



VAASAN AMMATTIKORKEAKOULU  
VASA YRKESHÖGSKOLA  
UNIVERSITY OF APPLIED SCIENCES

Jonas Kurt Mikael Granbäck

# ANALYS OCH DESIGN AV FAKTURERINGSSYSTEM

Företagsekonomi och turism  
2010

## **FÖRORD**

Detta lärdomsprov har gjorts våren 2010 i syftet att erhålla tradenomexamen inom utbildningsprogrammet för informationsbehandling vid Vasa yrkeshögskola. Kenneth Norrgård har fungerat som handledare.

Vasa 16 oktober 2010

---

Jonas Granbäck

VASA YRKESHÖGSKOLA

Utbildningsprogrammet för informationsbehandling

### ABSTRAKT

Författare	Jonas Granbäck
Lärdomsprovets titel	Analys och design av faktureringsystem
År	2010
Språk	svenska
Sidantal	34
Handledare	Kenneth Norrgård

---

Syftet med detta arbete har varit att planera och programmera ett faktureringsprogram till ett mindre företag. Efter att det gått igenom med företaget vilka funktioner som behövdes framkom det att programmet skulle göras simpelt och med endast de nödvändigaste funktionerna. Efter det gjordes ett analys- och designdokument där man använde sig av teori från boken Objektorienterad analys (Mathiassen m.fl. 2001). Och efter det påbörjades utvecklingen av programmet.

Resultatet blev ett delvis fungerande faktureringsprogram men en del av de funktioner som företaget önskade sig, har i skrivande stund ännu inte hunnit bli gjorda. Nu i efterhand har det också upptäckts några funktioner som saknas och dessa kommer också att implementeras senare.

VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES  
Utbildningsprogrammet för informationsbehandling

## ABSTRACT

Author                      Jonas Granbäck  
Title                         Analysing and designing of an invoicing system  
Year                         2010  
Language                    Swedish  
Pages                        34  
Name of Supervisor      Kenneth Norrgård

---

The purpose of this thesis has been planning and programming an invoicing system for a smaller company. After a research was done with the company about what functions they needed from the system, it was revealed that the system should be done as a simple system with just the most useful functions. A software requirements specification and a software design document was made with theory from the book, Objektorienterad analys och design (Mathiassen m.fl. 2001). Later on the coding of the system began.

The result became a partially functional invoicing system but some of the functions that the company wanted to have, has at the time of writing this thesis not been implemented yet. Afterwards it was also discovered that some functions are missing and these will also be implemented later.

---

Keywords                    invoicing, programming, database, analyze, design

## INNEHÅLL

ABSTRAKT .....	2
ABSTRACT .....	3
1 INLEDNING .....	6
1.1 Upprinnelse .....	6
1.2 Bakgrund .....	6
1.3 Syfte och metod .....	6
1.4 Avgränsningar .....	7
1.5 Arbetets upplägg .....	7
2 ANALYS .....	8
2.1 Uppgiften .....	8
2.1.1 Syfte .....	8
2.1.2 Systemdefinition .....	8
2.1.3 Systemomgivning .....	9
2.2 Analys av problemområde .....	10
2.2.1 Arkitektur .....	10
2.2.2 Databasmodell .....	12
2.2.3 Beskrivning av tabeller .....	13
2.3 Analys av användningsområde .....	16
2.3.1 Aktörer och användningsfall .....	16
2.3.2 Funktioner .....	20
2.3.3 Användargränssnitt .....	20
2.3.4 Teknisk plattform .....	23
2.4 Rekommendationer .....	23
2.4.1 Systemets användbarhet och genomförbarhet .....	23
2.4.2 Strategi och fortsatt utveckling .....	24
3 DESIGN .....	25
3.1 UPPGIFTEN .....	25

3.1.1	Syfte .....	25
3.1.2	Rättelser av analysen.....	25
3.1.3	Kravspecifikation .....	26
3.2	Teknisk plattform.....	27
3.2.1	Urustning och systemgränssnitt .....	27
3.2.2	Designspråk och databassystem.....	27
3.3	Arkitektur.....	27
3.3.1	Komponent- och processarkitektur .....	27
3.3.2	Standarder .....	28
3.4	Rekomendationer .....	28
3.4.1	Systemets användbarhet.....	28
3.4.2	Plan för ibruktagande .....	28
3.4.3	Implementeringsplan.....	28
4	PROGRAMMERING .....	29
4.1	Tillvägagångssätt .....	29
5	SAMMANFATTNING.....	33
	KÄLLFÖRTECKNING.....	34

# 1 INLEDNING

## 1.1 Upprinnelse

Detta lärdomsprov kommer att behandla hur man kan analysera och planera ett mjukvarusystem. I detta arbete presenteras en analys och en design av ett faktureringsprogram. Uppkomsten av detta projekt blev till mest av en slump efter en diskussion om faktureringsystem med Mats Granbäck under hösten 2009, och det blev också tidpunkten för starten av detta projekt. Eftersom Mats Granbäck var i behov av ett faktureringsystem till sitt mindre företag, föreslogs det honom att man kan skräddarsy ett sådant system till honom.

Företaget som är grundat årsskiftet 2009 och beläget i Pörtom, består endast av en person och därför behöver systemet endast fungera lokalt på en arbetsstation vilket underlättar projektet.

Efter att idén hade fötts började jag fundera över vilka funktioner som systemet behövde och det sammanställdes ett analysdokument. Efter att dessa var färdigställda så påbörjades själva utvecklingen av programmet.

## 1.2 Bakgrund

Eftersom det kan vara svårt att upprätthålla en bokföring i pappersformat underlättar det mycket med ett digitalt faktureringsystem. Snabbt och enkelt kan man söka fram äldre beställningar eller kontaktuppgifter till tidigare kunder som exempel. Dessutom har man alltid alla ”dokument” på ett och samma ställe med ett digitalt faktureringsystem och man har inga problem med lösa papper.

## 1.3 Syfte och metod

Syftet med projektet är att gå igenom vilka funktioner som uppdragsgivaren behöver och att utifrån det skapa ett så enkelt och användarvänligt system som möjligt.

En stor vikt har lagts på att inga onödiga funktioner skall finnas med utan endast väsentligaste funktionerna. Och genom att använda sig av ett analysdokument som grund för detta får man en väldigt bra insyn över vilka funktioner som skall finnas med före man börjar själva programmeringsprocessen.

#### **1.4 Avgränsningar**

Jag valde att avgränsa vissa funktioner i själva programmeringen p.g.a. tidsbrist tills denna dokumentation gjordes. Men gränssnittet är delvis gjort och systemets funktioner kommer att färdigställas i efterhand. Och även möjliga tillägg kommer att implementeras om sådana uppkommer.

#### **1.5 Arbetets upplägg**

Lärdomsprovet är uppdelat i en teoretisk del samt en praktisk del. Den teoretiska delen är dokumenterad i kapitel 2-3, där det går igenom en analys och en design av bokföringssystemet. För att skapa dessa dokument diskuterades det med beställaren om vilka funktioner som han önskade och genom att skapa dokumenten fick man en underliggande modell till systemet. Modellen för analysen och designen har hämtats från boken *Objektorienterad analys och design (Mathiassen m.fl. 2001)* och beskriver på ett objektorienterat vis systemets olika komponenter samt deras relationer till varandra.

Den praktiska delen består av själva programmeringen av systemet som gjordes i programmet Microsoft Visual Basic 2007 och relationsdatabasen som gjordes i Microsoft Office Access 2007. Den praktiska delen presenteras endast som skärmdumpar i kapitel 4. Den praktiska delens resultat kan också ses som lärdomsprovets slutresultat.



## **2 ANALYS**

### **2.1 Uppgiften**

#### **2.1.1 Syfte**

Systemet som används av användaren skall innehålla register över kunder, produkter, beställningar och fakturor. Med hjälp av programmets funktioner skall användaren ha möjlighet att lägga till, ta bort och redigera poster i systemets register samt skriva ut fakturor och rapporter.

#### **2.1.2 Systemdefinition**

Systemdefinitionen är här beskriven med hjälp av VATOFA –kriteriet (Mathiassen 2005, 58.) och denna beskrivning borde ge en god bild på de önskemål och krav som finns på systemet. VATOFA består av sex olika delar, en del för varje bokstav. V står för villkor under vilka systemet kommer att skapas och användas. A står för användningsområde, d.v.s. var och vem som kommer att använda sig av systemet. T står för teknologi som används för att utveckla och använda sig av systemet. O står för de objekt som finns i problemområdet. F står för funktionalitet, d.v.s. vilka systemets funktioner skall vara. Och A står för ansvar som systemet har mot sin omgivning.

#### **Villkor**

Systemet kommer att ligga på en lokal dator men när man vill köra en uppdatering av produkt registret från importören bör den vara kopplad till internet.

Man skall kunna söka bland kunder, beställningar, produkter och fakturor.

Skriva ut fakturor och rapporter på produkter i det egna lagret.

Kunna ge en faktura två olika status, betald och icke betald.

Systemets språk kommer endast att vara svenska.

Användningsområde	Systemet skall användas på en dator av en användare och alla funktioner skall vara tillgängliga.
Teknologi	Systemet skall köras i Windows miljö, databasen skall göras som en relationsdatabas i Microsoft Access 2007 och programmeringen görs i Visual .NET-miljö.
Objekt	En kund, en produkt, en faktura, en beställning, en lagerlista.
Funktionalitet	Huvudfunktionerna för användaren är att lägga till produkter, kunder, beställningar, skapa och skriva ut faktura. (en närmare analys av funktionerna i form av en händelsetabell och en funktionslista ingår sedan i kapitel 2.3.2).
Ansvar	Systemet är tänkt att vara ett fristående system och skall inte skicka till andra system. Allt som registreras görs via formulärinmatning förutom då man lägger in/uppdaterar produkter från importören.

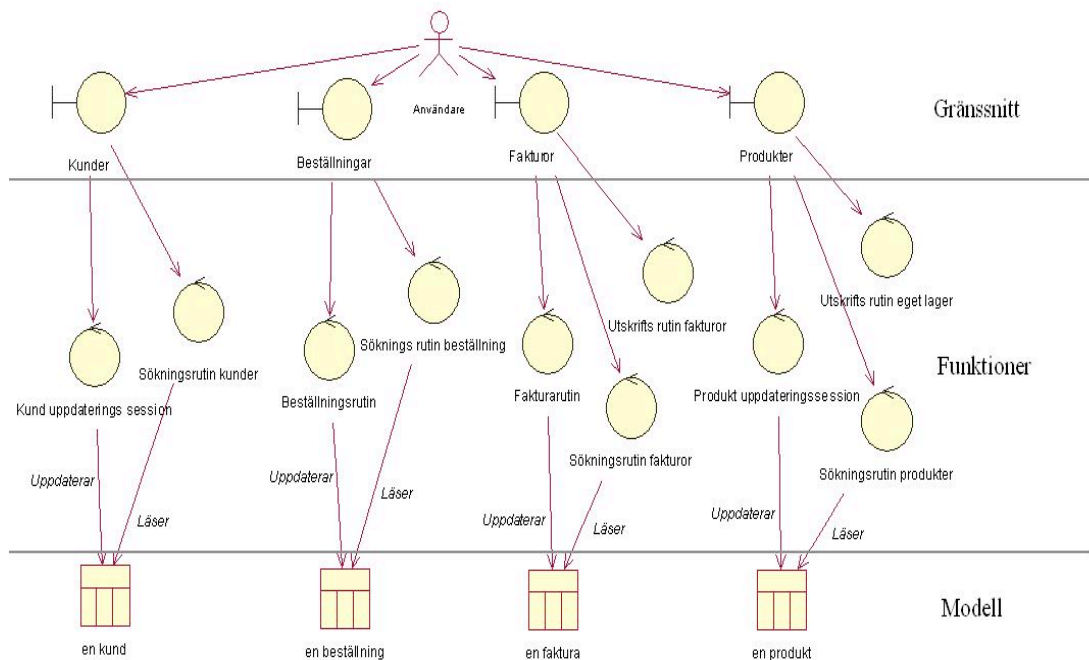
### 2.1.3 Systemomgivningen

Problemområde:

Systemet skall hjälpa användaren att hålla koll på följande register. Kunder, produkter, beställningar och fakturor. Systemet skall också ha funktioner som hjälper användaren att lägga till och redigera poster i tidigare nämnda kategorier. Systemet skall också ta emot produktlistor från importören

Användningsområde:

Systemet skall användas av en användare i taget. Med hjälp av systemet skall användaren kunna lägga till och editera poster i kund, faktura, produkt och beställnings registerna med hjälp av systemets funktioner. Samt kunna skriva ut fakturor och produktregister.



Figur 1. Objektdiagram.

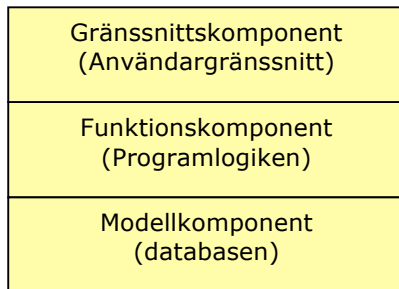
Objektdiagrammet visar hur gränssnittet, funktionerna och databasen kommer höra samman.

## 2.2 ANALYS AV PROBLEMOMRÅDE

I kapitel 2.2 beskrivs relationsdatabasen och de tabeller som har definierats i denna analys.

### 2.2.1 Arkitektur

Systemet följer den s.k. treskiktsarkitektur och består således av en gränssnitts-, funktions- och modellkomponent.



Figur 2. Principbild på en treskiktsarkitektur.

Indelningen utgår ifrån objekt diagrammet och principen är följande:

1. Alla objekt som definierats med stereotypen ”boundary” blir en del av gränssnittskomponenten.
2. Alla objekt som definierats med ”control” tillhör funktionskomponenten.
3. Alla objekt som definierats som ”table” bildar modellkomponenten.

Gränssnittskomponentens objekt:

- Kunder –fönster
- Beställningar –fönster
- Produkter –fönster
- Fakturor –fönster

Funktionskomponentens objekt:

- Kund uppdateringssession (Lägg till/Ta bort/Söka/Redigera/Spara/Avbryt)
- Produkt uppdateringssession (Lägg till/Ta bort/Söka/Redigera/Spara/Avbryt)
- Fakturarutin (Lägg till/Ta bort/Söka/Redigera/Spara/Avbryt/Skriv ut)

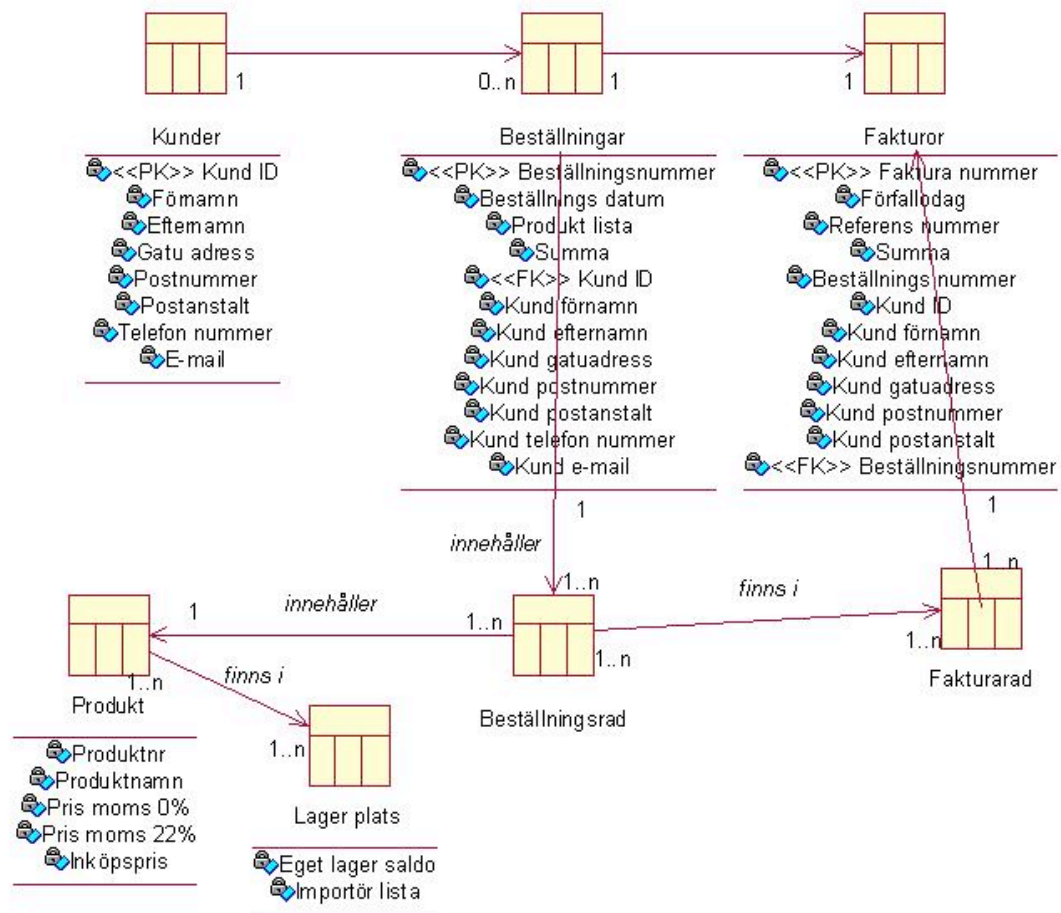
- Beställningsrutin (Skapa/Lägg till/Ta bort/Söka/Redigera/Spara/Avbryt)

#### Modellkomponentens objekt

- Kunder
- Produkter
- Beställningar
- Fakturor

#### **2.2.2 Databasmodell**

Här visas en databas modell och relationerna mellan de olika databaserna samt varje databas informationsfält. PK står för Primary Key eller primär nyckel och FK står för Foreign Key eller främmande nyckel. En primär nyckel har alltid ett unikt värde så att värdet kan kopplas till en speciell post. En främmande nyckel är värdet på en primärnyckel i en annan tabell. Då en kolumn deklarerats som en främmande nyckel så innebär det beroende mellan de två tabellerna. Modellen kan ses som ett ER-schema och borde kunna ligga till underlag för relationsdatabasen som skall grundas.



Figur 3. Databasmodell.

### 2.2.3 Beskrivning av tabeller

I analyskedet är följande tabeller identifierade. I figurerna finns en beskrivning för varje kolumn och vilken datatyp som angetts. Den första tabellen kunder (Figur 3.) innehåller uppgifter för kundregistret, så som kundens namn och kontaktuppgifter. Den andra tabellen beställningar (Figur 4.) innehåller data om beställningar som beställningsnummer, beställda produkter och beställarens uppgifter. Följande tabell fakturor (Figur 5.) är skapad från en beställning och innehåller kundens kontaktuppgifter samt beställningens summa, fakturans referenser och information om fakturan är betald

eller obetald. Följande tabell produkter (Figur 6.) innehåller uppgifter för produktregistret, så som produktnamn och priser. Tabellen lager plats (Figur 7.) innehåller uppgifter om i vilket lager en specifik produkt finns. Tabellen beställningsrad (Figur 8.) innehåller uppgifter om de produkter som finns i beställningen. Tabellen fakturarad (Figur 9.) innehåller uppgifter om de produkter som tillhör en specifik faktura.

kunder : Tabell			
	Fältnamn	Datatyp	Beskrivning
🔑	kund_ID	Räknare	Kundes ID-nummer
	Fornamn	Text	Kundens förnamn
	Efternamn	Text	Kundens efternamn
	Gatu_adress	Text	Kundens gatu adress
	Postnummer	Tal	Kundens postnummer
	Postadress	Text	Kundens postadress
	Telefon_nummer	Tal	Kundens telefon nummer
▶	Email	Text	Kundens e-post adress

Figur 4. Tabellen kunder.

beställningar : Tabell			
	Fältnamn	Datatyp	Beskrivning
🔑	beställnings_nummer	Räknare	Beställningens nummer
	kund_id	Tal	Beställarens kund ID-nummer
	beställnings_datum	Datum/tid	Datum då beställningen gjordes
	produktlista	Text	Lista på produkter i beställningen
	summa	Valuta	Beställningens summa
	kund_fornamn	Text	Beställarens förnamn
	kund_efternamn	Text	Beställarens efternamn
	kund_gatuadress	Text	Beställarens gatu adress
	kund_postnummer	Tal	Beställarens postnummer
	kund_postanstalt	Text	Beställarens postanstalt
	kund_telefon_nummer	Tal	Beställarens telefon nummer
	kund_email	Text	Beställarens e-post adress

Figur 5. Tabellen beställningar.

fakturor : Tabell			
	Fältnamn	Datatyp	Beskrivning
	faktura_ID	Räknare	Fakturans ID-nummer
	faktura_nummer	Tal	Fakturans nummer
	bestallningsnummer	Tal	Fakturans tillhörande beställningsnummer
	forfallodag	Datum/tid	Fakturans förfallodag
	referens_nummer	Tal	Fakturans referens nummer
	summa	Valuta	Fakturans summa
	kund_id	Tal	Fakturans tillhörande kund ID
	kund_fornamn	Text	Fakturans tillhörande kund förnamn
	kund_efternamn	Text	Fakturans tillhörande kund efternamn
	kund_gatuadress	Text	Fakturans tillhörande kund gatu adress
	kund_postnummer	Tal	Fakturans tillhörande kund postnummer
	kund_postanstalt	Text	Fakturans tillhörande kund postanstalt
	Betald	Jaj/Nej	Värde på om fakturan är betald eller inte

Figur 6. Tabellen fakturor.

produkter : Tabell			
	Fältnamn	Datatyp	Beskrivning
	produkt_ID	Räknare	Produktens ID-nummer
	Produkt_kod	Text	Produktens kod
	produkt_namn	Text	Produktens namn
	Pris_moms_0	Tal	Produktens pris med moms 0%
	pris_moms_22	Tal	Produktens pris med moms 22%
	Inköps_pris	Tal	Produktens inköpspris

Figur 7. Tabellen produkter.

Lager_plats : Tabell			
	Fältnamn	Datatyp	Beskrivning
	Produkt_kod	Tal	Produktens kod
	Eget_lager_saldo	Tal	Saldo i eget lager
	Importör	Tal	Saldo i importörens lager

Figur 8. Tabellen lager plats.



beställningsrad : Tabell			
	Fältnamn	Datatyp	Beskrivning
🔑	beställnings_nummer	Tal	Beställningens nummer
	produkt_kod	Text	Produktens kod
	produkt_namn	Text	Produktens namn
	pris_moms_0	Tal	Produktens pris med moms 0%
	pris_moms_22	Tal	Produktens pris med moms 22%

Figur 9. Tabellen beställningsrad.

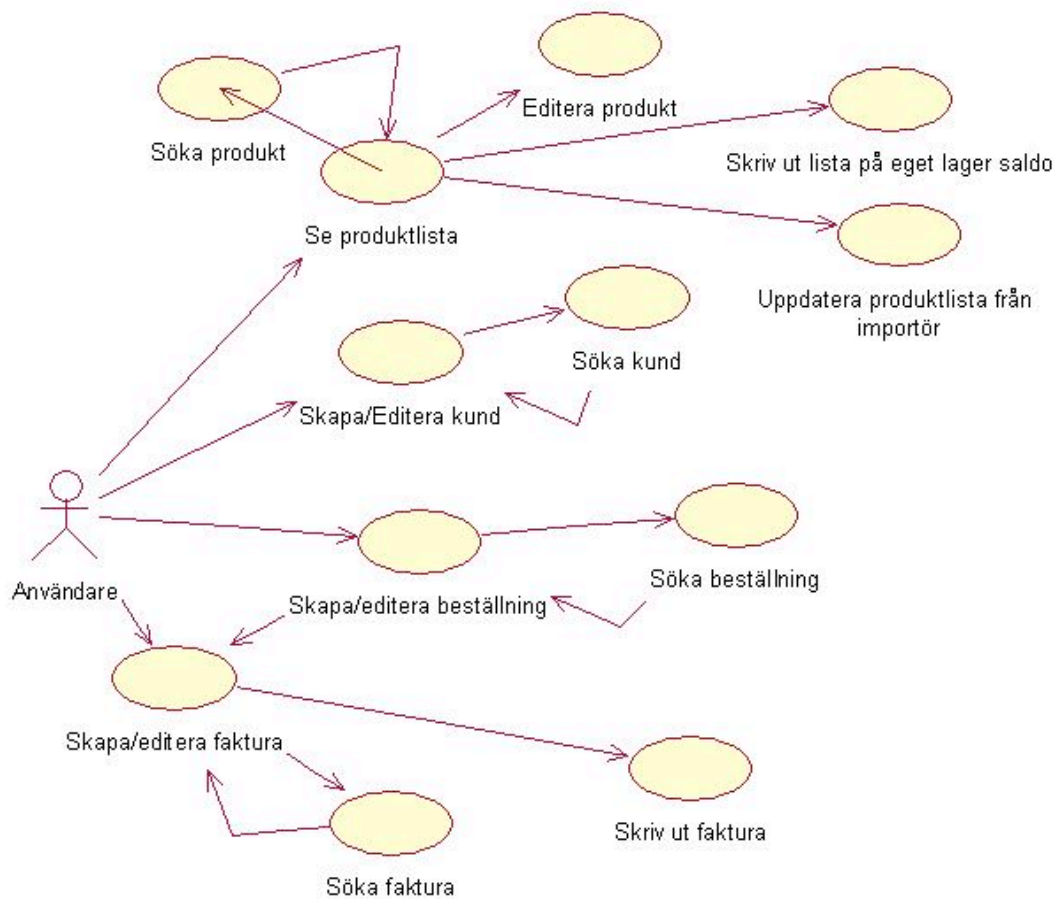
fakturarad : Tabell			
	Fältnamn	Datatyp	Beskrivning
🔑	faktura_nummer	Tal	Fakturans nummer
	Beställnings_nummer	Tal	Beställningens nummer
	Produktlista	Text	Produkter som tillhör fakturan

Figur 10. Tabellen fakturarad.

## 2.3 ANALYS AV ANVÄNDNINGSSOMRÅDE

### 2.3.1 Aktörer och användningsfall

Systemets användningsfall finns angivna i figur 10 och är analyserade i detta kapitel. Dessa analyser ligger till underlag för funktionstabellen i kapitel 2.3.2.



Figur 11. Systemets användningsfall

### Aktörer

I analysen har endast en aktör identifierats, eftersom det endast kommer att finnas en användare av systemet. Och systemet använder sig inte av någon utomstående aktör.

### Användarfall

Följande användarfall är identifierade och analyserade. Händelseförloppen ser ut enligt följande.

Användningsfall: Skapa/editera kund.

1. Användaren lägger till en ny kund och följande uppgifter registreras:

- Förnamn.
- Efternamn.
- Gatuadress.
- Postnummer.
- Postanstalt.
- Telefon nummer.
- E-post adress.

2. Om en kund har blivit tillagd tidigare skall det finnas möjlighet att söka fram kunden och editera de registrerade uppgifterna eller att ta bort kunden.

Användningsfall: Skapa/editera beställning.

3. Användaren skapar en beställning och följande uppgifter registreras:
  - Kundens uppgifter.
  - Beställda produktens uppgifter.
4. Om en beställning har skapats skall följande händelser vara tillgängliga:
  - Söka efter specifik beställning
  - Möjlighet att editera eller ta bort beställningen.

Användningsfall: Skapa/editera fakturor.

5. Användaren skapar en faktura och följande uppgifter registreras:
  - Kundens uppgifter.
  - Beställda produktens uppgifter.
  - Fakturans förfallodag, slutsumma och betalningens status.
6. Om en faktura har skapats skall följande händelser vara tillgängliga:

- Söka efter specifik faktura.
- Möjlighet att editera eller ta bort fakturan.
- När man ser att en faktura blivit betald skall man kunna ändra status på den till betald.

Användningsfall: Skriv ut faktura.

7. Om en faktura har blivit skapad skall användaren ha möjlighet att skriva ut den.

Användningsfall: Se produktlista.

8. Användaren kontrollerar sin produktlista och kan söka specifik produkt.

Användningsfall: Lägg till/editera produkt

9. Användaren kan lägga till eller redigera en produkt.

10. Följande uppgifter registreras:

- Produktnummer.
- Produktnamn.
- Inköpspris.
- Pris moms 0 %.
- Pris moms 22 %.

Användningsfall: Skriv ut lista på eget lager saldo.

11. Användaren kan skriva ut en lista på de produkter som han har i sitt egna lager vid inventering.

Användningsfall: Uppdatera produktlista från importör.

12. Användaren kan uppdatera produktlistan från importörens produktlista.

### 2.3.2 Funktioner

Systemet bör ha de funktioner som sammanfattas i tabell 1. En funktion är direkt programmeringsbar och har en klar process som startar med en impuls från användaren t.ex. genom ett knapptryck. Funktionernas komplexitet och typ uppskattas i tabell 1 för att ha en utgångspunkt för vilka resurser som krävs och hur lång tid det tar för att realisera dem.

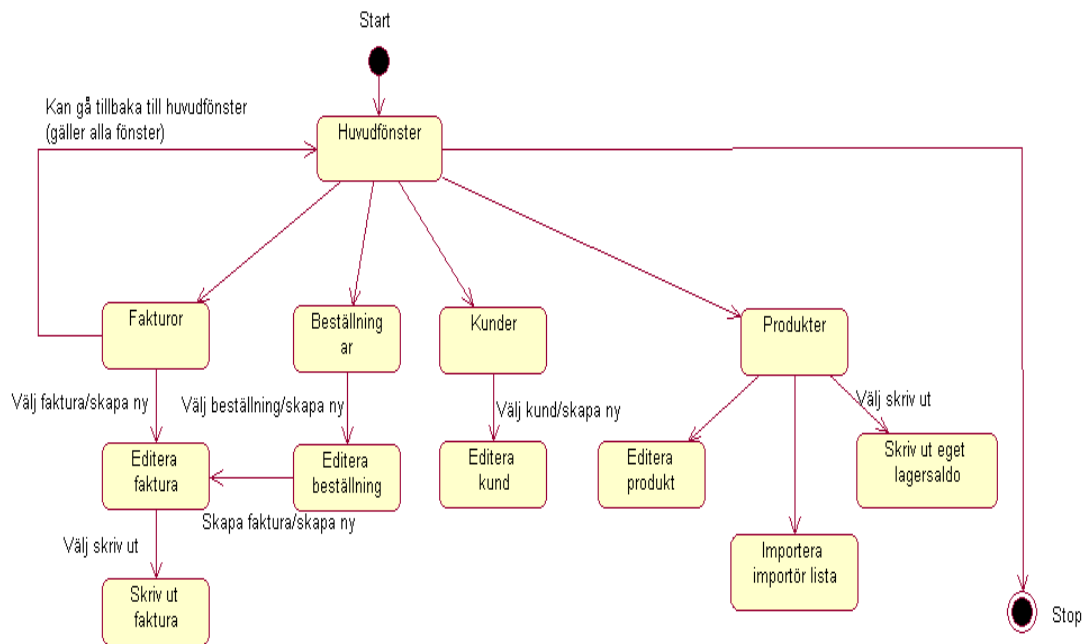
Tabell 1. Preliminär funktionslista

Nr	Funktion	Komplexitet	Typ
1	Välja under meny beställningar, kunder, produkter, fakturor	Enkel	Läs och visa
2	Uppdatering av grunddata. Lägga in ny kund, ta bort kund, editera kund, lägga in ny produkt, ta bort produkt, editera produkts, lägga in ny beställning, ta bort beställning, editera beställning, lägga in ny faktura, ta bort faktura, editera faktura, importera importörs lista	Omfattande	Uppdatera
3	Sök kund, sök beställning, sök faktura, sök produkt	Normal	Läs och visa
4	Skriva ut faktura, skriva ut lager saldo.	Enkel	Läs och skriv ut

### 2.3.3 Användargränssnitt

Systemet skall designas med ett väldigt enkelt användargränssnitt som skall vara lätt att använda. Från alla underfönster skall man kunna navigera sig till huvudfönstret direkt om man inte är i redigeringsläge och inte har sparat de ändringar man gjort. Nedan visas ett navigeringsdiagram och en prototyp av hur ett fönster för kundregistret kan komma att se ut.

Navigeringsdiagrammet visar vilka dialoger/fönster systemet kan komma att innehålla och visar hur man navigerar sig mellan olika dialoger.



Figur 12. Navigeringsdiagram.

På nästa sida har en tidig prototyp för ett navigeringsfönster gjorts. Fönstret är tänkt för kund editering. Designen kommer antagligen att börja se annorlunda ut i slutskedet men funktionsmässigt borde det inte ändras så mycket.

Kund register

Sök fält

Kund ID

Förnamn

Efternamn

Gatu adress

Postnummer

Post anstalt

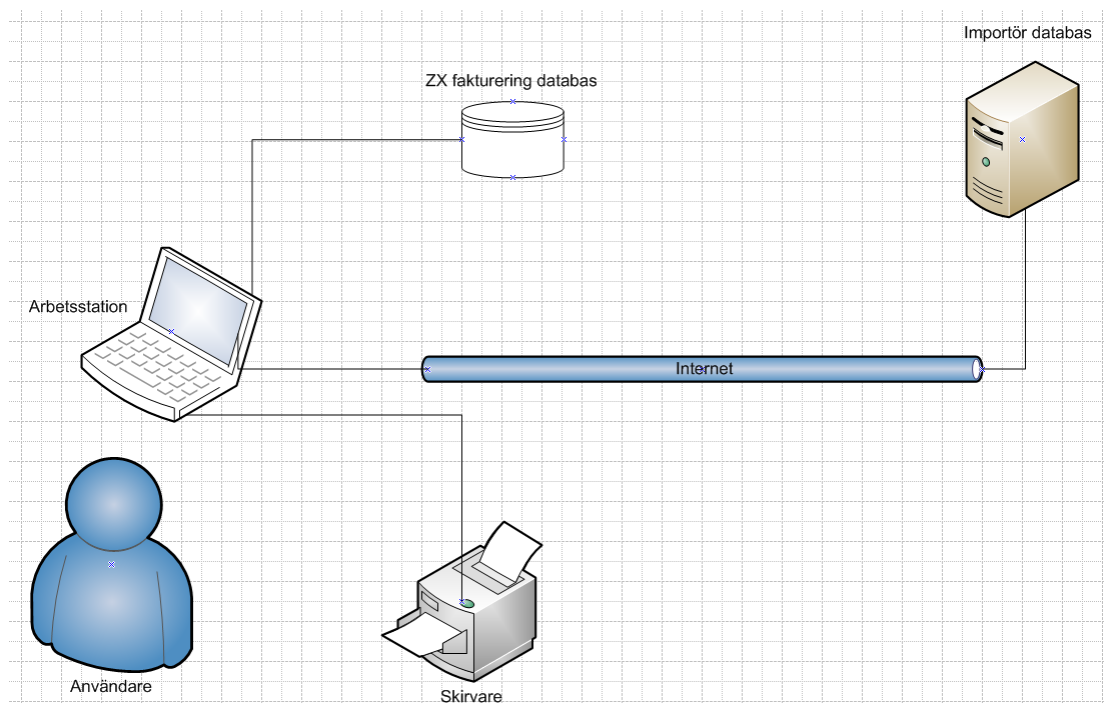
Telefon nummer

E-mail

Figur 13. Exempel på en fönsterskiss.

### 2.3.4 Teknisk plattform

Här beskrivs den tekniska plattformen för det planerade datasystemet i form av en systemskiss av apparaturer, programvara och databas samt kopplingar till övriga system.



Figur 14. Systemtes tekniska plattform.

## 2.4 REKOMMENADATIONER

### 2.4.1 Systemets användbarhet och genomförbarhet.

Ett system med den funktionalitet som har analyserats här kommer att vara användbart. Eftersom man redan i analyskedet vet att det kommer att underlätta faktureringen för användaren.

Utvecklaren har ingen tidigare erfarenhet av att utveckla system men med en god analys och grundstudier gjorda i programutveckling så ska det inte vara några



problem. Det finns även många böcker man kan dra nytta av om det är något man inte har en lösning på.

#### **2.4.2 Strategi för fortsatt utveckling**

Ingen särskild strategi kommer att användas, man kommer att börja utveckla systemet med hjälp av analysdokumentet. Och om man under utvecklingen upptäcker brister i analysen så kommer man att utforma en bättre idé som sedan kommer att dokumenteras i ett designdokument. Det system som analyserats här kommer att fodra 2-4 månaders arbete innan det är färdigt att tas i bruk.

## **3 DESIGN**

### **3.1 UPPGIFTEN**

#### **3.1.1 Syfte**

De primära syftena är att bokföra beställningar och fakturor på arbete och de produkter som kunden köpt och skriva ut fakturor. Och att också ha ett register över kunder och produkter i eget lager samt föra ett register på åtminstone en leverantörs produkter.

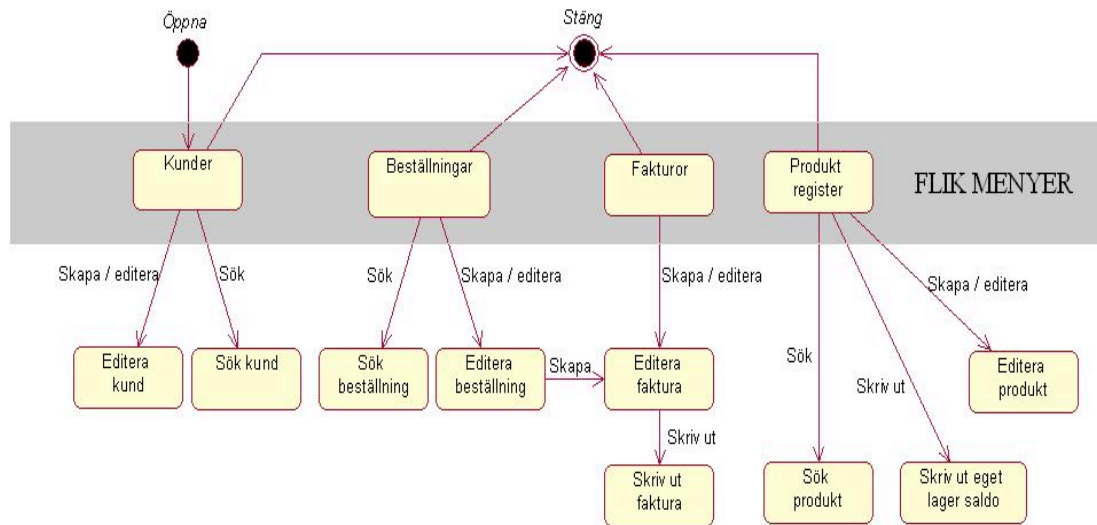
Det sekundära syftet är att användaren skall kunna lägga till kunder och produkter. Man skall även kunna uppdatera produkterna från importörens databas manuellt.

#### **3.1.2 Rättelser av analysen**

Efter en genomgång av analysen kom jag fram till att några ändringar fick göras. Databas tabellen lagerplats i databasen blir sammanförd med tabellen produkter, d.v.s. att hela lager plats tabellen blir borttagen.

Även användarfallet uppdatera produktlista från importör blir borttagen eftersom de olika importörerna har så olika fil typer på sina produktlistor, och det därför är svårt att få systemet att få dessa inlästa korrekt i databasen. I ett senare skede kan det tänkas att man lägger till denna funktion men tills vidare kommer produktlistorna att importeras direkt med Office Access.

Navigeringsdiagrammet har också ändrats ganska radikalt eftersom man valt att använda sig av olika flikar för systemets grund fönster. I stället för att man som tänkt från början alltid skulle ha navigerat från ett huvudfönster till dessa grund fönster. Ett uppdaterat navigeringsdiagram har gjorts. (Se figur 15).



Figur 15. Uppdaterat navigeringsdiagram.

### 3.1.3 Kravspecifikation

Med hjälp av systemet skall användaren kunna administrera de olika registren. Det skall vara möjligt att skriva ut fakturorna och markera dem som betalda. Söka efter en särskild post i produkt- och kundregistret samt att skriva ut kund- och produktregistren. Systemet kommer att köras lokalt på datorer med Microsoft Windows operativsystem. En mera ingående specifikation kan ses i kapitel 2.1.2 med hjälp av VATOFA-kriteriet.

ISO-standarden ISO 9126 har varit som underlag vid utformningen av detta system. Standarden definierar sex begrepp som beskriver programvarukvalitet och kan ses specifikt för detta system i följande tabell 2.

Tabell 2. Tabell över kravspecifikationen.

<b>Nr</b>	<b>Krav</b>	<b>Specifikation</b>
1	Användbarhet	Systemet är enkelt att använda och endast de nödvändigaste funktionerna finns. Systemet är gjort för att köras på Windows plattform och ingen vikt har fästs på att kunna köra mot andra system.
2	Effektivitet	Programmet är effektivt och inga långa svarstider finns.
3	Tillförlitlighet	De funktioner som finns hittills fungerar som dem ska. Eftersom programmet körs lokalt har det inte tagits någon ställning på säkerhet. Men säkerhetskopiering av data rekommenderas.
4	Underhållsvänligt	Programkoden är väl strukturerad för fortsatt utveckling.
5	Flexibilitet	Systemet är planerat för fortsatt utveckling.
6	Flyttbarhet	En del funktioner är väl indelade och kan enkelt överföras till andra system.

## **3.2 TEKNISK PLATTFORM**

### **3.2.1 Utrustning och systemgränssnitt**

Systemet designades för att utvecklas på datorer tillräckligt kraftfulla för operativsystemet Windows XP eller nyare utgåva. Stöd för .NET måste installeras och för att vissa funktioner skall fungera måste skrivare vara installerad.

### **3.2.2 Designspråk och databassystem**

Programmet utvecklas i Microsoft Visual Basic 2005. Utskriftsrapporter görs med hjälp av Crystal Reports och databasen görs i Microsoft Office Access 2007.

## **3.3 ARKITEKTUR**

### **3.3.1 Komponent- och processarkitektur**

Det valdes en traditionell skiktad arkitektur med tre komponenter: ett användargränssnitt, en funktionskomponent och en modellkomponent.

Systemet körs endast på en lokal dator och därför finns inga andra problem faktorer som inte operativsystemet har hand om. Styrningen av systemet förläggs till användargränssnittskomponenten som styrs av operativsystemet. Skrivaren hanteras också av operativsystemet.

### **3.3.2 Standarder**

Utformningen av fönster följer Windows-standarden.

## **3.4 REKOMENDATIONER**

### **3.4.1 Systemets användbarhet**

Ett av de första kriterierna som togs fram var att systemet skulle vara enkelt och lättanvänt och efter att de första versionerna av systemet har gjorts tycker man att detta uppfylls.

### **3.4.2 Plan för ibruktagande**

Eftersom systemet kommer att göras så lättanvänt och logiskt som möjligt kommer det endast att behövas en enkel genomgång av funktionerna med beställaren när systemet är färdigställt.

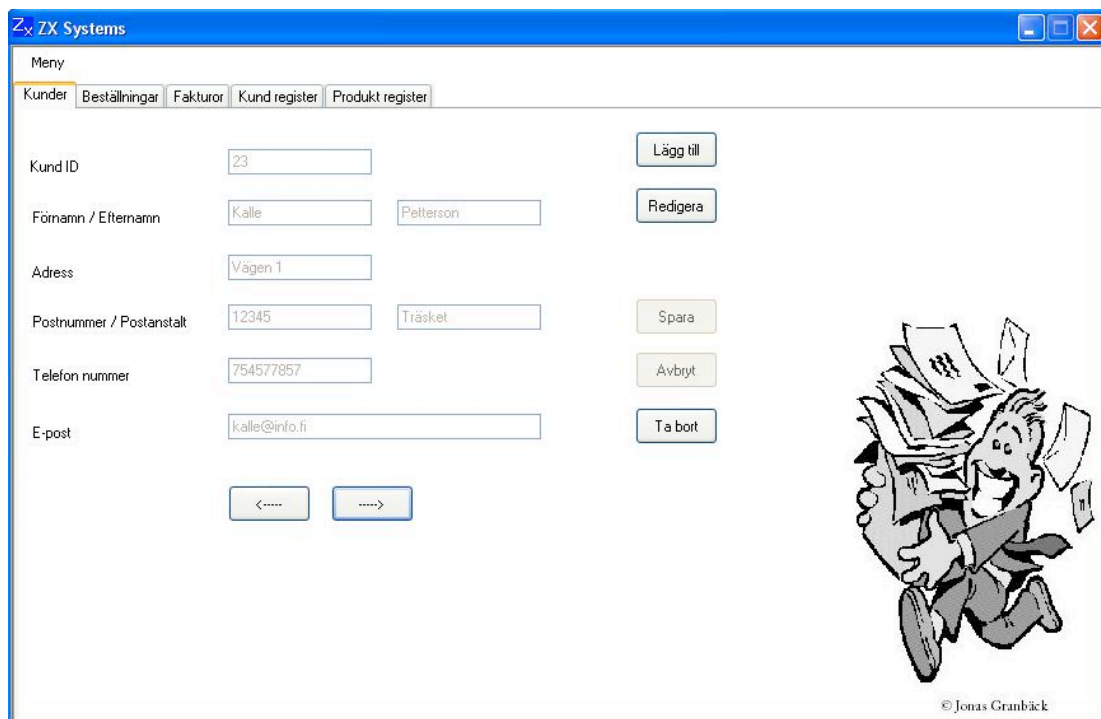
### **3.4.3 Implementeringsplan**

Systemet kommer att utvecklas under våren - sommaren 2011. Utvecklingen av systemet kommer att ske av en person. Vid upprepade tillfällen en gång per månad kommer utvecklingen att presenteras för beställaren för genomgång. Uppskattningsvis kommer utvecklingen fordra 2-4 månader för att slutföra utvecklingen.

## 4 Programmering

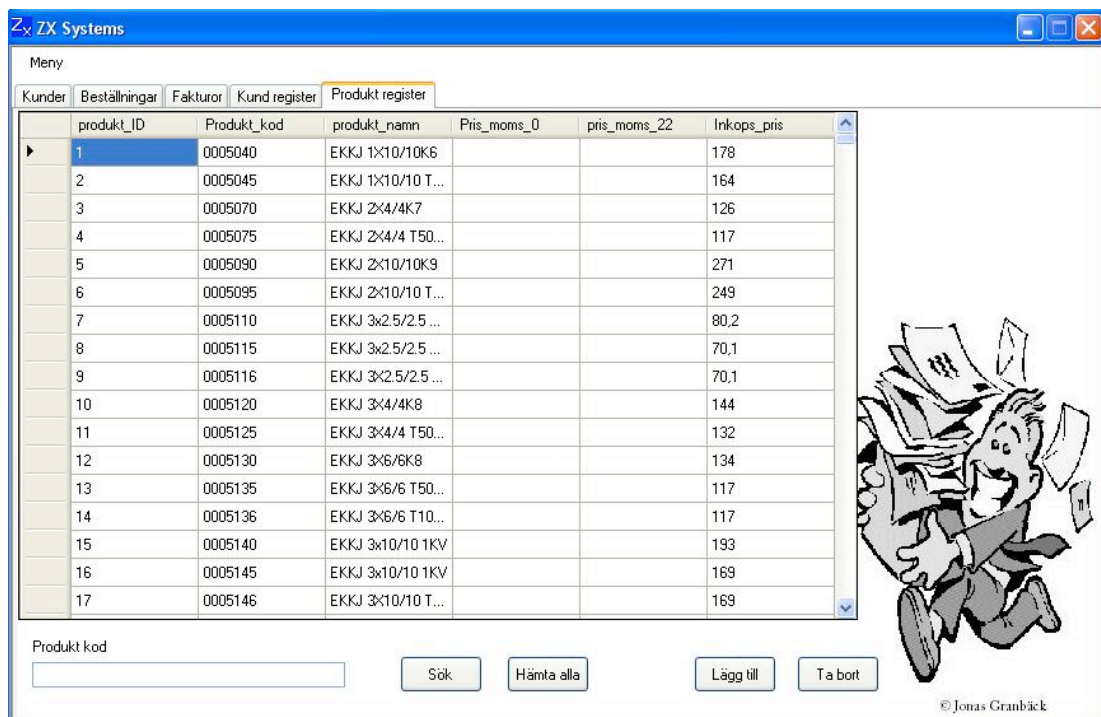
### 4.1 Tillvägagångssätt

Efter att analysdokumentet var gjort påbörjades utvecklingen av programmet. Först skapades en databas i Access med tidigare planerade tabeller och relationerna skapades mellan tabellerna. Sedan påbörjades själva kodskrivandet i Visual Studio, programmeringsspråket som användes var Visual Basic även kallat vb. Gränssnittet utvecklades först och man tyckte att det blev enklast och snyggast att använda en horisontell flikmeny som grund efter lite skissande. Inga starka färger användes utan man använde sig av en grå skala. Efter att gränssnittet hade de grundläggande kommandoknapparna och textboxarna skapades en anslutning till databasen och man började programmera funktionerna. Om jag stötte på problem med att få funktionerna att fungera tog man hjälp av olika forum på internet. Och dessa forum var till stor hjälp under programmeringen. Här nedanför kommer det att presenteras en förhandsvisning av det som nu i skrivande stund har blivit gjort.



Figur 16. Startfönster och tillika kund administration flik.

Här kommer användaren att fylla i kontakt information om aktuella kunder eller söka reda på information om kunderna om han så behöver. Med Pil-knapparna under textfälten kan man navigera mellan de registrerade kunderna. Och kommando knapparna på högersida kan man redigera och ta bort kunder ur registret.



Figur 17. Fliken produkt register.

I detta fönster kan användaren söka igenom sitt produktregister och få fram prisuppgifter. Användaren kan dessutom lägga till nya produkter eller ta bort.

The screenshot shows a software window titled "ZX Systems" with a menu bar containing "Kunder", "Beställningar", "Fakturor", "Kund register", and "Produkt register". The "Beställningar" tab is active. The interface includes a "Sök" button on the left. The main area contains several text input fields for customer and order details, and a set of action buttons on the right.

Field Label	Field Type
Beställnings nummer	Text input
Kund ID	Text input
Förnamn	Text input
Efternamn	Text input
Gatu adress	Text input
Post nummer	Text input
Post anstalt	Text input
Telefon nummer	Text input
E-post	Text input
Produkt nummer	Text input
Antal	Spin box

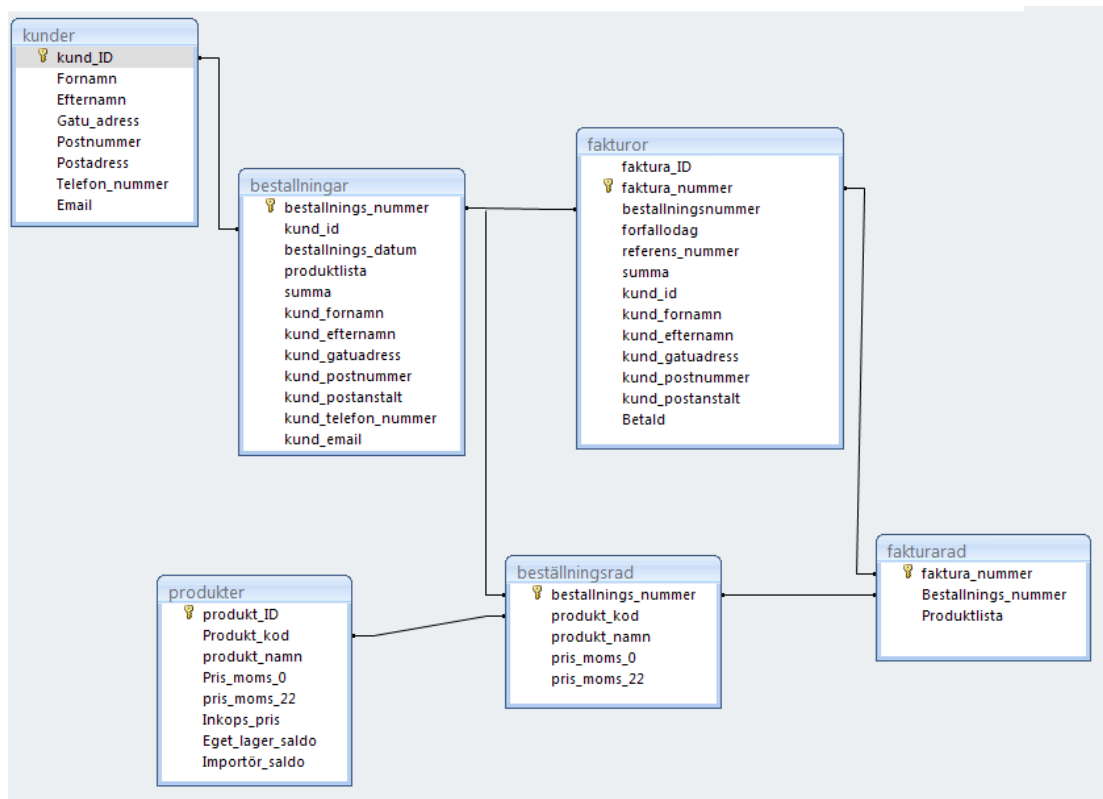
Buttons on the right side of the form:

- Skapa beställning
- Redigera
- Spara
- Ta bort
- Lägg till
- Skapa faktura

Figur 18. Fliken beställningar.

I detta fönster kan användaren hantera gjorda beställningar eller skapa nya. Genom att skriva in data i önskat textfält och klicka på sök knappen uppe till vänster så visar programmet beställningen som överensstämmer med sökvärdet. Med funktions knapparna till höger kan användaren skapa en ny beställning om han/hon fyllt i alla textfält. Eller göra ändringar i befintliga beställningar. Sedan kan användaren också skapa en faktura till beställningen med knappen "skapa faktura" längst ner till höger.





Figur 19. Databasens tabell relationer.

Här är en skärmbild tagen från databasens tabell relationer i Access. Här kan man se hur de olika tabellerna relaterar till varandra.

## 5. SAMMANFATTNING

I detta lärdomsprov har det beskrivits hur man kan analysera och planera ett faktureringsprogram. Det har varit ett lärorikt projekt att arbeta med där jag märkt att med en god planering är chansen att få en bra slutprodukt god. Eftersom jag var ganska grön inom området då jag började med detta projekt har jag lärt mig mycket om hur man bör gå tillväga och vilka aspekter man skall fundera över när man skall utveckla ett program. Gör man ingen analys före utan börjar direkt med att utveckla programmet hamnar man säkert att gå tillbaka några steg och göra om då man märker att en viss sak inte kommer att fungera. Även om man tycker det känns onödigt att sitta och analysera från en början så lönar det sig alltid efteråt. Men jag tror också att man inte kan göra en så god analys att man kan följa den exakt under utveckling, det uppstår alltid situationer som man inte kunnat förutspå i analysen. Och då hamnar man att analysera om i efterhand.

Själva programmeringen av systemet var också något som jag var ganska ovan vid när jag började. Även om det också var intressant att göra tycker jag själv att analyserandet var intressantare, och något jag skulle kunna tänka mig att fortsätta med i andra projekt. Utvecklingen av systemet har inte färdigställts ännu, men efter att man har arbetat omkring en månad med det så är gränssnittet och basfunktionerna gjorda. En del problem har uppstått under tiden av utvecklingen men det har lösts med hjälp av olika programmerings forum på webben. Eftersom tiden varit knapp under sommaren och hösten att slutföra programmet kommer det under våren att slutföras till en helt fungerande produkt.

## KÄLLFÖRTECKNING

### 1. Tryckta arbeten

Mathiassen, Lars, Munk-Madsen, Andreas, Nielsen, Peter Axel, Stage, Jan 2001. Objektorienterad analys och design. Andra upplagan. Lund. Studentlitteratur.

Fowler, Martin, Kendall Scott, 1997. UML distilled : applying the standard object modeling language. Reading: Addison-Wesley.

-, 1991. Information technology - Software product evaluation - quality characteristics and guidelines for their use. Switzerland: ISO/IEC

### 2. Elektroniska publikationer

Microsoft MSDN library, 2010. An essential source of information for developers using Microsoft tools Tillgänglig i form av www-dokument:  
URL:<http://msdn.microsoft.com/en-us/library>

Programming & Web Development Community, 2010. Tillgänglig I form av www-dokument: URL:<http://www.dreamincode.net/forums>