</param>
```


Webbtjänst för projektledare

```
/// <returns>en array av double som innehåller projektbidrag efter
projektslut</returns>

[WebMethod(Description="metoden hämtar projektsbidrag")]

    public double[] getProjektsBidrag(int projektid)
    {
        if(Context.Cache["Autentisering"]==null)
        {
            //return "Ange ditt AnvändareNamn och Lösenord";
            throw MySoapException("Ange ditt
            AnvändareNamn och Losenord");
        }
        else if((bool)Context.Cache["Autentisering"]==
false)
        {
            //return "Du har inte Administratörs rätt!";
            throw MySoapException("***Du har inte
            Administrators rätt***");
        }
        else
        {
            try
            {
                return
                mydatabas.getProjectBidrag(projektid);
            }
            catch(System.Exception ex)
            {
                throw MySoapException("***kan inte
                uppdateras DatabasServern!***");
            }
        }
    }

    /// <summary>
    /// hämtar projekt primärnyckel
    /// </summary>
    ///<returns>en array av integar som innehåller all
    projektId</returns>

[WebMethod(Description="metoden hämta projektId som är lagrad")]
    public int[] getProjectId()
    {
        try
        {
            int[] myProjectId=
            mydatabas.getProjectId("select * from
            euprojekt;");
            return myProjectId;
        }
        catch(SoapException ex)
        {
            throw MySoapException("***can not Reader
            from Project table!***");
        }
    }
}
```

Webbtjänst för projektledare

```
                }
            }
        }
    }
}
```

2. Databas.cs

```
using System;
using System.Data.OleDb;
using Microsoft.Data.Odbc;
using System.Data;

using System.Collections;
using System.ComponentModel;

using System.Diagnostics;
using System.Web;
using System.Web.Services;

using System.Net;
using System.Web.Services.Protocols;
using System.Xml;
namespace projekt_WS
{
    /// <summary>
    /// klassen Databas hanterar kopplingen med databasen.
    /// skappas coonnectionstring mot databasen.
    /// </summary>
    public class Databas
    {
        string connectionString = @"Driver={MySQL ODBC 3.51
Driver};SERVER=ds01.gotdns.org;Database=tds01088;uid=tds01088;pwd=tds0
1088;";

        OdbcConnection co;

        public Databas()
        {
        }
    }
}
```

Webbtjänst för projektledare

```
///<summary>
///metoden hämtar användareNamn och lösenord från databasen
///</summary>
///<param name="sSql">en sql fråga</param>
///<returns>en dataset vilket inneåller användareanman och lösenord
</returns>
    public DataSet getNamePass(string sSql)
    {
        DataSet myDataSet=new DataSet();
        try
        {
            co= new OdbcConnection(connectionString);
                co.Open();

                string sSql1=sSql;

                OdbcDataAdapter myAdapter= new
                OdbcDataAdapter(sSql1,co);

                myAdapter.Fill(myDataSet, "Admin");

                co.Close();

        }
        catch(Microsoft.Data.Odbc.OdbcException ex)
        {

            throw ex;

        }
        return myDataSet;
    }

    /// <summary>
    /// metoden används för lagring av projekt kostnader och bidrag
    /// </summary>
    /// <param name="sSql1"> en sql fråga som används för
lagring</param>
    /// <returns>en sträng</returns>

    public string setProjectInfo(string sSql1)
    {
        OdbcCommand objcommand=new OdbcCommand();
        objcommand.CommandText=sSql1;
        try
        {
            co= new OdbcConnection(connectionString);
                objcommand.Connection=co;
                co.Open();
                objcommand.ExecuteNonQuery();
                co.Close();

        }
        catch(Microsoft.Data.Odbc.OdbcException ex)
        {
            throw ex;
        }
    }
}
```

Webbtjänst för projektledare

```
        }
        return "En rad har lagrad!";
    }
}

/// <summary>
/// meroden används för att hämta projektnamn från databasen
/// </summary>
/// <returns> en array som inneåller all projekt namn</returns>

public string[] getProjectName()
{
    try
    {
        int i =0;
        OdbcCommand objcommand=new OdbcCommand();
        OdbcDataReader objReader;
        objcommand.CommandText="select * from
        euprojekt;";

        co= new OdbcConnection(connectionString);
        objcommand.Connection=co;
        co.Open();

        objReader=objcommand.ExecuteReader();

        //läsa antal rader i tabeln!
        string [] myProjectNamn=new string[rowCount()];

        while(objReader.Read())
        {

            myProjectNamn[i]=objReader.GetValue(1).ToString
            ();

            i++;
        }
        co.Close();
    return myProjectNamn;

    }

    catch(Microsoft.Data.Odbc.OdbcException ex)
    {
        Service1 my=new Service1();
        throw my.MySoapException("kan inte nås
        MysqlServern");
    }

    return null;
}

/// <summary>
/// hämtat projekt primärnyckel från databasen
/// </summary>
/// <returns></returns>
public int[] getProjectId()
```

Webbtjänst för projektledare

```
{
    try
    {
        int i =0;
        OdbcCommand objcommand=new OdbcCommand();
        OdbcDataReader objReader;
        objcommand.CommandText="select * from
        euprojekt;";

        co= new OdbcConnection(connectionString);
        objcommand.Connection=co;
        co.Open();

        objReader=objcommand.ExecuteReader();

        //läsa antal radar i tabeln!

        int[] myProjectId= new int[rowCount()];

        while(objReader.Read())
        {

            myProjectId[i]=int.Parse(objReader.GetValue(0).ToString());

            i++;
        }
        co.Close();
        return myProjectId;

    }
    catch(Microsoft.Data.Odbc.OdbcException ex)
    {
        Service1 my=new Service1();
        throw my.MySoapException("kan inte nå
        MySqlServern");
    }

    return null;
}

/// <summary>
/// hämtat projekt primärnyckel från databasen
/// </summary>
/// <returns></returns>
public int[] getProjectId(string sSql)
{
    try
    {
        int i =0;
        OdbcCommand objcommand=new OdbcCommand();
        OdbcDataReader objReader;
        objcommand.CommandText=sSql;
        co= new OdbcConnection(connectionString);
        objcommand.Connection=co;
```

Webbtjänst för projektledare

```
        co.Open();
        objReader=objcommand.ExecuteReader();

        //läsa antal rader i tabeln!

        int[] myProjectId= new int[rowCount()];

        while(objReader.Read())
        {

            myProjectId[i]=int.Parse(objReader.Get
            tValue(0).ToString());
            i++;
        }
        co.Close();
        return myProjectId;

    }

    catch(Microsoft.Data.Odbc.OdbcException ex)
    {

        Servicel my=new Servicel();
        throw my.MySoapException("kan
        inte nås MysqlServern");

    }

    return null;

}

/// <summary>
/// genererar antal rader i projekt tabeln
/// </summary>
/// <returns> en integar av antal rader</returns>
public int rowCount()
{
    int antal=0;
    DataSet myDataSet=new DataSet();
    try
    {
        co= new OdbcConnection(connectionString);
        co.Open();
        string sSql1="Select * from euprojekt;";
        OdbcDataAdapter myAdapter= new
        OdbcDataAdapter(sSql1,co);
        myAdapter.Fill(myDataSet,"Admin");
        antal =myDataSet.Tables[0].Rows.Count;
        co.Close();

    }
    catch(Microsoft.Data.Odbc.OdbcException ex)
    {
        throw ex;

    }
    return antal;
}
```

Webbtjänst för projektledare

```
}

/// <summary
/// hämtar projektkostnader från databasen
/// </summary>
/// <param name="projektid"></param>
/// <returns>array av double som innehåller projektkostnader</returns>

public double[] getProjectCost(int projektid)
{
    try
    {
        double [] projektskostnad=new double[6];
        double ledningLon=0.0;
        double anstalldLon=0.0;
        double bostadbidrag=0.0;
        double socialbidrag=0.0;
        double ovrigakostnader=0.0;

        OdbcCommand objcommand=new OdbcCommand();
        OdbcDataReader objReader;
        objcommand.CommandText=String.Format("select * from
        euprojekt where projektId={0}", projektid);

        co= new OdbcConnection(connectionString);
                objcommand.Connection=co;
                co.Open();
        objReader=objcommand.ExecuteReader();

        if(objReader.Read())
        {

            ledningLon=double.Parse(objReader.GetValue(2).ToString());
            anstalldLon=double.Parse(objReader.GetValue(3).ToString());
            bostadbidrag=double.Parse(objReader.GetValue(4).ToString());
            socialbidrag=double.Parse(objReader.GetValue(5).ToString());
            ovrigakostnader=double.Parse(objReader.GetValue(6).ToString()
            );

            projektskostnad[0]=ledningLon;

            projektskostnad[1]=anstalldLon;

            projektskostnad[2]=bostadbidrag;

            projektskostnad[3]=socialbidrag;

            projektskostnad[5]=ovrigakostnader;
        }
        co.Close();
    }
    return projektskostnad;
}
```

Webbtjänst för projektledare

```
    }
    catch (Microsoft.Data.Odbc.OdbcException ex)
    {
        Servicel my=new Servicel();
        throw my.MySoapException("kan inte nås MysqlServern");
    }

    return null;
}

/// <summary>
/// hämtar bidrag summor efterprojektslut från databasen
/// </summary>
/// <param name="projektid"></param>
/// <returns> en array av double som innehåller bidrag
/// summor</returns>

public double[] getProjektBidrag(int projektid)
{
    try
    {
        double [] projektskostnad=new double[2];
        double bostadbidrag=0.0;
        ouble socialbidrag=0.0;

        OdbcCommand objcommand=new OdbcCommand();
        OdbcDataReader objReader;
        objcommand.CommandText=String.Format("select * from
        euprojekt_6 where projektId={0}", projektid);

        co= new OdbcConnection(connectionString);
        objcommand.Connection=co;
        co.Open();

        objReader=objcommand.ExecuteReader();

        if(objReader.Read())
        {
            bostadbidrag=double.Parse(objReader.GetValue(3).ToString());
            socialbidrag=double.Parse(objReader.GetValue(4).ToString());
            projektskostnad[0]=bostadbidrag;
            projektskostnad[1]=socialbidrag;
            return projektskostnad;
        }
        co.Close();

    }
    catch (Microsoft.Data.Odbc.OdbcException ex)
    {
        Servicel my=new Servicel();
        throw my.MySoapException("kan inte nås MysqlServern");
    }
}
```


Webbtjänst för projektledare

```
    }  
    return null;  
    }  
  }  
}
```

3. PasswordProvider.cs

```
namespace projekt_WS  
{  
    using System;  
    using Microsoft.Web.Services.Security;  
    using Microsoft.Web;  
    using System.Text;  
    using System.Security.Permissions;  
    using System.IO;  
    using System.Security.Cryptography;  
    using System.Data;  
  
    using System.Collections;  
    using System.ComponentModel;  
  
    using System.Diagnostics;  
    using System.Web;  
    using System.Web.Services;  
  
    using System.Net;  
    using System.Web.Services.Protocols;  
    using System.Xml;  
  
    /// <summary>  
    /// Summary description for Class1.  
    /// </summary>  
  
    [SecurityPermission(SecurityAction.Demand,  
        Flags= SecurityPermissionFlag.UnmanagedCode)]  
  
    public class PasswordProvider : IPasswordProvider  
    {
```

Webbtjänst för projektledare

```
        Databas mydatabas=new Databas();
        public string GetPassword(UsernameToken userName)
        {
            // Ensure the SOAP message sender passed a UsernameToken.
            if (userName == null)
                throw new ArgumentNullException();

            byte[] encodedUsername =
                System.Text.Encoding.UTF8.GetBytes(userName.Username);
            string username=
                System.Text.Encoding.UTF8.GetString(encodedUsername);
            try
            {
                DataSet myDataSet=mydatabas.getNamePass("select
                * from Admin");
                foreach(DataRow myRow in myDataSet.Tables[0].Rows)
                {
                    if( username==myRow[0].ToString())
                    {
                        return myRow[1].ToString();
                    }
                }
            }
            catch(System.Exception ex)
            {
                throw new Service1().MySoapException("fel
                uppstår i databasen");
            }
            return "";
        }
    }
}
```

Webbtjänst för projektledare

Bilag B klienters källkod

Adminstratörs application klient 1

1. Form1.cs [design]

Webbtjänst för projektledare



2. form1.cs

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Web;
using System.Xml;
using System.Web.Services.Protocols;
using System.Net;
using System.Web.Services;
using System.Diagnostics;
using System.Threading;
using System.Globalization;
using Microsoft.Web.Services;
using Microsoft.Web.Services.Security;
using System.Security.Cryptography;
using System.IO;
using System.Text;
using Microsoft.Web.Services.Security.X.509;

namespace admin
{
    /// <summary>
```

Webbtjänst för projektledare

```
/// Summary description for Form1.
/// </summary>
public class Form1 : System.Windows.Forms.Form
{

    Service1.Service1 proxy=new Service1.Service1();

    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.Container components = null;

    public Form1()
    {

        // Required for Windows Form Designer support

        InitializeComponent();

        // TODO: Add any constructor code after InitializeComponent call

    }

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    protected override void Dispose( bool disposing )
    {
        if( disposing )
        {
            if (components != null)
            {
                components.Dispose();
            }
        }
        base.Dispose( disposing );
    }

    #region Windows Form Designer generated code

    #endregion

    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.Run(new Form1());
    }
}
```

Webbtjänst för projektledare

```
    }

    private void button1_Click(object sender, System.EventArgs e)
    {
        addUserNamePass();
    }

    /// <summary>
    /// Send usernamn and password with Soap message
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void addUserNamePass()
    {
        string namn=this.textBox1.Text.ToString();
        string password=this.textBox2.Text.ToString();

        if(namn!="" && this.textBox2.Text.ToString()!="")
        {
            //Creat UsrenameToken with usernamn and password
            UsernameToken userToken;
            userToken = new UsernameToken(namn,password,
                PasswordOption.SendHashed);
            try
            {
                Service1.Service1 proxy=new Service1.Service1();

                // Gets the SoapContext associated with the SOAP request.
                SoapContext requestContext =
                    proxy.RequestSoapContext;
                // Sets the TTL to 1 minute.
                requestContext.Timestamp.Ttl = 60000;

                // Adds the token to the SOAP header.
                requestContext.Security.Tokens.Add(userToken);
                // Signs the SOAP message using the UsernameToken.

                requestContext.Security.Elements.Add(new
                Signature(userToken));

                // compare if the urname and password is valid
                if(proxy.validUser())
                {
                    string[] myarray=proxy.getProjectName();

                    this.comboBox1.Items.Clear();

                    this.comboBox2.Items.Clear();

                    foreach(string instring in myarray)
                    {

                        this.comboBox1.Items.Add(instring);

                        this.comboBox2.Items.Add(instring);
                    }
                }
            }
        }
    }
}
```

Webbtjänst för projektledare

```
    }

    MessageBox.Show("valid UserName and password");
}

else
{
    MessageBox.Show("not valid password!");
}
}
catch (System.Web.Services.Protocols.SoapException se)
{
    MessageBox.Show(se.ToString());
}
catch (Exception ex)
{
    Error("Exception caught while invoking an XML Web service.",
        ex.ToString());
    return;
}
}

else
MessageBox.Show("skriv ditt AnvändareNamn och Lösenord!");
}
```

```
    /// <summary>
    /// Check the valid typ of the password
    /// </summary>
    /// <param name="password">password</param>
    /// <returns>bool data typ</returns>

private bool validDigits(string password)
{
    foreach(char c in password)
    {
        if(char.IsLetter(c))
        {
            return false;
        }
    }
    return true;
}

/// <summary>
```

Webbtjänst för projektledare

```
/// metoden anropas när systemadministratör vill spara kostnader till
ett projekt
/// </summary>
private void InserIntoDatabas(){

    if(this.textBox3.Text==" " || this.textBox6.Text==" " || this.textBox7.Text==" " || this.textBox8.Text==" " || this.textBox9.Text==" " || this.textBox10.Text==" ")
    {
        ErrorHandler("fylla all Information!");
    }
    else
    {

        try
        {
            string
projektNamn=this.textBox3.Text;
            if(
validDigits(this.textBox6.Text.ToString())&&
validDigits(this.textBox7.Text.ToString())&&
validDigits(this.textBox8.Text.ToString())&&validDigits(this.textBox9.Text.ToString())&& validDigits(this.textBox10.Text.ToString()))
            {
double projektLedarealon=double.Parse(this.textBox6.Text.ToString());
double anstaldeLon= double.Parse(this.textBox7.Text.ToString());
double bostadBidrag= double.Parse(this.textBox8.Text.ToString());
double socialBidrag=double.Parse(this.textBox9.Text.ToString());
double ovriga = double.Parse(this.textBox10.Text.ToString());

                MessageBox.Show(proxy.insertIntoProject(projektNamn,projektLedarealon,anstaldeLon,bostadBidrag,socialBidrag,ovriga));
                string[] myarray=proxy.getProjectName();

                this.comboBox1.Items.Clear();

                this.comboBox2.Items.Clear();
                foreach(string instring in myarray)
                {

                    this.comboBox1.Items.Add(instring);

                    this.comboBox2.Items.Add(instring);

                }
            }
            else
            {
                ErrorHandler("Ange DecimalTal");
            }
        }
    }
}
catch(System.Web.Services.Protocols.SoapException ex)
{
    MessageBox.Show(ex.ToString());
}
}
```


Webbtjänst för projektledare

```
private void button3_Click(object sender, System.EventArgs e)
{
    setProjektInt();
}

/// <summary>
/// uppdaterar projekt listan
/// </summary>

public void setprojektInt()
{
    int indexId= this.comboBox2.SelectedIndex;
    if(indexId!=-1)
    {
        string
        myprojectnamn=this.comboBox2.SelectedItem.ToStr
ing();
        try
        {
            int [] myintarray=proxy.getProjectId();
            int projectId=myintarray[indexId];
            this.label20.Text=projectId.ToString();

            if(validDigits(this.textBox4.Text.ToString())||validDigits(th
is.textBox5.ToString()))
            {
                // skickas parametrar till webbtjänstsmetoden.
                float bid=float.Parse(this.textBox4.Text.ToString());
                double bidrag1=double.Parse(this.textBox4.Text.ToString());
                double bidrag=double.Parse(this.textBox5.Text.ToString());

                System.DateTime mydate= System.DateTime.Now.Date;
                char spliter=' ';
                string[] Naps=mydate.ToString().Split(spliter);
                string myCurrentDate=Naps[0].ToString();

                MessageBox.Show(proxy.insertIntoProjectFinsh(projectId,myproj
ectnamn,myCurrentDate,bidrag1,bidrag).ToString());

            }
            else

                ErrorHandel("Ange heltal eller decimalTal!")
        }
    }
}
catch(System.Exception ex)
{
    ErrorHandel(ex.ToString());
}
}
else
```

Webbtjänst för projektledare

```
{
    MessageBox.Show("Ange ett projektNamn!");
}

/// <summary>
/// Gets error messages
/// </summary>
/// <param name="sSql"></param>

private void ErrorHandlerel(string sSql)
{
    MessageBox.Show(sSql.ToString());
}

private void button2_Click(object sender, System.EventArgs e)
{
    Inserintodatabas();
}

/// <summary>
/// Kalkylerar kosynader och Intäkter
/// </summary>

private void button4_Click(object sender, System.EventArgs e)
{
    calc();
}

public void calc()
{
    this.richTextBox1.Text="";
    if(this.comboBox1.SelectedIndex==-1)
    {
        ErrorHandlerel("Välja ett projekt!");
    }
    else
    {
        try
        {
            int indexId= this.comboBox1.SelectedIndex;
            int [] myintarray=proxy.getProjectId();
            int projectId=myintarray[indexId];
            double[]
                kostnader=proxy.getProjectCost(projectId);

            double[]
                bidragReducering=proxy.getProjectBidrag(project
                Id);
            if(bidragReducering !=null)
            {
                double anstalldAvgifter=kostnader[1]*0.31;
                double anstalldSkatt =kostnader[1]*0.25;
                double bostadbidragEPro=kostnader[2]-
                bidragReducering[0];
            }
        }
    }
}
}
```

Webbtjänst för projektledare

```
double socilbidragEPro= kostnader[3]-
bidragReducering[1];

double totalKosnader= kostnader[0]+
kostnader[1]+kostnader[5];
double
minskingskosnader=bostadbidragEPro+socilbidragEPro
+anstalldAvgifter+anstalldSkatt;
double currentcost=totalKosnader-
minskingskosnader;
double
betalningår=bostadbidragEPro+socilbidragEPro;

double
rantaOchSuma=currentcost+((currentcost/100)*0.04);

double monder=betalningår/12;
double kvarAttbetal=rantaOchSuma -betalningår;
double rantaEfterBetalning=kvarAttbetal/monder;
double antalMonader=rantaEfterBetalning+12;

this.richTextBox1.AppendText(@"Den totala
kostnader för projektet är : " + "-----"
+totalKosnader      +" Mkr \n"

      +" 1. löner till
prjektledning:"+kostnader[0] +" \n 2. Löner till
anställda :"+ kostnader[1]+" \n 3. övriga
kostnader för projektet:"+ kostnader[5]+" \n\n"

      +"Minskning av Kostnader är följande\n"

      +"Bostadbidragsreducering      :
+"-----" + bostadbidragEPro +" Mkr\n"

      +"Socialbidragsreducering      :
+"-----" +socilbidragEPro      +" Mkr\n"

      +"AnställdeSkatt i projekt löpt (/0,25): "
+"-----"+ anstalldSkatt+" Mkr\n"

      +"ScocialAvgifter i projekt löpt(/0,31): "
+"-----"+ anstalldAvgifter+" Mkr\n"

      + "Kostnaderna minskamas
med: (" +bostadbidragEPro+" ; "+socilbidragEPro+" ;
"+anstalldAvgifter+" ; "+ anstalldSkatt

      +" )\n\n"+

      " Vi har räknat 4% ränta på summor\n\n"+

      "Totalprojektskostnader minus minskning av
bidrag och sammhälletsvinst är :"+ currentcost+"
Mkr\n"

      + "enligt scenario 2 [14] betalning för
ett år är" +" "+ betalningår +" \n"
```

Webbtjänst för projektledare

```
        + "betalning för en månad är "+ " "+
monder +"Mkr\n"

        + "kvar att betala efter ett år är"+ " "+
kvarAttbetal +"Mkr\n"

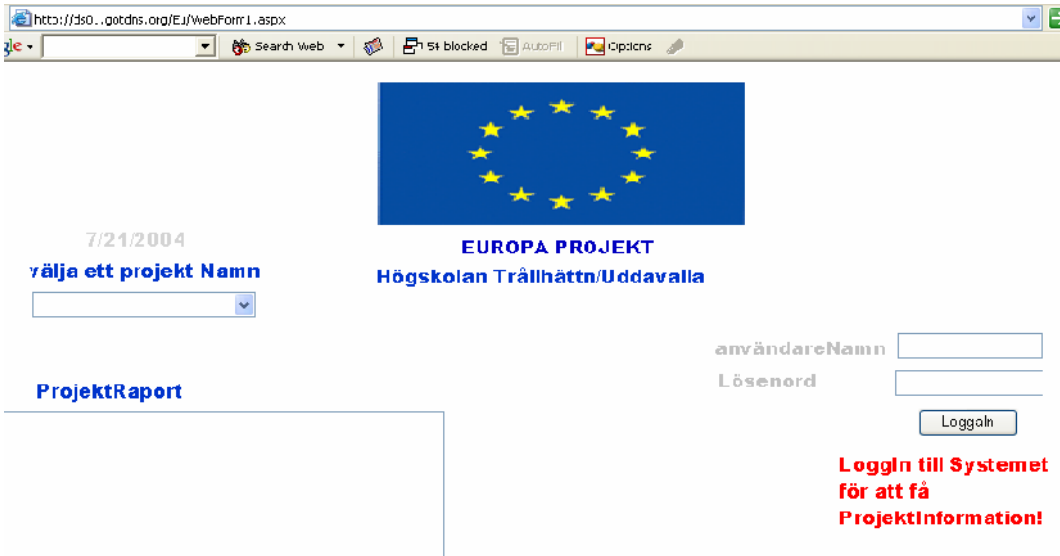
        +"Antal månader för betalning av
projektKostnader är " +" "+ antalMonader+"Mån");
    }

else
{
    this.richTextBox1.AppendText("projektet har inte slutat!");
}
}
catch(SoapException ex)
{
    ErrorHandel(ex.ToString());
}
}
}

public void Error(string sfel ,string ex)
{
    MessageBox.Show(sfel+ ex);
}
}
}
```

Webbtjänst för projektledare

Webb applikation 3.WebForm1.aspx[design]



http://ds0..gotdns.org/EJ/WebForm1.aspx

7/21/2004

välja ett projekt Namn

EUROPA PROJEKT
Högskolan Trållhätt/Uddavalla

ProjektRaport

användareNamn

Lösenord

LoggaIn

**LoggaIn till Systemet
för att få
ProjektInformation!**

4. WebForm1.aspx.cs

För källkoden se bilag B 2