

SCA

- Smart Contact Access



**LUNDS
UNIVERSITET**

Lunds Tekniska Högskola

**LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds Tekniska Högskola, LTH**

Examensarbete:
Atdhe Grajqevci

© Copyright Atdhe Grajqevci

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Media-Tryck
Biblioteksdirektionen
Lunds universitet
Lund <2008>

Sammanfattning

SCA – Smart Contact Access

Målet med projektet är att utveckla ett system där det skall gå att komma åt information från en intern databas på ett enkelt sätt med en mobil enhet.

Eftersom varje enskild databas är unik så skall det gå att justera systemet enkelt genom ett administrationsgränssnitt så att det passar till varje databas.

Informationen är oftast känslig och skall inte åtkommas av obehöriga användare, så den mobila enheten behöver autentiseras för att skydda informationen.

Företagets acceptanskrav för detta systemet är att alla applikationerna skall vara fungerande samt att webbtjänstens säkerhet är tillräckligt hög att inga obehöriga kan komma åt information från företagets intranät.

Systemet kommer att bestå av ett par applikationer:

1. Webbtjänst som skall kommunicera med databasen och skicka information till webb- och den tjocka klienten.
2. Webbaserad klient, skall kunna visa data från databasen.
3. Tjock klient, skall kunna visa samt uppdatera data från databasen.
4. Administrationsgränssnitt som skall kunna konfigurera webbtjänsten.

Den slutgiltiga produkten av webbklienten och den tjocka klienten visas i figur 1 resp. figur 2. För autentisering av användaren används telefon ID:et och sköts automatiskt av den tjocka klienten. Med webbklienten skriver man in ett ID i adresslänken för autentisering. Klienterna innehåller två olika sökfunktioner. Från varje träff från sökningen kan ytterligare information fås fram genom ”Visa detaljer” funktionen på klienterna. Med den tjocka klienten kan man även uppdatera kontaktpersonernas information, detta går inte med webbklienten.

Företaget blev nöjd med den slutgiltiga produkten vilket kommer att utvecklas vidare av företaget och även säljas.



Figur 1



Figur 2

Nyckelord: Webbtjänst, .net framework, mobilapplikation, C#, compact framework

Abstract

SCA – Smart Contact Access

The goal with this project is to develop a system that makes it possible to access information from a local database on an easy and convenient way with a mobile device. Because every database is unique, this system needs to be configurable with an administration interface so that it will fit all databases.

This kind of information is most likely confidential of some sort, this system needs to be protected against unauthorized people and because of this. The mobile device needs to be authenticated before being usable.

The company's main requirement for this system is that all applications needs to be working and that the web service's security is high enough that unauthorized people can't access any important information from the company's intranet.

The system will consist of several applications:

1. A web service that will communicate with the database and send information to the web and fat client.
2. A web based client, able to display the database information on the screen.
3. A fat client, able to display and update the database information.
4. An administration interface, able to configure the web service and the database queries used.

The completed version of the web client is shown in Image 1 and the fat client is shown in Image 2. To authenticate the user, the device ID is checked and is done automatically by the fat client. For the web client, an ID is typed in the address link for authentication. Both clients have two different search functions. More information about every search result can be displayed by clicking the "Visa detaljer" link under the result name. With the fat client the user are even able to update a contact person's information, this is not available on the web client.

The company was satisfied with the final version of the product and is going to be developed further and even be sold to customers.



Image



Image 2

Keywords: Web service, .net framework, mobile application, C#, compact framework

Förord

Jag vill ge IXX IT-Partner ett särskilt tack som har gett mig erfarenheter inom arbetslivet och denna arbetsuppgift samt utvecklarna Marcus Sarkar, Carl-Oscar Ejwertz, Martin Sjögren och Per Resell för all hjälp de har gett mig under utvecklandets gång. Samt ett tack till alla andra personer från IXX samt Monika Christensson och Mattias Christensson.

Jag vill också tacka Lise Jensen från Campus Helsingborg för all rådgivning samt hjälp jag har fått från henne.

Innehållsförteckning

1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Målsättning	1
2 Problemformulering	2
2.1 Delproblem	2
2.1.1 Intern databas	2
2.1.2 Unika databaser	2
2.1.3 Klientautentisering	2
3 Avgränsningar	3
4 Huvudkrav	3
5 Säkerhet inom webbtjänst	4
5.1 Möjliga lösningar	5
5.2 Analys av lösningar	6
5.3 För- och Nackdelar	8
5.4 Konklusion av analys	8
6 Metodbeskrivning	9
6.1 Kravspecifikation	10
6.2 Samla information om säkerhet inom webbtjänst	10
6.3 Webbtjänst	10
6.4 Administrationsgränssnitt	10
6.5 Webbklient	11
6.6 Tjock klient	11
6.7 Uppdatera kravspecifikation	11
7 Resultat	12
7.1 Webbtjänst	15
7.2 Administrationsgränssnitt	16
7.3 Webbklient	20
7.4 Tjock klient	23
7.5 Sammanfattning	26
7.6 Framtida arbete	26
8 Generella referenser	27
9 Terminologi	28
10 Bilagor	30
10.1 Kravspecifikation	30

1 Inledning

IXX IT-Partner AB är ett företag som utvecklar och integrerar IT-system, konsultverksamhet, drift och nätverksövervakar, utbildning samt programvarutjänster. Företagets syfte med detta projekt är att undersöka om det går att skapa ett system med tillräckligt hög säkerhet så att databas samt intranät information inte kan avläsas, sedan skapa ett sådant system.

1.1 Bakgrund

Jag skall skapa ett system som skall kommunicera med en databas genom en brandvägg. Systemet kommer att bestå av ett par applikationer:

5. Webbtjänst som skall kommunicera med databasen och skicka information till webb- och den tjocka klienten.
6. Webbaserad klient, skall kunna visa data från databasen.
7. Tjock klient, skall kunna visa samt uppdatera data från databasen.
8. Administrationsgränssnitt som skall kunna konfigurera webbtjänsten.

De två klienterna skall vara anpassade till mobiltelefoner. Den webbaserade klienten skall vara kompatibel med de flesta mobilwebbläsare.

1.2 Syfte

Denna rapport kommer att beskriva vilka problem jag fick arbeta med, hur jag gick till väga för att lösa dem och slutligen vilka resultat jag fick ut i slutändan.

Examensarbetets syfte är jag skall tillämpa mina kunskaper och färdigheter jag har förvärvat under studietiden. Efter att examensjobbet är klart så skall jag föra en presentation för företaget samt skolan. Jag skall dessutom kritiskt granska och diskutera ett annat examensjobb.

1.3 Målsättning

Företagets mål med detta projekt var att undersöka om det gick att utveckla ett system som kunde användas globalt och inte endast internt. Det var först och främst tänkt att använda systemet som kunddatabas.

Jag vill lösa problemet på bästa möjliga sett med mina kunskaper inom programmering och säkerhet. Dessutom utveckla systemet färdigt inom tidsgränsen med så lite avgränsningar som möjligt.

Efter utvecklingen är klar så skall jag göra en presentation av systemet samt förhoppningsvis ha gjort ett gott intryck för IXX IT-Partner AB.

2 Problemformulering

Det skall gå att komma åt information från en intern databas på ett enkelt sätt med en mobil enhet. Eftersom varje enskild databas är unik så skall det gå att justera systemet enkelt genom ett administrationsgränssnitt så att det passar till varje databas.

Informationen är oftast känslig och skall inte åtkommas av obehöriga användare, så den mobila enheten behöver autentiseras för att skydda informationen.

Företagets acceptanskrav på systemet är att alla applikationerna skall vara fungerande samt att säkerheten på webbtjänsten är tillräckligt hög att ingen obehörig skall komma åt någon information som kan vara till skada för företaget nät eller databas.

2.1 Delproblem

Följande avsnitt handlar om uppdelning av problemet samt ett par mindre problem med en liten kort förklaring över varje problem.

2.1.1 Intern databas

Ett internt nät skyddas av en brandvägg och en direkt kommunikation från databasen till en klient är riskabelt eftersom en klient som ligger tillgängligt på Internet är inte säker nog att hantera hemlig information som databas anslutning samt tabell- och kolumnnamn.

2.1.2 Unika databaser

Eftersom detta system är menat att användas av flera olika nät, så skall systemet vara så dynamisk som möjligt. Det vill säga att det skall vara lätt att konfigurera vilken databas som skall anslutas samt vilka tabeller och kolumner som skall skrivas ut på klienterna.

2.1.3 Klientautentisering

Informationen som skall visas på klienterna är oftast känslig information som inte skall vara tillgänglig allmänt. Ett exempel på information är ett företags kundinformation och det skulle vara mindre bra ifall en obehörig fick tag på den informationen. Alltså måste enheten som använder klienterna autentiseras så att bara behöriga personer kan få tillgång. Ett login system är inte acceptabelt från företaget eftersom det ska gå snabbt att använda klienterna vid upprepad användning.

3 Avgränsningar

Jag kommer att prioritera utveckling av själva systemet för företaget. Försöka implementera så många funktioner som möjligt. Eftersom jag prioriterar utveckling så kommer jag begränsa undersökning av säkerhet inom webbtjänst till de få lösningarna som jag beskriver i kapitel 5.1.

4 Huvudkrav

Av problemformuleringen så framstod de mest grundläggande kraven. Dessa krav satte grunden till systemet och efteråt så tillkom en del ytterligare krav efter intervju med företagets utvecklare samt undersökning på liknande system.

- *Webbklienten skall kunna visa information från en databas.*
- *Tjocka klienten skall kunna visa information från en databas.*
- *Tjocka klienten skall kunna uppdatera information i en databas.*

Krav som beskriver huvuduppgiften med själva systemet.

- *Webbtjänsten skall ha metoder som kommunicerar med databasen.*

Eftersom den interna databasen inte kan kommunicera direkt med klienterna så finns det en webbtjänst som tar hand om kommunikationen med databasen.

- *Webbtjänsten skall vara skyddad mot obehöriga.*
- *All databas information skall vara skyddad mot obehöriga.*

Säkerhetskrav som är viktiga eftersom det ökar risken för intrång om webbtjänsten samt databas informationen läcker ut.

- *Ett administrationsgränssnitt för webbtjänsten skall kunna konfigurera databas frågor (queries).*
- *Med administrationsgränssnitt skall det gå att administrera vilka telefoner som får använda klienterna.*

Genom att använda sig av ett gränssnitt så blir det lättare att anpassa systemet för alla databaser. Access kontrollen görs även med detta gränssnitt.

- *Endast registrerade användare får använda klienterna.*
- *Klienterna skall autentiseras vid användning.*

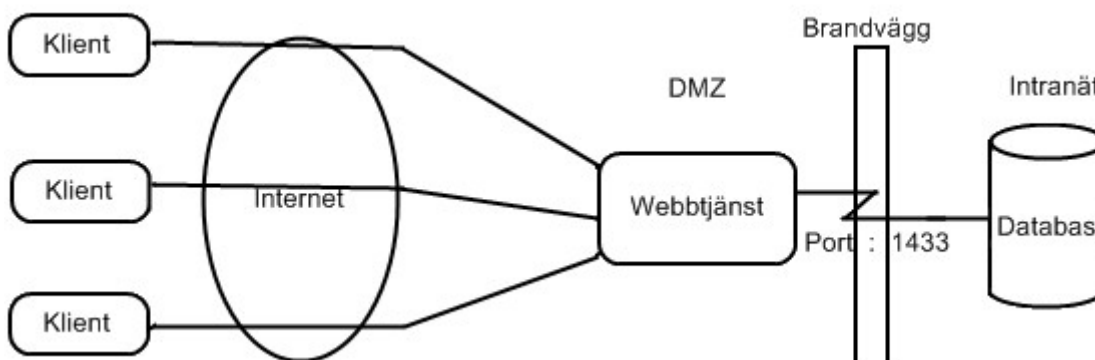
För att skydda den information som visas på klienterna från obehöriga personer.

5 Säkerhet inom webbtjänst

Man kan kort säga att en webbtjänst är samma sak till datorprogram som webbsidor är till människor. Med hjälp av en sådan kan ett program från olika ställen få lätt tillgång till delad information. Innan webbtjänster hade tillräcklig hög säkerhet så var det problem att förvara känslig information som sjukhus patienters journaler etc, på det säkerhetsriskerade Internet. Men nu så vågar fler företag bygga ut sina system för fler användare eftersom säkerheten har utvecklats mycket vilket ger dem till exempel ett mycket bättre förhållande med kunder.

Webbtjänster använder sig av XML för att kommunicera med olika system. XML är ett språk som är läsligt för människor men för att de olika systemen skall kunna förstå ett XML dokument, behövs ett schema som förklarar hur dokumentet skall läsas.

Men problemet med att hantera känslig information är ju fortfarande det samma. En webbtjänst måste ha tillräcklig hög säkerhet så att obehöriga personer inte kan komma åt informationen.



Figur 5.1

Denna webbtjänst som jag kommer utveckla skall alltså vara installerad på en server som kommer att kommunicera med ett företags intranät. Detta intranät skyddas av en brandvägg och all kommunikation som går mellan klienterna och webbtjänsten kommer av standard vara skyddad med SSL kryptering, förkortning av *Secure Sockets Layer* som är en säkerhetsmekanism som uppfanns av Netscape och används för att kryptera kommunikationen mellan två värdar. Ofta är ena värden en webbserver (ex. onlinebutik) och den andra är en webbläsare (ex. Internet Explorer). Tanken med att kryptera data som sänds till och från dessa två värdar är att ingen annan användare på nätverket (oftast internet) ska kunna avlyssna eller manipulera data.

Webbtjänsten kommer att ligga på DMZ, förkortning av *DeMilitarized Zone*, "demilitariserad zon", ett vanligt förekommande arrangemang för datorer som måste vara mer eller mindre direkt anslutna till Internet för att fylla sin

funktion och därför exponeras mot omvärlden. Maskiner som placerats i ett DMZ måste anses vara utsatta för alla typer av attacker och ska därför inte innehålla någon funktion som inte absolut behövs för uppgiften eftersom varje extra funktion innebär ökad sårbarhet.

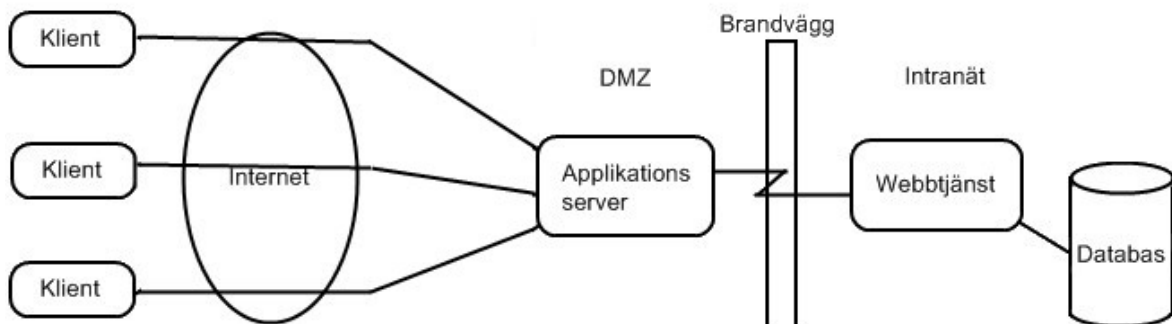
All information mellan databas och själva webbtjänsten går genom port 1433 som är standardport för SQL databas. Figur 5.1 visar detta. Denna port är inte krypterad vilket gör att om en obehörig får tillgång till webbtjänsten, vilket kan vara väldigt vanligt eftersom den är exponerad mot Internet, denne obehörige kommer alltså att ha tillgång till allting inne i företagets intranät.

Detta problem kommer jag att försöka analysera och hitta en lösning till.

5.1 Möjliga lösningar

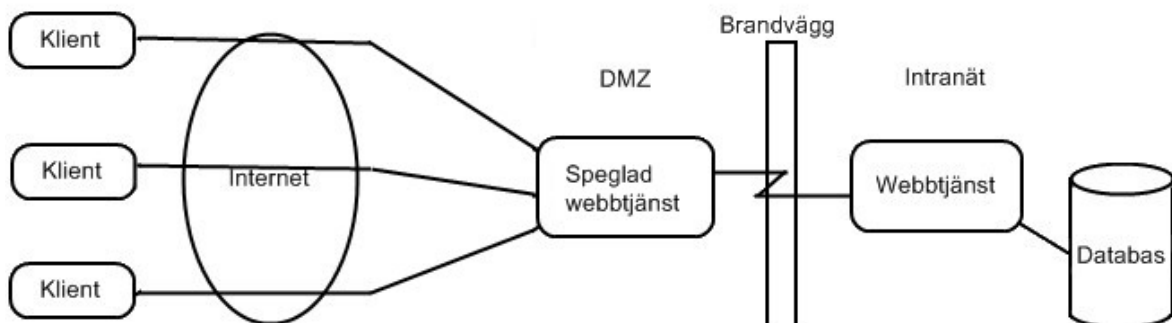
Jag har tre lösningar till problemet som jag analyserar. De följande tre lösningarna:

- A) Mellan webbtjänst och klienterna kan man använda sig av en applikationsserver som tar hand om trafiken i mellan. Figur 5.1.1 beskriver hur uppbyggnaden kommer att se ut.



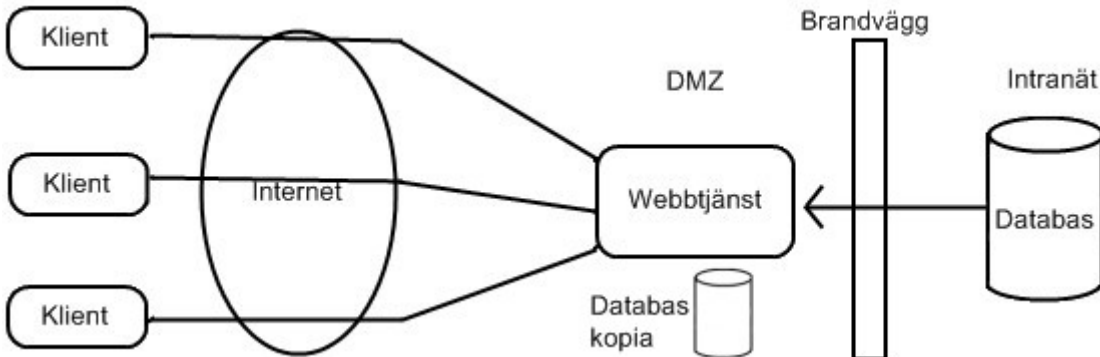
Figur 5.1.1

- B) En annan lösning som liknar lösning A) är att istället för att använda sig av en applikationsserver så använder man en ytterligare webbtjänst som använder endast metoderna som klienterna behöver. Figur 5.1.2 visar kortfattat denna lösning.



Figur 5.1.2

- C) Istället för att webbtjänsten kommunicerar med databasen, så kan man ha en databas som kopierar över det data som behövs till webbtjänsten med jämna mellanrum eller vid ändring av data i databasen.

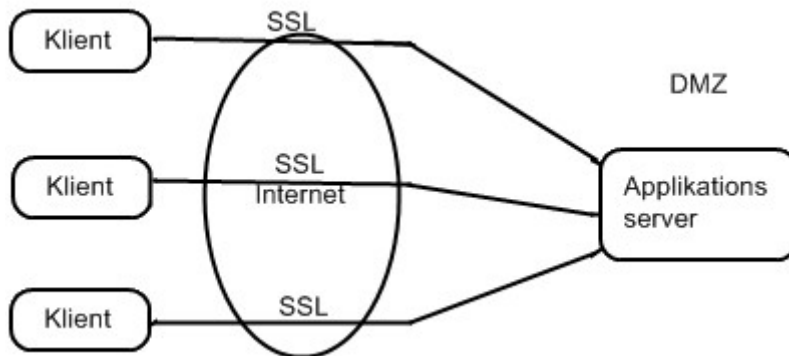


Figur 5.1.3

5.2 Analys av lösningar

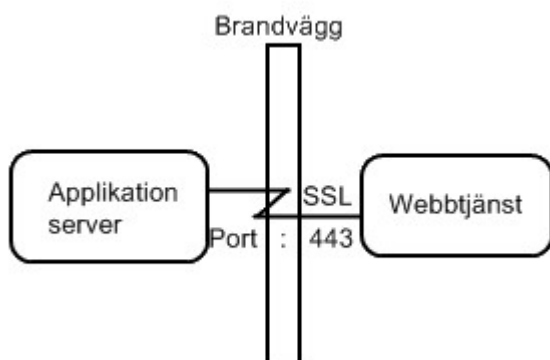
Det som gör att dessa lösningar hjälper är att inget mer än nödvändigt kommer exponeras igenom brandväggen.

- A) Kommunikationen mellan klienterna och applikationsservern är krypterad med SSL kryptering vilket skyddar informationen som skickas mellan systemen, gör det oläsligt för obehöriga samt detektering vid ändring av information.



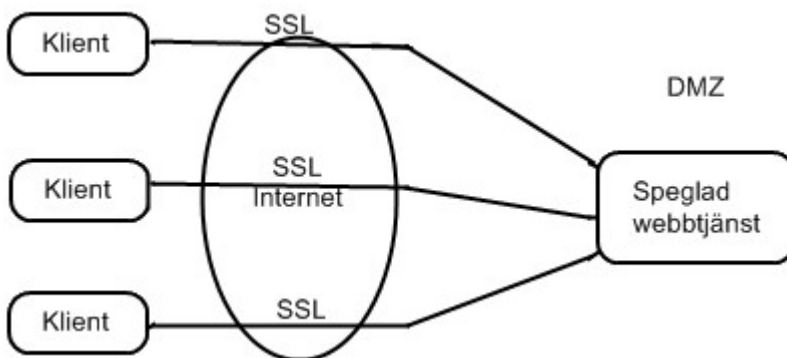
Figur 5.2.1

Även kommunikationen mellan applikationsservern och webbtjänsten är krypterad med SSL. SSL använder sig av en standardport på brandväggen som är Port 443.



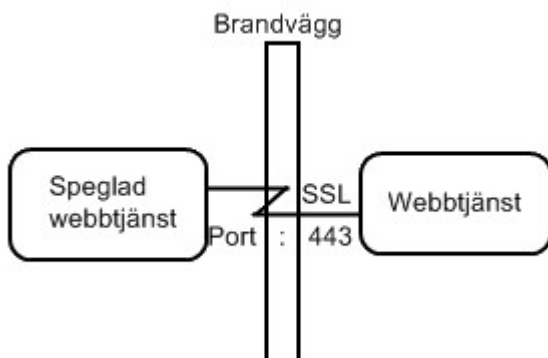
Figur 5.2.2

B) Samma som lösning A) så kommer kommunikationen mellan klienterna och den speglade webbtjänsten vara krypterad med SSL.



Figur 5.2.3

Denna lösning är i stort sätt samma som A) fast med en speglad webbtjänst istället för en applikationsserver. Kommunikationen mellan speglad webbtjänst och webbtjänsten i intranätet går också igenom Port 443.



Figur 5.2.4

C) Om det inte går någon kommunikation in mot intranätet, så kan inte obehöriga komma åt någon information som kan skada nätet. Ingen Port kommer att vara öppen in mot intranätet men då måste databasen själv uppdatera den kopia som bevaras tillsammans med webbtjänsten vid varje ändring eller vid bestämda intervaller.

Problemet med denna lösning är ju att databas information kommer publiceras ut i DMZ vilket är inte bra om det är känslig information som används. Dessutom så skall det gå att uppdatera databas information vilket kommer att bli besvärligt om webbtjänsten inte kan kommunicera med den riktiga databasen.

5.3 För- och Nackdelar

- A) En fördel med denna lösning är att även om den en obehörig skulle komma åt applikationsservern så kommer denne inte kunna göra mycket mer som skulle vara till skada till företagets nät. En nackdel är att det är för det mesta besvärande att hantera en applikationsserver samt att sätta upp en. Det finns även en risk att en obehörig kan komma åt webbtjänsten i det inre nätet men den risken är inte särskilt stor.
- B) Fördelen med denna lösning är ungefär samma som A), även om en obehörig skulle komma åt den speglade webbtjänsten så kommer denne inte kunna göra någon skada för nätet. En annan fördel är att det är mycket enklare att använda sig av en speglad webbtjänst än applikationsservern. Nackdelen är den samma som A), det finns fortfarande en liten risk att en obehörig kommer åt den inre webbtjänsten.
- C) En stor fördel med denna lösning är att det finns ingen risk överhuvudtaget att en obehörig kommer åt det inre nätet eller databasen. Men nackdelen är ju att den databas information som skall användas till klienterna kommer vara lätt tillgänglig.

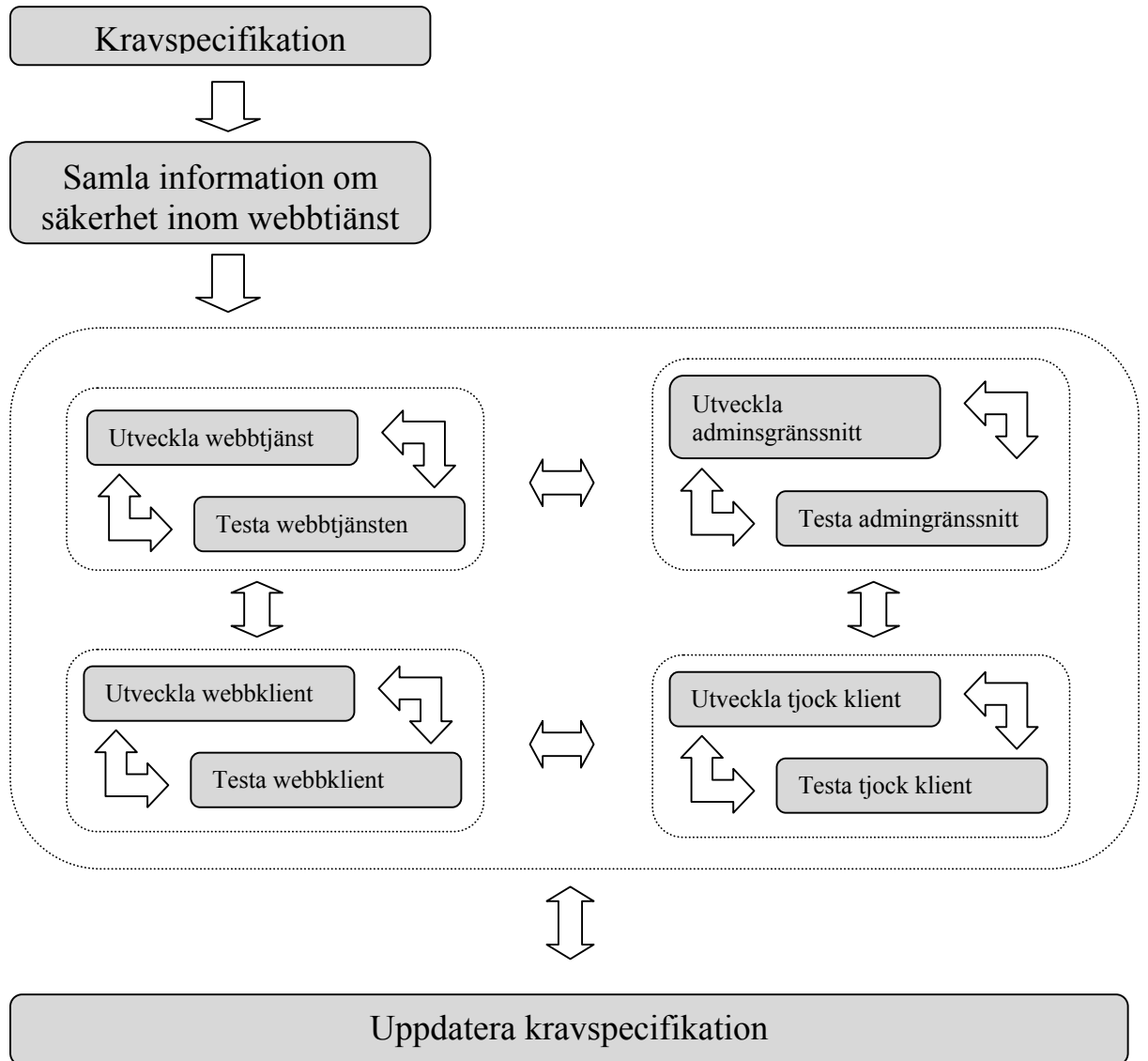
5.4 Konklusion av analys

Lösning C) kommer inte att fungera med detta system eftersom den information som kommer användas kan variera beroende på vad administratören väljer och då kan känslig information väljas och det vill företag inte exponera på DMZ. Denna lösning kommer jag alltså att utesluta på grund av det.

Lösning A) och B) fungerar båda med mitt system och de löser det faktiska problemet. Men eftersom användning av applikationsservrar är mer besvärligt än att bara skapa en speglad webbtjänst, så har jag bestämt mig för att basera min lösning på B).

6 Metodbeskrivning

Figur 6.1 beskriver kort hur jag har tänkt gå till väga med detta projekt. En mer noggrann beskrivning av varje moment kan läsas efter bilden.



Figur 6.1

6.1 Kravspecifikation

Här så börjar jag skriva en kravspecifikation för systemet jag skall utveckla. Genom att använda mig av eliciteringstekniker som interview, similar companies samt brainstorming kan jag få ut de mest grundläggande kraven som jag beskriver under kapitel 4.

6.2 Samla information om säkerhet inom webbtjänst

Denna uppgift har jag fått i uppdrag från företaget, att undersöka säkerhetsmöjligheterna inom webbtjänst. Jag skall begränsa undersökningen till ett par av de vanligaste lösningarna till problemet som är beskrivna under kapitel 5.

6.3 Webbtjänst

Det första jag skall utveckla är webbtjänsten som skall fungera som centrum för systemet. Den tar hand om all databas kommunikation och skickar vidare all information till klienterna. Även kontroll av access till klienterna samt alla tillåtna telefoner och ID sköts och sparas tillsammans med webbtjänsten. Alla klienterna, webbklient, tjock klient samt administrationsgränssnitt, skall ha en referens till denna webbtjänst.

Under utvecklandets gång skall testning pågå parallellt.

6.4 Administrationsgränssnitt

Administrationsgränssnittet skall ha funktioner att ändra alla databas frågor som används av webbtjänsten samt skall kunna konfigurera access kontrollen. Denna klient skall ligga inom det lokala nätet och inte vara tillgängligt från internet av säkerhetsskäl. Detta gränssnitt kommer jag att utveckla efterhand när idéer ändras och när webbtjänsten eller klienterna behöver ändras.

Testning sker parallellt med utveckling.

6.5 Webbclient

En klient som skall kunna visa information från en databas genom att fråga webbtjänsten för informationen. Användaren kan söka efter ett ord som skickas till webbtjänsten som skickar vidare en fråga till databasen. Databasen returnerar alla träffar till webbtjänsten som skickar det vidare till webbklienten, därefter modifieras informationen av webbklient och skriver ut den läsligt. Det finns två olika sökningar vars funktioner kan ändras med administrationsgränssnittet.

Testning sker parallellt med utveckling.

6.6 Tjock klient

En klient till mobila enheter som skall kunna visa information från en databas genom webbtjänsten, precis som webbklienten. Men denna klient skall även kunna uppdatera den visade informationen. Precis som webbklienten, fungerar den tjocka klienten på samma sätt. Använder sig av samma webbtjänst funktioner för att få tag på information från databasen. Även på denna klient finns det två olika sökningar.

Samma som de andra klienterna kommer testning ske parallellt med utveckling.

6.7 Uppdatera kravspecifikation

Under utvecklandet av webbtjänst och klienter så kommer en del egna krav tillkomma och dessa skall då läggas till i kravspecifikationen. Detta sker under utvecklingen när dessa krav uppkommer.

7 Resultat

I slutändan så lyckades jag skapa ett system efter företagets behov. Systemet fungerar som det var planerat av företaget samt innehåller funktioner som var önskade.

Vid skapande av kravspecifikationen så började jag med att intervjua utvecklarna samt min handledare från företaget och utav det så tillkom de grundläggande huvudkraven till systemet.

- *Webbklienten skall kunna visa information från en databas. (Krav: SCA-HK01)*
- *Tjocka klienten skall kunna visa information från en databas. (Krav: SCA-HK02)*
- *Tjocka klienten skall kunna uppdatera information i en databas. (Krav: SCA-HK03)*

Av dessa krav så tillkom det fler som vad för typ av information och hur det skall visas på dessa klienter.

- *Det skall finnas en sökruta där användaren kan skriva in ett sökord på webbklienten. (Krav: SCA-WC05)*
- *Det skall finnas en sökruta där användaren kan skriva in ett sökord på den tjocka klienten. (Krav: SCA-FC07)*

Då blev det bestämt att informationen som kommer visas på klienterna beror på vad användaren söker efter.

Vilken typ av information som skall kunna sökas skall bestämmas av administratören.

- *Ett administrationsgränssnitt för webbtjänsten skall kunna konfigurera databas frågor (queries). (Krav: SCA-HK07)*

Med hjälp av detta administrationsgränssnitt så kommer alltså administratören kunna ändra vilka tabeller och kolumner som skall kunna sökas med klienterna.

- *Det skall gå att ändra databas frågorna, vilka kolumner och tabeller som skall läsas med administrationsgränssnittet. (Krav: SCA-AI03)*

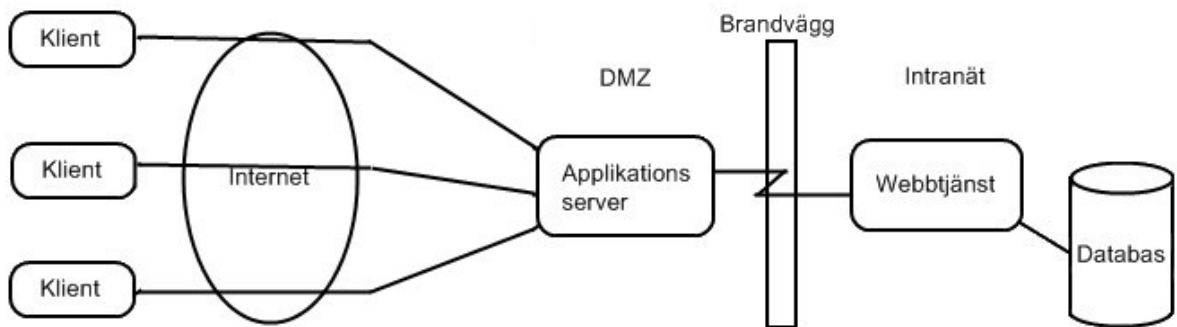
Men inte bara vilken information som skall kunna visas på klienterna, så skall man även kunna tillåta vilka telefoner som får använda klienterna med administrationsgränssnittet. Detta krav bestämdes av företaget.

- Med administrationsgränssnitt skall det gå att konfigurera vilka telefoner får använda klienterna. (Krav: SCA-HK08)
- Med admingränssnittet skall det gå att lägga till identitetsnummer som används av webbklienten. (Krav: SCA-AI05)
- Med administrationsgränssnittet skall det gå att ta bort identitetsnummer som används av webbklienten. (Krav: SCA-AI06)

Efter att jag har skrivit ner de mest grundläggande kraven i kravspecifikation så fortsatte jag med nästa moment som var att undersöka om säkerheten inom webbtjänst.

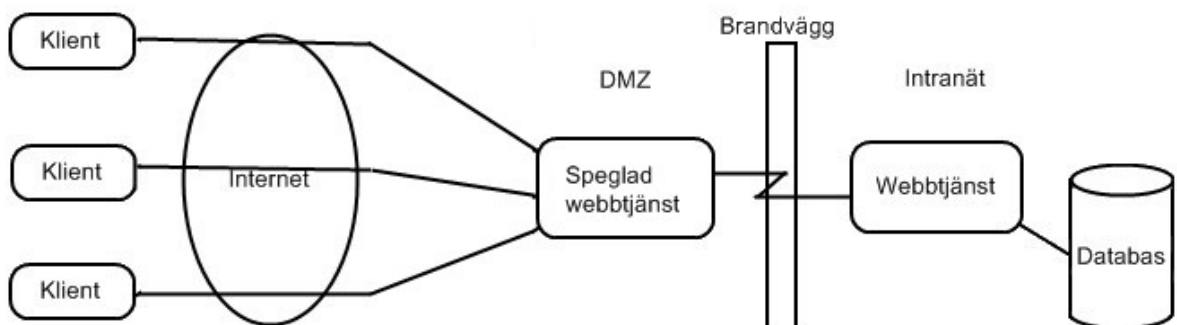
Som beskrivet i kapitel 3 så har jag bestämt mig för att begränsa undersökningen till tre lösningar:

- A) Mellan webbtjänst och klienterna kan man använda sig av en applikationsserver som tar hand om trafiken i mellan. Figur 7.1 beskriver hur uppbyggnaden kommer se ut.



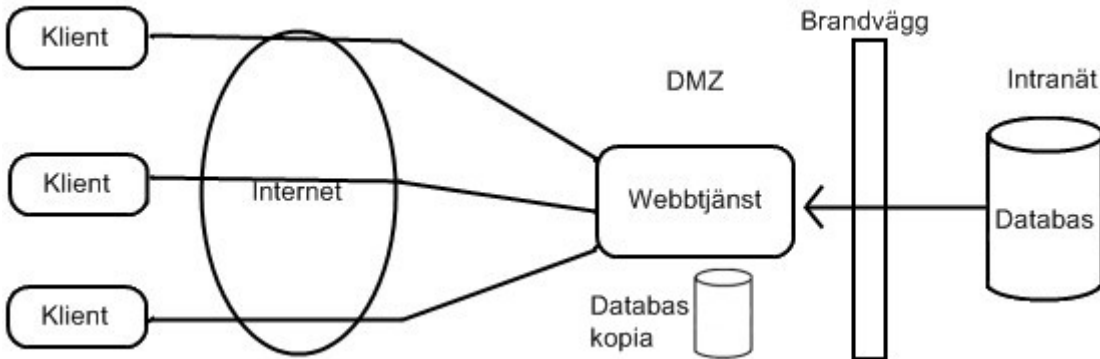
Figur 7.1

- B) En annan lösning som liknar lösning A) är att istället för att använda sig av en applikationsserver så använder man en ytterligare webbtjänst som använder endast metoderna som klienterna behöver. Figur 7.2 visar kortfattat denna lösning.



Figur 7.2

C) Istället för att webbtjänsten kommunicerar med databasen, så kan man ha en databas som kopierar över det data som behövs till webbtjänsten med jämna mellanrum eller vid ändring av data i databasen.



Figur 7.3

Av dessa tre lösningar så bestämde jag mig av att använda lösning B) . En mer detaljerad beskrivning av lösningarna samt konklusion kan läsas i kapitel 5.

Ett par krav tillkom av säkerhets undersökningen som beskriver den lösning jag valde.

- *Webbtjänsten skall vara skyddad mot obehöriga. (Krav: SCA-HK05)*
- *All databas information skall vara skyddad mot obehöriga. (Krav: SCA-HK06)*
- *Alla funktioners returvärden som används av webbklient eller mobilklient skall skickas till en speglad webbtjänst först. (Krav: SCA-WS05)*
- *Webb- och mobilklienterna får endast kommunicera med den speglade webbtjänsten, inte webbtjänsten som ligger i det lokala nätet. (Krav: SCA-WS06)*

7.1 Webbtjänst

Efter att ha fått ihop en del krav som beskriver system i helhet, påbörjade jag utveckling av systemet. Det första jag började utveckla var webbtjänsten.

Denna webbtjänst kommer att fungera som ett kommunikationscentrum för klienterna och databasen. All information som klienterna vill ha måste frågas webbtjänsten först som frågar databasen och returnerar sedan all information till klienterna som åtkoms. Men på grund av säkerhetsrisk så går det inte att bara använda sig av en sådan webbtjänst utan som det var förklarat i kapitel 7 kom jag fram till att använda mig av en ytterligare webbtjänst som skall ligga exponerad mot internet medan den fullständiga webbtjänsten skall vara skyddad av en brandvägg och ligga lokalt med databasen. Klienterna kommer alltså att kommunicera med den speglade webbtjänsten istället och sedan vidarebefordrar frågan till den lokala webbtjänsten som tar det vidare till databasen.

Den speglade webbtjänsten består av endast de nödvändiga metoderna som den tjocka klienten samt webbklienten använder. Den lokala webbtjänsten innehåller fler metoder eftersom administrationsgränssnittet använder andra metoder som inte de två klienterna gör.

7.2 Administrationsgränssnitt

Jag börja utveckla administrationsgränssnittet lite parallellt med webbtjänsten, vilket gav nya idéer samt vilket sätt webbtjänsten behöver programmeras för att administrationsgränssnittet skall fungera.

Det finns ett antal saker som kan administreras, detta kan väljas längst upp på administrationsgränssnittet (figur 7.2.1).

Ändra yttre sök-query :: Ändra inre sök-query :: Ändra visa detaljer-query
Access kontroll för webbklient :: Access kontroll för mobilsklient :: Övriga inställningar

Figur 7.2.1

Yttre och inre sök-query är bara ett mer generellt uttryck för de två sökningar som klienterna har.

Första gången administratören installerar systemet måste samtliga databasfrågor ändras för att systemet ska fungera korrekt. Vid ändring av en databasfråga så skall ett antal fält fyllas i (figur 7.2.2).

Kolumner som skall ingå, med en IDkolumn först (kolumner skiljs med komma ,)

Nuvarande kolumner: `cmp_id`, `name` Namn, `address` Adress, `phone` Telefon

Tabellen som skall sökas:

Nuvarande tabell: `company`

Kolumnen som sökordet skall jämföra med:

Nuvarande sökkolumn: `name`

Spara

OBS! Var god och använd inga citattecken (") i texten.

Vad ska denna sökning kallas: (t.ex. Företag)

Nuvarande söknamn: `Företag`

Spara

Figur 7.2.2

För att kunna fylla i denna information måste administratören veta hur databasen ser ut och vilka namn olika tabeller/kolumner har. De andra databasfrågorna konfigureras på ett ungefär samma sätt som figur 7.2.2.

Vid access kontroll för webbklient ser gränssnittet ut på följande sätt som figur 7.2.3 visar:

Länk som hemsidan befinner sig på:

Nuvarande länk: <http://utvtest/WebClient/>

Nuvarande id som är tillåtna:

e06d69d7-e983-437a-ad41-f8c9e8b0c2ce	▲
f0b48f93-a7a7-45f6-9fb5-f403bab04520	
b1c5ebbd-d9c0-4d1e-9f1c-55dfc71cb01d	☰
f868ba5f-5291-4a28-aa7b-1858a019c5ff	
08d66121-1840-4c0c-b82f-d11706e5f0d2	
0f7ca269-bf12-435b-9b54-d160f1c645fb	▼

Ta bort valt id: Kopiera ett valt id:

Lägg till ett id:

Figur 7.2.3

I det första textfältet skall adressen till hemsidan fyllas i, detta kommer att hjälpa administratören när denne ska dela ut ID till flera användare. Om administratören väljer ett ID i listan och sedan trycker ”Kopiera”-knappen så kommer hela länken tillsammans med ID kopieras till utklipp som administratören sedan kan snabbt mejla till en användare. Detta underlättar när det finns till exempel 20 användare som skall ha ett ID var. Följande länk är ett exempel på vad som kopieras vid tryck av ”Kopiera”-knappen:

<http://utvtest/WebClient/SearchPage.aspx?device=f868ba5f-5291-4a28-aa7b-1858a019c5ff>

Under access kontroll för mobilklient ser gränssnittet ut som i figur 7.2.4.

Nuvarande mobil ID som är blockerade:

Ta bort valt id:

Nuvarande pinkoder som är tillåtna:

730802

840653

946213

132696

788453

998541

Ta bort valt pinkod:

Lägg till en pinkod:

Figur 7.2.4

Den första listan innehåller alla telefon ID som är blockerade. En telefon blir blockerad då en telefon har gjort tre felaktiga PIN inmatningar. Endast administratören kan ta bort ett blockerat ID. Den andra listan innehåller alla pinkoder som kan användas för att acceptera en telefon. Efter att en användare har matat in en korrekt pinkod så registreras telefon ID som ett tillåtet ID och användaren behöver aldrig mata in någon kod för att använda klienten igen. Efter att en pinkod har används, tas den bort automatiskt.

Den sista biten av administrationsgränssnittet är ”Övriga inställningar” (figur 7.2.5).

Ändra connection string för databasen

Skriv in hela connection string vid ändring.

Exempel: Data Source=MIN_DATOR\SQLEXPRESS;Initia

Nuvarande sträng: Data Source=UTVTEST\SQLEXPRESS

Ändra

Ändra Administrationslösenord

Gamla lösenordet:

Nya lösenordet:

Bekräfta lösenordet:

Ändra

Figur 7.2.5

På denna del finns det endast två olika funktioner; ändra anslutningssträng för databasen och ändra administrationsgränssnitt lösenordet.

7.3 Webbclient

Jag fokuserade på att utveckla webbclienten före den tjocka klienten. Webbclienten kan användas både från en dator samt mobiltelefoner som har stöd för webbläsare. Storleken på klienten är anpassad till mobiltelefoner med minst 128 pixlar bred skärm. Figur 7.3.1 beskriver hur den slutgiltiga webbclienten ser ut.



En väldigt simpel hemsida som består av en sökruta, två sökval samt en knapp. Det första som kontrolleras när man öppnar hemsidan är ID:et som befinner sig i sökvägen. Länken till denna hemsida kan till exempel vara så här: <http://utvtest/WebClient/SearchPage.aspx?device=c1bc410a-9bb0-43f9-a6fe-ce8029b5e71d> Värdet som kommer efter "device=" är alltså ett ID.

Figur 7.3.1

Om detta ID inte finns i access listan så kommer hemsidan inte visas utan något annat, vilket beror på vilken webbläsare man använder. Till exempel Mozilla Firefox kommer visa ett felmeddelande (figur 7.3.2) medan Internet Explorer visar en helt tom sida.

XML-tolkningsfel: inget element hittat
Adress: http://utvtest/WebClient/SearchPage.aspx?device=test
Radnummer 1, Kolumn 1:

^

Figur 7.3.2

Det som skickas till webbtjänsten när man trycker på sökknappen är vilken sträng som finns i textrutan, vilken sökning som är vald samt vilken sidnummer som nuvarande skall visas. Webbtjänsten bearbetar denna information och skapar en databas fråga som skickas till databasen.

En tabell med alla träffar som sökordet fick skickas från databasen till webbtjänsten som gör om tabellen till ett XML dokument, ett exempel av hur ett sådant XML dokument kan se ut beskriv i figur 7.3.3, sedan skickas detta dokument till webbklienten och skriver ut informationen på skärmen, figur 7.3.4 är ett exempel på detta.

```

- <Table>
  - <Row0>
    <ColumnName0>cmp_id</ColumnName0>
    <Column0>28</Column0>
    <ColumnName1>Namn</ColumnName1>
    <Column1>Qoppa</Column1>
    <ColumnName2>Adress</ColumnName2>
    <Column2>Qoppa blv. 27</Column2>
    <ColumnName3>Telefon</ColumnName3>
    <Column3>027-272727</Column3>
  </Row0>
</Table>

```

Figur 7.3.3



Figur 7.3.4

Om en sökning har fler än 25 träffar så kapar webbtjänsten av förfrågan till databasen så att det returneras endast 26 träffar. På så vis blir det mindre belastning på både databasen samt mobiltelefonens bandbredd. Varför det kapas av vid 26 och inte 25 är för att webbklienten måste veta om det finns fler än 25 träffar eller 25 träffar exakt. Om fallet är att det finns fler träffar så kommer webbklienten endast att skriva ut 25 träffar på skärmen men i slutet av sidan kommer sidnummer finnas som fungerar som länkar (figur 7.3.5).

Webbklienten kan alltså räkna ut hur många träffar totalt som finns genom att göra en förfrågan till webbtjänsten som ställer en databasfråga som endast returnerar värdet på hur många träffar som finns. Med denna speciella databas fråga så är belastningen på databasen fortfarande så liten som möjligt. När webbklienten vet hur många träffar det totalt finns kan den räkna ut hur många sidor som behövs.



Figur 7.3.5

En annan funktion som webbklienten har är [visa detaljer]. Varje träff har ett eget ID (i figur 7.3.3 står detta ID i första kolumnvärdet) när användaren trycker på en [visa detaljer]-länk så skickas detta ID till webbtjänsten. Webbtjänsten använder detta ID som ett sökord och används i en databasfråga för att få tag på alla objekt som tillhör den träff användaren valde.

En liten förklaring till vad ett sådant objekt är till exempel: Den första sökningen söker på företag, när man då trycker [visa detaljer] för ett företag så kommer alla kontaktpersoner från detta företag visas på skärmen (figur 7.3.6). En sådan kontaktperson är alltså ett objekt.



Även denna sida är begränsad till 25 per sida. I ett fall då det finns fler än 25 objekt så behandlas detta precis på samma sätt som sökningssidan gjorde.

Den sista funktionen som webbklienten har är den alternativa sökningen, i detta fall heter den ”Efternamn”. Denna sökning söker alltså inte efter företag utan den söker igenom alla objekt istället.



Figur 7.3.6

Webbtjänsten har ett par speciella funktioner till denna typ av sökning som skapar en databas fråga vilket returnerar

```

- <Table>
- <Row0>
  <ColumnName0>cmp_id</ColumnName0>
  <Column0>3</Column0>
  <ColumnName1>Företag</ColumnName1>
  <Column1>Gamma</Column1>
  <ColumnName2>Adress</ColumnName2>
  <Column2>Gamma blv.4</Column2>
  <ColumnName3>Telefon</ColumnName3>
  <Column3>033-333333</Column3>
  <ColumnName4>NextTable</ColumnName4>
  <Column4>next</Column4>
  <ColumnName5>Namn</ColumnName5>
  <Column5>Fredrik</Column5>
  <ColumnName6>Iname</ColumnName6>
  <Column6>Karlsson</Column6>
  <ColumnName7>TelefonNr</ColumnName7>
  <Column7>035-45645</Column7>
  <ColumnName8>Mobil</ColumnName8>
  <Column8>072-5856582</Column8>
  <ColumnName9>Extra</ColumnName9>
  <Column9>Two</Column9>
</Row0>
</Table>

```

inte endast objekt träffarna utan även vilka företag dessa kontaktpersoner befinner sig i (Figur 7.3.7). Skillnaden mellan dessa två sökningar kan jämföras med figur 7.3.3 samt 7.3.8 som beskriver XML dokumenten som skickas till webbklienten.

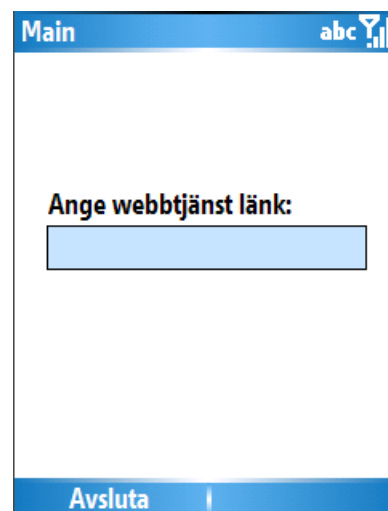
Figur 7.3.7

Figur 7.3.8

7.4 Tjock klient

Utveckling av den tjocka klienten påbörjade jag lite senare under projektets gång. Denna klient fungerar i stora drag på samma sätt som webbklienten, layout ser lite annorlunda ut dock. En större sak som skiljer de två klienterna åt är autentiseringen och med den tjocka klienten kan man uppdatera information också.

Första gången en telefon använder denna klient så kommer den att be användaren skriva in länken för den speglade webbtjänsten (figur 7.4.1). Detta är nödvändigt eftersom detta system kommer att användas av olika företag och då ska ju inte alla klienter kommunicera med samma webbtjänst. När användaren har skrivit in länken till den speglade webbtjänsten, kontrolleras det först om den är korrekt och klienten får kontakt med webbtjänsten. Efter det så registreras länken i telefonens register vilket gör att användaren aldrig behöver skriva in det igen så länge länken är giltig.



The screenshot shows a mobile application interface. At the top, there is a blue header bar with the text 'Main' on the left and 'abc' followed by a signal strength icon on the right. Below the header, the main content area is white and contains the text 'Ange webbtjänst länk:' followed by a light blue rectangular input field. At the bottom of the screen, there is another blue bar with the text 'Avsluta' on the left and a vertical line separator on the right.

Figur 7.4.1

När telefonen får kontakt med webbtjänsten så kommer telefonens ID nummer skickas till webbtjänsten för att kontrolleras om det är ett tillåtet ID eller



The screenshot shows a mobile application interface. At the top, there is a blue header bar with the text 'Main' on the left and 'abc' followed by a signal strength icon on the right. Below the header, the main content area is white and contains the text 'Ange pin:' followed by a light blue rectangular input field. At the bottom of the screen, there is another blue bar with the text 'Avsluta' on the left and a vertical line separator on the right.

Figur 7.4.2

blockerat. Om telefonens ID är varken blockerat eller tillåtet så kommer en PIN kod förfrågan visas av klienten (figur 7.4.2).

Om användaren anger ett korrekt PIN så kommer webbtjänsten registrera telefon ID:et automatiskt men om användaren anger 3 felaktiga PIN försök så kommer telefon ID:et blockeras och endast administratören kan ta bort ett blockerat ID. Om en telefon är registrerad så kommer användaren aldrig behöva ange något för att använda klienten.



Ange sökord:

1 Sök Företag

2 Sök Efternamn

3 Visa detaljer

4 Inställningar

5 Avsluta

Fler träffar

Meny

Figur 7.4.3



Ange sökord:

Namn:

Adress: Qoppa blv. 27

Telefon: 027-272727

Fler träffar

Meny

Figur 7.4.4

Alpha

AlphaBeta

Beta

Chi

Delta

Digamma

Epsilon

Eta

Gamma

Iota

Kappa

Lambda

Mu

Done

Figur 7.4.5

Användaren väljer vilken typ av sökning som skall göras genom menyn. Med samma meny finns ”visa detaljer” funktionen men den är endast tillgänglig när minst en träff finns i ComboBoxen. Den tjocka klientens funktioner använder



Ange sökord:

Namn:

Adress: Alpha blv.3

Telefon: 011-111111

1 Nästa 25 >

2 < Föregående 25

Fler träffar

Meny

Figur 7.4.6

samma webbtjänst metoder som webbklienten gör. En liten skillnad är när det finns fler än 25 träffar tillgängliga. Med den tjocka klienten kan man inte välja vilken sida, utan endast nästa 25 eller föregående 25 träffar (figur 7.4.6).

En speciell funktion som den tjocka klienten har men inte webbklienten har, är funktionen som kan uppdatera databas information. Denna funktion kan komma åt från ”Visa detaljer” fönstret (figur 7.4.7). Den information som kan ändras är endast de objekt som visades på ”Visa detaljer” fönstret när man valde ”Ändra information.

På ”Ändra information” fönstret så finns det två stycken ComboBoxar, i den första samlas alla objekt och i den andra så finns de kolumner vars värde kan editeras (figur 7.4.8). Det nuvarande värdet uppdateras varje gång en ändring görs. Sist på sidan finns det en textruta där användaren kan ange det nya värdet till det objektet och kolumn som är valt. När användaren har valt ett objekt, en kolumn och skrivit in ett värde i textrutans så uppdateras informationen när användaren trycker den vänstra meny knappen.

SCAUpdate abc 🔍

Vilket namn:
Henrik Paulsson

Vilket fält:
phone

Nuvarande värde:
025-60408

Nytt värde:
[Empty text input field]

Ändra | Meny

Figur 7.4.8

SCADetails 🔍

Olle Larsson
Telefon: 025-11245
Mobil: 077-4564656

Henrik Paulsson
Telefon: 025-60408
Mobil: 070-4546465

1 Ändra information
2 Återvänd till sökning
3 Avsluta

Fler träffar | Meny

Figur 7.4.7

7.5 Sammanfattning

En liten svaghet systemet har är själva webbtjänsten. Det tillkommer nya risker efterhand som svagheter inom den nuvarande säkerheten, alltså måste webbtjänsten uppdateras med jämna mellanrum.

En annan svaghet med webbklienten och dess autentisering är att dessa ID som används för access kan läckas ut. En administratör kan ta bort ett ID från accesslistan, men det är inte alltid en administratör vet om ett ID är läckt eller inte. Men med den tjocka klienten så är det telefon ID som registreras, så det blir svårare att läcka ut access till den tjocka klienten.

Det är en layout brist på administrationsgränssnittet samt den tjocka klienten. De två klienterna är inte särskilt tilltalande och kunde ha gjorts bättre. Med administrationsgränssnittet så kommer administratören få problem om denne inte vet något om SQL, om mer tid hade spenderats på administrationsgränssnittet så skulle det kunna göras mer visuellt och lättare för alla. Layouten för webbklienten fick många positiva kommentarer från IXX IT-Partner AB.

Fast de få bristerna på systemet, så är det rätt användarvänligt. Vid demonstration hos IXX så fick utvecklarna testa systemet och det blev ett antal positiva kommentarer om systemets uppbyggnad och webbklientens layout. Utvecklarna testade även att koppla upp systemet med en slumpmässigt vald databas och tested fungerade utmärkt.

En fråga från utvecklarna om möjlighet att lägga till en kommentar för telefon ID/webbklient ID för att administratören skall veta vilka ID tillhör vilka personer. Vilket skulle vara mycket möjlig men fick tyvärr prioriteras bort p.g.a. andra uppgifter som skulle göras först.

I slutändan så var IXX nöjd med produkten, acceptanskravet har uppfyllts samt att jag har gjort ett gott intryck för företaget vilket jag hade hoppats på.

7.6 Framtida arbete

Detta system har många möjligheter av att utvecklas vidare. Företaget ville ha systemet så dynamiskt som möjligt, att det ska gå att konfigurera så mycket som möjligt. Som systemet är just nu så finns det en del som inte är konfigurerbar som till exempel antalet träffar per sida.

Det är inte endast den dynamiska delen som kan utvecklas mer utan även fler funktioner kan läggas till, ett exempel är att det ska finnas en möjlighet att lägga till en helt ny kontaktperson till ett företag, med det nuvarande systemet så går det endast att uppdatera befintliga personer. En funktion där man kan lägga till historik till diverse kontaktpersoner är också en annan möjlig utveckling.

8 Generella referenser

1. www.stardeveloper.com/articles/display.html?article=2003052201&page=1 (Februari 2008) En artikel hos Stardeveloper.
2. www.codeguru.com/csharp/csharp/ (Februari 2008)Handledning om C# av CodeGuru.
3. www.codeproject.com/KB/webservices/aspwebsvr.aspx (Februari 2008)Handledning om webbtjänst av CodeProject.
4. <http://www.codeproject.com/KB/security/DotNetCrypto.aspx> (Mars 2008)Kryptering i C# av CodeProject.
5. <http://msdn2.microsoft.com/en-us/library/aa302384.aspx> (Mars 2008)Autentisering inom ASP.NET.
6. www.windowsecurity.com/articles_tutorials/web_server_security/ (April 2008) WindowsSecurity hemsida.
7. msdn2.microsoft.com/en-us/library/ms996415.aspx (April 2008) MSDN hemsida om säkerhet inom webbtjänst.
8. msdn2.microsoft.com/en-us/library/ms951257.aspx (April 2008) MSDN hemsida om WS-Security.
9. arno.uvt.nl/show.cgi?fid=28992 (April 2008) Ett project om webbtjänst säkerhet av Thomas Schepers.
10. msdn2.microsoft.com/en-us/library/ms977256.aspx (April 2008) MSDN hemsida om webbtjänst säkerhet.

9 Terminologi

I fallet finns det ett antal begrepp och termer som används för att beskriva saker som inte säger sig naturligt vad de är. Dessa kommer att beskrivas i detta avsnitt.

1. **Databas** - samling information som är organiserad på ett sådant sätt att det är lätt att söka efter och hämta enskilda bitar information, samt ofta även att ändra informationen.
2. **Brandvägg** - kontrollmekanism för nätverkstrafik. Brandväggar kopplas mellan två eller flera nätverk (där ett ofta är Internet) men kan även installeras som en mjukvara i den dator som ska skyddas.
3. **Applikation** - även kallad som tillämpningsprogram, typ av datorprogram som fyller ett direkt syfte för användaren. Tillämpningsprogram kan vara nyttoprogram som ordbehandlingsprogram och kalkylprogram.
4. **Webbtjänst** - betecknar webbaserade datorprogram som kommunicerar och samarbetar dynamiskt med andra webbtjänster på samma vis som en människa kan surfa till olika webbsidor.
5. **Tunn klient** - program som inte körs lokalt på datorn, till exempel en webbsida som körs från en server.
6. **Tjock klient** - program som körs lokalt på datorn, till exempel ordbehandlingsprogram som installeras på datorn.
7. **Intranet/Internt nät** - intranät är ett slutet nätverk för datorer. Det är styrt vilka som ska ha tillgång till systemet. Det är tänkt att intranet ska vara en kanal mellan personer inom den egna organisationen, ibland kan det förekomma att externa användare kan få tillgång, till exempel kunder.
8. **Autentisering** - kontroll av uppgiven identitet, till exempel vid inloggning.
9. **Databas fråga** - en rad kod med villkor som skrivs till en databas som returnerar information som stämmer överens med villkoren.
10. **XML** - eXtensible Markup Language (XML) är ett universellt och utbyggbart märkspråk. Syftet med XML är att kunna utväxla data mellan olika informationssystem. Detta görs genom att skicka data som ren text; text som även kan förstås av människor.

11. **DMZ** - förkortning av *DeMilitarized Zone*, "demilitariserad zon", ett vanligt förekommande arrangemang för datorer som måste vara mer eller mindre direkt anslutna till Internet för att fylla sin funktion och därför exponeras mot omvärlden, till exempel externa webbservrar och e-postservrar.
12. **Port** - är i datorteknisk terminologi en passage där i förväg specificerade typer av datatrafik kan passera eller nekas passera. Denna kontroll kan exempelvis ske med hjälp av en brandvägg.
13. **SSL** - Secure Sockets Layer är en säkerhetsmekanism som uppfanns av Netscape och används för att kryptera kommunikationen mellan två värdar. Ofta är ena värden en webbservare (ex. onlinebutik) och den andra är en webbläsare (ex. Internet Explorer). Tanken med att kryptera data som sänds till och från dessa två värdar är att ingen annan användare på nätverket (oftast internet) ska kunna avlyssna eller manipulera data.

10 Bilagor

10.1 Kravspecifikation

10.1.1 Inledning

Ett system skall skapas som skall kommunicera med en databas genom en brandvägg. Systemet kommer att bestå av ett par applikationer:

1. Webbtjänst som skall kommunicera med databasen och skicka information till webb- och den tjocka klienten.
2. Ett admingränssnitt som har konfigurationsmetoder för webbtjänsten.
3. Webbaserad klient, skall kunna visa data från databasen.
4. Tjock klient, skall kunna visa samt uppdatera data från databasen.

Den webbaserade applikationen skall stödjas av så många mobil webbläsare som möjligt.

10.1.2 Definitioner och förkortningar

- **Webbläsare:** Blädderprogram eller webbklient, för sökning på webbservrar i Internet. Webbläsaren tolkar html-koden och gör dokumentet lämpligt för visning på skärmen. De kan även arbeta med hyperlänkar och hämta information från andra ställen i nätet.
- **HTML:** Hypertext Markup Language, det språk som webbsidorna skrivs med och som är ett språk för strukturerade hypertextdokument. HTML-dokument är ett dokument på en server som är kodad (skriven) i Html-kod.
- **Databas:** En samling information som är organiserad på ett sådant sätt att det är lätt att söka efter och hämta enskilda bitar information, samt ofta även att ändra informationen.
- **Tjock klient:** Ett program som körs lokalt på en lokal dator, till skillnad från en tunn klient som är till exempel en webbsida som körs från en server.
- **Webbtjänst:** betecknar webbaserade datorprogram som kommunicerar och samarbetar dynamiskt med andra webbtjänster på samma vis som en människa kan surfa till olika webbsidor.
- **XML:** eXtensible Markup Language (XML) är ett universellt och utbyggbart märkspråk. Syftet med XML är att kunna utväxla data mellan olika informationssystem. Detta görs genom att skicka data som ren text; text som även kan förstås av människor.

10.1.3 Krav

10.1.3.1 Generella krav

Krav:	Beskrivning:
SCA-GK01	Generella krav förkortas med SCA-GKxx där xx är ett stigande nummer som börjar på 01.
SCA-GK02	Huvudkrav förkortas med SCA-HKxx där xx är ett stigande nummer som börjar på 01.
SCA-GK03	Webbtjänst krav förkortas med SCA-WSxx där xx är ett stigande nummer som börjar på 01.
SCA-GK04	Administrationsgränssnitt krav förkortas med SCA-AIxx där xx är ett stigande nummer som börjar på 01.
SCA-GK05	Webbklient krav förkortas med SCA-WCxx där xx är ett stigande nummer som börjar på 01.
SCA-GK06	Tjock klient krav förkortas med SCA-FCxx där xx är ett stigande nummer som börjar på 01.

10.1.3.2 Huvudkrav

Krav:	Beskrivning:
SCA-HK01	Webbklienten skall kunna visa information från en databas.
SCA-HK02	Tjocka klienten skall kunna visa information från en databas.
SCA-HK03	Tjocka klienten skall kunna uppdatera information i en databas.
SCA-HK04	Webbtjänsten skall ha metoder som kommunicerar med databasen.
SCA-HK05	Webbtjänsten skall vara skyddad mot obehöriga.
SCA-HK06	All databas information skall vara skyddad mot obehöriga.
SCA-HK07	Ett administrationsgränssnitt för webbtjänsten skall kunna konfigurera databas frågor (queries).
SCA-HK08	Med administrationsgränssnitt skall det gå att administrera vilka telefoner som får använda klienterna.
SCA-HK09	Endast registrerade användare får använda klienterna.
SCA-HK10	Klienterna skall autentiseras vid användning.

10.1.3.3 Webbtjänst

Krav:	Beskrivning:
SCA-WS01	Webbtjänsten skall programmeras i C#
SCA-WS02	Om accesstiden mellan webbtjänsten och databasen överstiger 10 sekunder, blir det timeout.

SCA-WS03	Databas information skall skickas i egen definierat XML schema.
SCA-WS04	Max antalet träffar webbtjänsten får från databasen är 26 .
SCA-WS05	Alla funktioners returvärden som används av webbklient eller mobilklient skall skickas till en speglad webbtjänst först.
SCA-WS06	Webb- och mobilklienterna får endast kommunicera med den speglade webbtjänsten, inte webbtjänsten som ligger i det lokala nätet.

10.1.3.4 Administrationsgränssnitt

Krav:	Beskrivning:
SCA-AI01	Det skall finnas en autentiseringssida för admingränssnittet.
SCA-AI02	Det skall gå att ändra lösenordet för autentiseringssidan med själva administrationsgränssnittet.
SCA-AI03	Det skall gå att ändra databas frågorna, vilka kolumner och tabeller som skall läsas med administrationsgränssnittet.
SCA-AI04	Med administrationsgränssnittet skall det gå att testa databas frågorna.
SCA-AI05	Med admingränssnittet skall det gå att lägga till identitetsnummer som används av webbklienten.
SCA-AI06	Med administrationsgränssnittet skall det gå att ta bort identitetsnummer som används av webbklienten.
SCA-AI07	Det skall gå att lägga till pin koder som används av den tjocka klienten med administrationsgränssnittet.
SCA-AI08	Det skall gå att ta bort pin koder som används av den tjocka klienten med administrationsgränssnittet.
SCA-AI09	Det skall gå att ta bort telefon ID som är blockerade med administrationsgränssnittet.

10.1.3.5 Webbklient

Krav:	Beskrivning:
SCA-WC01	Webbklienten skall programmeras i asp.net.
SCA-WC02	Webbklienten kan bara användas om identitetsnummret är ett giltigt nummer.
SCA-WC03	Webbklienten skall vara anpassad till mobiltelefoner.
SCA-WC04	Webbklienten skall vara anpassad till minst 128 pixlar breda skärmar.
SCA-WC05	Det skall finnas en sökruta där användaren kan skriva in

	ett sökord på webbklienten.
SCA-WC06	Det skall finnas två olika sökningar på webbklienten.
SCA-WC07	Det skall finnas en knapp som utför sökningen.
SCA-WC08	Max 25 träffar får visas per sida.
SCA-WC09	Om fler än 25 träffar finns, skall länkar som tar användaren till andra sidor visas längst ner på sidan.
SCA-WC10	Varje träff skall ha en "Visa detaljer" länk.
SCA-WC11	"Visa detaljer" sidan skall visa en ny lista på annan information.
SCA-WC12	Längst ner på sidan skall en "Hem" länk visas.
SCA-WC13	"Hem" länken tar användaren till första sidan.
SCA-WC14	All text och förklaringar skall skrivas på svenska.

10.1.3.6 Tjock klient

Krav:	Beskrivning:
SCA-FC01	Tjocka klienten skall programmeras i compact framework.
SCA-FC02	Vid användning av klienten skall en kontroll först göras om telefon ID är blockerat.
SCA-FC03	Om telefon ID är inte blockerat skall det kontrolleras om det existerar i tillåtna ID listan.
SCA-FC04	Om telefon ID är varken blockerat eller finns i tillåtna ID listan, skall ett inmatning av pinkod fönster visas för användaren.
SCA-FC05	Om användaren matar in en tillåten pinkod skall telefon ID registreras i tillåtna ID listan.
SCA-FC06	Om inmatning av pinkod är felaktigt tre gånger skall telefon ID registreras i blockerade ID listan.
SCA-FC07	Det skall finnas en sökruta där användaren kan skriva in ett sökord på den tjocka klienten.
SCA-FC08	En combobox skall fyllas med varje träff från en utförd sökning.
SCA-FC09	Användaren kan utföra båda sökningarna genom ett val på den högra menyn.
SCA-FC10	Om minst en träff existerar skall det gå att välja "Visa detaljer" genom den högra menyn.
SCA-FC11	Om träffarna överstiger 25 stycken skall den vänstra menyn bli tillgänglig på sök fönstret samt visa detaljer fönstret.
SCA-FC12	Med den vänstra menyn på sök fönstret samt visa detaljer fönstret kan användaren gå till nästa sida som visar nästa max 25 träffar.

SCA-FC13	Med den vänstra menyn på sök fönstret samt visa detaljer fönstret kan användaren gå till föregående sida som visar de föregående 25 träffarna.
SCA-FC14	På visa detaljer fönstret har den högra menyn ett val som tar användaren till ett fönster där användaren kan ändra och uppdatera databas informationen.
SCA-FC15	På ändra information fönstret skall det finnas en combobox som innehåller varje träff från visa detaljer fönstret.
SCA-FC16	På ändra information fönstret skall det finnas en combobox som innehåller alla kolumn namn som går att ändra.
SCA-FC17	På ändra information fönstret skall det finnas en textbox där användaren kan ange det nya värdet längst ner på fönstret.
SCA-FC18	På ändra information fönstret skall den vänstra menyvalet utföra ändringen.
SCA-FC19	All text och förklaringar skall skrivas på svenska.