



# TEKNISKA HÖGSKOLAN HÖGSKOLAN I JÖNKÖPING

## **Webbapplikation för recepthantering Konvertering av recept från PDF till databas**

### **Web application for recipe management Converting recipes from PDF to database**

Ruben Fast

EXAMENSARBETE 2012  
ÄMNE Datateknik

---

Postadress:  
Box 1026  
551 11 Jönköping

Besöksadress:  
Gjuterigatan 5

Telefon:  
036-10 10 00 (vx)

Detta examensarbete är utfört vid Tekniska Högskolan i Jönköping inom ämnesområdet Datateknik. Arbetet är ett led i den treåriga högskoleingenjörsutbildningen.  
Författarna svarar själva för framförda åsikter, slutsatser och resultat.

Examinator: Inger Palmgren

Handledare: Anders Carstensen

Omfattning: 15 hp (grundnivå)

Datum: 2012-05-28

---

Postadress:  
Box 1026  
551 11 Jönköping

Besöksadress:  
Gjuterigatan 5

Telefon:  
036-10 10 00 (vx)

## **Abstract**

The purpose of this thesis was to create a web application and a recipe database for a company named Familyfood and their customers to easily manage and search for recipes. Personal purpose was to find a simple approach to transfer information from Familyfoods PDF collection, where their recipes are stored into the newly developed database. The issues are based on the purpose of the thesis. The first issue is about how to find a simple procedure to transfer information from a PDF-file to a database. The other is about how to create user friendly forms, from the viewpoint of interaction design.

The thesis began with a feasibility study and analysis in which the question dealing with the transfer of information from PDF to the database was a major focus. The solution for this issue was resolved by an export of PDF to text file and an application written in PHP that reads the text file line by line. Using this application the information can be entered into the database.

The web application consisting of the administration and search function was created and written in the programming language PHP with MySQL as database. Development and testing of the application was made in a test environment consisting of a Web server.

The application and the database were implemented into the Familyfoods website.

The results of the thesis were a functional web application and its recipe database. Besides this, also a working application and a simple procedure for transferring information from a PDF-file to a database.

## Sammanfattning

Syftet med detta examensarbete var att skapa en webbapplikation och en receptdatabas åt företaget Familyfood och deras kunder för att enkelt kunna hantera och söka efter recept. Personligt syfte var att hitta ett enkelt tillvägagångssätt för att överföra information från Familyfoods PDF-samling, där deras recept ligger lagrade till den nyutvecklade databasen. Frågeställningarna baserades på syftet och är till mängden två stycken. Den första frågeställningen handlar om hur man med ett enkelt tillvägagångssätt överför informationen från en PDF-fil till en databas. Den andra handlar om hur man skapar användarvänliga formulär, ur synpunkt av interaktionsdesign.

Examensarbetet började med en förstudie och analys där frågeställningen som handlar om överföring av information från PDF till databas fick stort fokus. Denna frågeställning löstes genom export av PDF till textfil och en applikation skriven i PHP som läser av textfilen rad för rad. Med hjälp av detta kan sedan informationen matas in i databasen.

Webbapplikationen bestående av administration och sökfunktion skapades och skrevs i programmeringsspråket PHP med MySQL som databas. Utveckling och testning av applikationen gjordes i en testmiljö bestående av en webbserver.

Applikationen och databasen implementerades sedan på Familyfoods hemsida.

Resultatet av examensarbetet blev en fungerande webbapplikation med tillhörande receptdatabas. Förutom detta skapades även en applikation och ett enkelt tillvägagångssätt för att överföra informationen från en PDF-fil till en databas.

### Nyckelord

PHP, PDF, MySQL, SQL, Formulärdesign

## **Förord**

Jag vill rikta ett stort tack till Familyfood för att de tillät mig att göra detta examensarbete. Jag vill även rikta ett tack till Peter Püschel för all hjälp vid implementeringen av webbapplikationen på Familyfoods webbplats. Även ett tack till min handledare Anders Carstensen som har stöttat mig och hjälpt mig under examensarbetet.

Till sist ett tack till Luke Wroblewski för att han tillät mig att använda bilder från hans bok (*Web Form Design* [8]) i denna rapport.

# Innehållsförteckning

<b>I</b>	<b>Inledning .....</b>	<b>7</b>
1.1	BAKGRUND OCH PROBLEMBESKRIVNING .....	7
1.1.1	<i>Familyfood</i> .....	8
1.2	SYFTE OCH FRÅGESTÄLLNINGAR .....	8
1.2.1	<i>Syfte</i> .....	8
1.2.2	<i>Frågeställning</i> .....	8
1.3	AVGRÄNSNINGAR .....	9
1.4	DISPOSITION .....	9
<b>2</b>	<b>Teoretisk bakgrund .....</b>	<b>10</b>
2.1	PDF .....	10
2.2	PHP .....	10
2.2.1	<i>Kodexempel</i> .....	11
2.2.2	<i>Fördelar med PHP</i> .....	11
2.3	DATABAS .....	12
2.3.1	<i>SQL</i> .....	13
2.3.1.1	SELECT .....	13
2.3.1.2	INSERT INTO .....	14
2.3.1.3	UPDATE .....	14
2.3.1.4	DELETE.....	14
2.3.1.5	JOIN.....	15
2.3.2	<i>MySQL</i> .....	15
2.3.2.1	<i>phpMyAdmin</i> .....	15
2.4	FORMULÄRDESIGN .....	16
2.4.1	<i>Etiketter</i> .....	16
2.4.2	<i>Inmatningsfält</i> .....	18
2.4.3	<i>Åtgärdsknappar</i> .....	19
2.4.4	<i>Misstag och fel</i> .....	19
<b>3</b>	<b>Metod och genomförande.....</b>	<b>20</b>
3.1	KRAVSPECIFIKATION .....	20
3.2	UTVECKLINGSMILJÖ .....	20
3.2.1	<i>Adobe Dreamweaver</i> .....	20
3.2.2	<i>phpMyAdmin</i> .....	21
3.2.3	<i>Webbplats</i> .....	21
3.3	TESTMILJÖ .....	21
3.4	PDF TILL DATABAS .....	22
3.5	PHP FÖR HANTERING AV DATABAS .....	23
3.5.1	<i>Lägga till recept</i> .....	23
3.5.2	<i>Hämta recept</i> .....	25
3.6	STRUKTUR AV DATABAS .....	26
3.6.1	<i>tbl_recipe</i> .....	26
3.6.2	<i>tbl_ingredient</i> .....	26
3.6.3	<i>tbl_ingredientDetails</i> .....	26
3.6.4	<i>tbl_category</i> .....	27
3.6.5	<i>tbl_stat</i> .....	27
3.6.6	<i>tbl_picture</i> .....	27
3.7	DESIGN AV FORMULÄR.....	28
3.7.1	<i>Receptsökning</i> .....	28
3.7.2	<i>Detaljerad receptsökning</i> .....	28
3.8	IMPLEMENTATION .....	29
<b>4</b>	<b>Resultat och analys .....</b>	<b>30</b>
4.1	SÖKFUNKTION.....	30

4.2	ADMINISTRATION .....	31
4.2.1	Lägg till receipt.....	31
4.2.2	Hantera receipt.....	32
4.2.3	Hantera kategori.....	33
4.2.4	Statistik .....	33
<b>5</b>	<b>Diskussion och slutsatser .....</b>	<b>34</b>
5.1	RESULTATDISKUSSION .....	34
5.1.1	Hur överför man med ett enkelt tillvägagångssätt information från en PDF-fil till en databas?.....	34
5.1.2	Hur skapar man användarvänliga formulär, ur synpunkt av interaktionsdesign? .....	34
5.2	SLUTSATSER OCH REKOMMENDATIONER .....	34
5.3	VIDAREUTVECKLING.....	35
<b>6</b>	<b>Referenser .....</b>	<b>36</b>
<b>7</b>	<b>Sökord.....</b>	<b>37</b>

## Figurförteckning

FIGUR 1 TABELL PERSON.....	12
FIGUR 2 TABELL ADDRESS.....	12
FIGUR 3 RAD PERSON.....	12
FIGUR 4 TOP-PLACERAD ETIKETT [8].....	16
FIGUR 5 HÖGERJUSTERAD ETIKETT [8].....	17
FIGUR 6 VÄNSTERJUSTERAD ETIKETT [8].....	17
FIGUR 7 ÅTGÄRDSKNAPP [8].....	19
FIGUR 8 TEXTFIL.....	22
FIGUR 9 STRUKTUR AV DATABAS.....	26
FIGUR 10 DESIGN RECEPSTÖKNING .....	28
FIGUR 11 DESIGN DETALJERAD RECEPSTÖKNING .....	28
FIGUR 12 TECKENKODNING .....	29
FIGUR 13 SÖKRESULTAT .....	30
FIGUR 14 VISA RECEPT .....	30
FIGUR 15 LÄGG TILL RECEPT DEL1 .....	31
FIGUR 16 LÄGG TILL RECEPT DEL2.....	31
FIGUR 17 HANTERA RECEPT .....	32
FIGUR 18 REDIGERA RECEPT .....	32
FIGUR 19 HANTERA KATEGORI .....	33
FIGUR 20 STATISTIK .....	33



# I Inledning

Denna rapport är skriven av en student på Tekniska Högskolan i Jönköping, med inriktning datateknik i samband med sitt examensarbete. Detta är en avslutande del av en treårig kandidatsexamen.

Uppdraget för examensarbetet var att skapa en webbapplikation med tillhörande receptdatabas där kunder enkelt skulle kunna söka och hitta recept. Detta med hjälp av en sökning- och filtreringsfunktion så att kunderna kan hitta precis det recept de vill hämta. Uppdraget består även i att hitta ett enkelt tillvägagångssätt för att överföra information från en PDF-fil till en databas.

Uppdragsgivare var företaget Familyfood, som är beläget i Jönköping. Familyfood leverera matkassar åt familjer, i största delen södra Sverige.

## I.1 Bakgrund och problembeskrivning

Examensarbetet hade sin begynnelse i september 2011 efter kontakt med Gustav Österström från Science Park i Jönköping. Han ville ha råd angående ett nytt projekt åt ett företag vid namn Familyfood. De levererar matkassar åt familjer runt om i Sverige. Efter samråd mellan Inger Palmgren, programansvarig för linjen datateknik på Tekniska Högskolan i Jönköping, och Gustav ansågs projektet vara ett potentiellt examensarbete.

Familyfoods problem är att de samlar på sig massvis av recept. Dessa recept lagras sedan i pappersform och i digitalform som PDF. För att göra det hanterligt behöver de lagra sina recept i en databas. Med hjälp av en webbapplikation ska de sedan t.ex. kunna filtrera på olika ingredienser eller söka på ett specifikt recept. Tanken är att detta skall vara en tjänst för dem själva men även för deras kunder. Det ska finnas en användarvänlig sök- och filtreringsfunktion för deras kunder på den hemsidan som de redan tillhandahar. Familyfood behöver även en webbapplikation för administration av databasen. Där man ska kunna skapa, ändra och radera recept enkelt genom att logga in via en dator som har anslutning till internet.

Examensarbetet kommer även bestå av att undersöka hur man på ett enkelt sätt kan överföra informationen från PDF-dokumentet, där recepten ligger samlade, till en databas.

### **1.1.1 Familyfood**

Enligt Karin Ahlqvist och Lotta Alberius på Familyfood levererar Familyfood sedan 2007 en färdigplanerad matkasse med recept. Ett koncept de själva har utvecklat. Sedan 2010 är även Tobias Gustavsson delägare med särskilt ansvar för logistik och inköp. Familyfood är ett av Sveriges första matkasseföretag och har specialiserat sig på att leverera utvalda råvaror från småskaliga producenter i och omkring sitt leveransområde.

I Familyfoods kassar kan man varje vecka hitta en stor variation av råvaror. I kassen finns råvaror och recept till 4 personer för 5 vardagsmiddagar. Varje leverans innehåller 25 till 30 varor, mycket råvaror från regionen och ekologiska produkter. Ekologiska varor produceras i samspel med naturen utan kemiska bekämpningsmedel, konstgödsel och utan att störa djurens naturliga beteenden. Kött och fågel är alltid svenskt. Vid starten levererades matkassen främst i och omkring Jönköping, men finns i dag i stora delar av södra Sverige.

Familyfood arbetar med egna inköp från en nybyggd packcentral utanför Växjö och leveranser sker därifrån ut till kund med kylbilar för att säkerställa kylkedjan. Kundantalet har växt med jämn takt och i dagsläget levererars mellan 500-700 matkassar i veckan. Familyfood erbjuder även tilläggstjänster såsom fruktkasse, fredagsmyskasse och nyrostat kaffe.

## **1.2 Syfte och frågeställningar**

### **1.2.1 Syfte**

Syftet med examensarbetet är att skapa en webbapplikation där Familyfood själva och deras kunder enkelt skall kunna hitta recept ur en databas.

Personligt syfte är förutom ovanstående syfte att även hitta ett enkelt tillvägagångssätt för att överföra information från Familyfoods PDF samling, där deras nuvarande recept ligger lagrade till den nyutvecklade databasen.

### **1.2.2 Frågeställning**

- Hur överför man med ett enkelt tillvägagångssätt information från en PDF-fil till en databas?
- Hur skapar man användarvänliga formulär, ur synpunkt av interaktionsdesign?

### 1.3 Avgränsningar

I ett tidigt skede av examensarbetet fastställdes avgränsningar. Avgränsningar är till för att undvika att arbetet inte blir för stort och håller sig inom tidsramen. I detta examensarbete finns tre stycken avgränsningar fastställda.

Den första avgränsningen är att webbapplikationen inte kommer utvecklas för olika typer av mobiler, däribland smartphones och deras tillhörande webbläsare. Webbapplikationen kommer endast att utvecklas för de vanligaste webbläsarna i PC-miljö. Detta kommer inte att påverka syftet för arbetet. Familyfood hade inget krav på att webbapplikationen skulle vara anpassad för mobiler.

Den andra avgränsningen är att webbapplikationen inte kommer utvecklas och optimeras för sökmotorer som t.ex. Google, Bing och Yahoo. Detta även kallat sökmotorsoptimering (SEO – Search Engine Optimization).

En tredje avgränsning är att ingen tyngd kommer läggas på design av webbapplikationen i form av CSS stilar.

### 1.4 Disposition

Denna rapport är disponerad enligt Tekniska Högskolan i Jönköpings mall för examensarbeten. Rapporten är disponerad enligt följande.

Efter inledningen följer *Teoretisk bakgrund*. Detta avsnitt tar upp den teori som har legat till grunden för att kunna genomföra examenarbetet. Bland annat redovisas programmerings- och databasspråk och dess olika tekniker. Efter slutförd läsning av detta kapitel har man tillräcklig kunskap för att kunna läsa och förstå nästakommande avsnitt.

Detta avsnitt heter *Metod och genomförande*. Här kan man få en inblick i hela genomförandet av examensarbetet.

Nästa avsnitt heter *Resultat och analys*. Här beskrivs och presenteras resultatet av examensarbetet. Även en redogörelse av hur resultatet motsvarar de mål som sattes upp för arbetet.

Sista avsnittet heter *Diskussion och slutsatser*. I detta avsnitt diskuteras arbetet av författaren. Allt från egna tankar eller olika problem som dök upp under arbetet till metodval.

## 2 Teoretisk bakgrund

Detta avsnitt tar upp den teori som har legat till grunden för att kunna genomföra examenarbetet. Avsnittet börjar med en kort teoretisk genomgång av filformatet PDF. Sedan följer en genomgång av programmeringsspråket PHP och dess koppling till databashanteraren MySQL och manipulationsspråket SQL. Till sist en genomgång av formulärdesign, hur man designar ett användarvänligt formulär ur synpunkt av interaktion mellan användare och formuläret.

Efter slutförd läsning av detta kapitel har man tillräcklig kunskap för att kunna läsa och förstå nästkommande avsnitt.

### 2.1 PDF

PDF är en förkortning av Portable Document Format och är ett unikt digitalt filformat, utvecklat för att kunna utbyta dokument mellan olika datorer och plattformar [1]. Exempel på plattformar där PDF-filer kan öppnas är operativsystem som: Microsoft Windows, Mac OS och mobila operativsystem som Android [2].

Ett exempel på styrkan och fördelen hos PDF. Tänk dig följande scenario; ett dokument konverteras till PDF och är skapat via en dator med plattformen Microsoft Windows. PDF-filen öppnas och visas sedan med samma resultat på en dator som kör Mac OS. Hade inte dokumentet konverterats till PDF och sedan öppnats på datorn med Mac OS, är det inte säkert att t.ex. layouten i dokumentet hade visat samma resultat.

PDF är utvecklat av Adobe System som även har utvecklat det vanligaste programmet för att öppna PDF-filer, Adobe Acrobat.

### 2.2 PHP

PHP står för PHP: Hypertext Preprocessor och är ett programmeringsspråk för att skapa dynamiska webbplatser på internet. Språket är utvecklat av Rasmus Lerdorf med sin början år 1994. Som namnet anger bearbetas PHP-koden och skapas i form av hypertext, även kallat HTML. PHP är ett skriptspråk som bearbetas och körs direkt på webbservern, ett så kallat "server-side" språk.

Exempel på en PHP process från början till slut.

- En användare surfar in på en webbplats, genom att t.ex. mata in en adress i webbläsarens adressfält.
- Webbservern till webbplatsen instruerar PHP-motorn på servern att bearbeta och köra PHP-skriptet på sidan.
- När skriptet har körts och är klart sänds den bearbetade HTML-sidan till webbläsaren. Alltså skickas ingen PHP-kod till webbläsaren utan den bearbetas på webbservern och skickas sedan som hypertext (HTML) [3].

### 2.2.1 Kodexempel

Nedan är ett exempel på en PHP-fil där PHP-scriptet skickar tillbaka texten *Hej Världen* till webbläsaren.

```
<?php $variabel = 'Hej Världen'; ?>
<html>
<head></head>
<body>

<?php
    echo $variabel;
?>
</body>
</html>
```

Kodexempel 1 PHP

Resultat av kodexempel 1:

Hej Världen

Som man kan se i kodexempel 1 är PHP-koden inbäddat mellan HTML-taggarna. PHP-kod kan förekomma vartsomhelst i filen. För att deklarerera för servern vart PHP-koden börjar och slutar används `<?php` som starttagg och `?>` som sluttagg. Vissa servrar kräver att man skriver `<?php`, för andra räcker bara `<? [3]`.

### 2.2.2 Fördelar med PHP

En stor fördel med att använda sig av PHP är att många webbhotell stödjer språket på deras servrar. Förutom det kan PHP även köras på många olika plattformar; t.ex. Windows, Linux, Mac OS X [3].

## 2.3 Databas

En databas består av en samling data som är organiserad och strukturerad i tabeller för att skapa användbar information. Informationen är sedan enkel att hämta och läsa från databasen.

Person	
Person_nr	PK
Förnamn	
Efternamn	

Figur 1 Tabell Person

Adress	
AdressID	PK
Person_nr	FK
Adress	
Postnr	
Stad	

Figur 2 Tabell Adress

Strukturen av en tabell i en databas består av kolumner (attribut) och rader. Nedan är ett exempel på en rad i en databas med data. Strukturen på tabellen är samma som i figur 1.

Kolumn		
Person_nr	Förnamn	Efternamn
987654-3217	Kalle	Svensson

Rad

Figur 3 Rad Person

Det finns olika typer av databaser, t.ex. relationsdatabas och objektorienterad databas. I detta examensarbete kommer databasen vara en relationsdatabas. Fördelen med en relationsdatabas är att man kan koppla ihop och skapa relationer mellan tabellerna i databasen. Fördelen med detta är att man genom en SQL-fråga kan t.ex. få reda på vilken adress en viss person bor på genom att veta förnamn och efternamn [4]. Se figur 1 och 2.

För att man ska kunna koppla ihop två tabeller med varandra behövs två olika sorters nycklar. En primärnyckel och en främmande nyckel.

En primärnyckel (PK) identifierar unikt varje rad i en tabell. Varje rad måste ha en egen unik nyckel. Primärnyckeln skall vara en kolumn i en tabell och dess värden får aldrig ändras. I figur 1 är kolumnen *Person\_nr* en primärnyckel för tabellen *Person*. Personnummer är ett bra exempel på en PK, eftersom dess värde aldrig kommer att ändras. En person kommer alltid ha samma personnummer hela livet. Ett sämre exempel hade varit efternamn. En person kan under sin livstid ändra sitt efternamn och flera personer kan ha samma efternamn. Ett annat alternativ är att använda sig av en nyckel vars värde är ett heltal som automatiskt ökar när en ny rad läggs till. En primärnyckel får aldrig vara ett tomt fält.

En främmande nyckel (FK) motsvarar en primärnyckeln i en annan tabell och behövs för att kunna koppla ihop två tabeller med varandra. I figur 2 är *Person\_nr* en främmande nyckel. Dess motsvarighet är primärnyckeln i figur 1 [4].

### 2.3.1 SQL

SQL är ett språk för att kunna komma åt och manipulera data i en databas. Detta görs med hjälp av en SQL-fråga. Det finns ett flertal olika kommandon man kan ange i en fråga för att filtrera fram det resultat man vill ha. Det finns fyra stycken huvudkommandon; SELECT, INSERT INTO, UPDATE och DELETE.

#### 2.3.1.1 SELECT

SELECT-kommandot används för att välja data som skall hämtas från en databas. Exempel. Hämta all data från tabell *Person* i Figur 1.

```
SELECT *  
FROM Person  
WHERE Efternamn = 'Andersson'
```

Kodexempel 2 Select

SELECT anger vilka kolumner som skall hämtas. \* anger att alla kolumner skall hämtas från tabellen. FROM anger vilken tabell data skall hämtas ifrån. WHERE begränsar antal rader som skall hämtas utifrån ett angett kriterium. I detta fall hämtas endast de rader vars efternamn är Andersson [5].

### 2.3.1.2 INSERT INTO

Som namnet avslöjar används INSERT INTO-kommandot för att lägga till en ny rad i en tabell som finns i en databas.

Exempel. Lägg till ny rad i tabellen *Person* i Figur 1.

```
INSERT INTO Adress  
VALUES (1, 'Storgatan 5', '123 45', 'Jönköping')
```

Kodexempel 3 Insert Into

INSERT INTO anger i vilken tabell raden skall läggas till. VALUES anger de värden som skall matas in i kolumnerna i den nya raden. I detta fall är AdressID primärnyckeln för tabellen och ökar automatiskt sitt värde varje gång en ny rad läggs till. Därför anges inte detta värde. Värden skrivs i den ordning som kolumnerna är strukturerad. Är värdet en textsträng anges citationstecken i början och i slutet.

### 2.3.1.3 UPDATE

För att ändra/uppdatera en rad i en databas används UPDATE-kommandot.

```
UPDATE Person  
SET Efternamn ='Andersson'  
WHERE Person_nr = 987654-3217
```

Kodexempel 4 Update

UPDATE anger i vilken tabell som rad/rader skall uppdateras. SET anger för vilka kolumner som ska uppdateras. Här är det viktigt att använda sig av WHERE i sin SQL-fråga. Med hjälp av WHERE specificerar man vilken rad/rader man vill uppdatera. Hade man i ovanstående exempel tagit bort WHERE i SQL-frågan hade alla rader i tabellen uppdaterats och ändrat efternamn till Andersson [5].

### 2.3.1.4 DELETE

Det kan vara användbart att kunna ta bort en rad i en databas. För detta används DELETE-kommandot.

```
DELETE FROM Person  
WHERE Person_nr = 987654-3217
```

Kodexempel 5 Delete

DELETE FROM anger i vilken tabell en rad skall tas bort. Här är det lika viktigt att använda WHERE i sin SQL-fråga som när man använder sig av UPDATE. Utan WHERE skulle man ta bort alla rader i hela tabellen [5].



### 2.3.1.5 JOIN

För att få fram data från två eller flera tabeller som är kopplade, används JOIN. Det finns fyra olika typer som används för olika typer av syften; Inner Join, Left Join, Right Join och Full Join. I detta exempel kommer INNER JOIN att användas. Inner Join visar de rader där det finns minst en matchning i båda tabellerna.

```
SELECT Person.Förnamn, Person.Efternamn, Adress.Stad
FROM Person
INNER JOIN Stad
ON Person.Person_nr = Adress.Person_nr
```

Kodexempel 6 Inner Join

Med hjälp av INNER JOIN anges vilka tabeller som skall användas. I SQL-frågan används en primärnyckel och en främmande nyckel för att koppla ihop tabellerna. ON anger de nycklar som finns i tabellerna [5].

### 2.3.2 MySQL

MySQL som bildades och till sin början utvecklades av två svenskar och en finländare, släppte sin första version 1995. MySQL är ett gratis alternativ för databashantering och är ett program med en öppen källkod. MySQL kan integreras med ett antal olika programmeringsspråk, däribland PHP för att skapa dynamiska hemsidor. Fördelen med MySQL är, förutom att det är enkelt att installera, dess användarvänlighet. Om man använder MySQL som databashanterare är det möjligt att använda sig av SQL-frågor [6].

#### 2.3.2.1 phpMyAdmin

phpMyAdmin är ett populärt webbinterface för att via webben administrera en MySQL-databas. Programmet är skrivet i PHP och finns gratis att ladda ner på internet. Många webbhotell väljer att installera och tillhandahålla phpMyAdmin på deras servrar [7].

## 2.4 Formulärdesign

Ett formulär har i regel tre olika element. Dessa är: etiketter, inmatningsfält och åtgärdsknappar. Etikett är en kortfattad fråga som ställs till användaren. Utan en etikett skulle man inte veta vilket svar man skall ange i inmatningsfältet. Ett inmatningsfält är till för att användaren skall kunna svara på den fråga som anges i etiketten. Med åtgärdsknappen slutförs formuläret och värden skickas iväg.

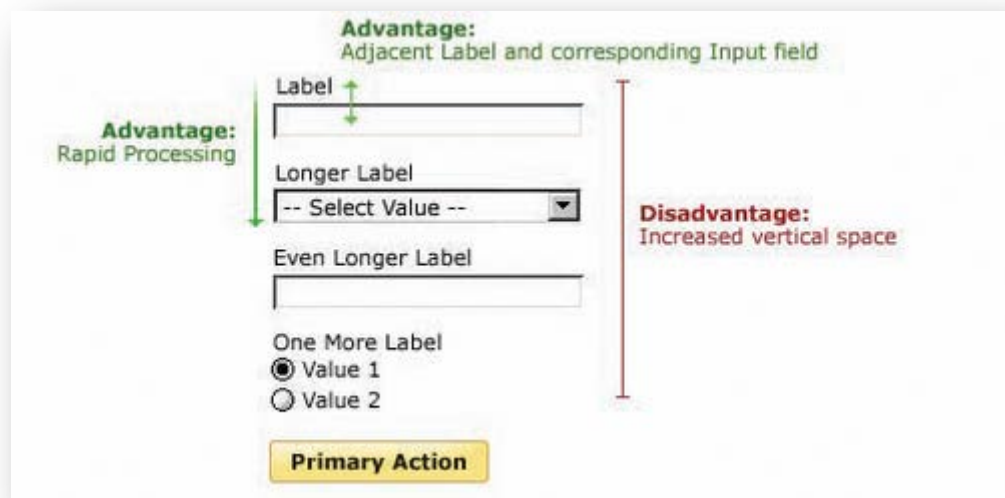
Formulärdesign är viktig för att skapa en god användarvänlighet och en bra interaktion mellan formuläret och användaren. Men hur designar och skapar man ett bra formulär? Nedan följer några riktlinjer som kan följa [8].

### 2.4.1 Etiketter

En etikett skall vara kort och av ett naturligt språk. Det finns tre olika sätt att placera en etikett i ett formulär.

- **Top-placerad etikett**

Med top-placerad etikett är det enkelt för användaren ta sig igenom fältet. Etiketten ligger över inmatningsfältet. Det ligger även nära inmatningsfältet och man läser i en riktning, uppifrån och ner. Nackdelen är att det tar vertikalt mycket plats [8]. Se figur 4.



Figur 4 Top-placerad etikett [8]

- **Högerjusterad etikett**

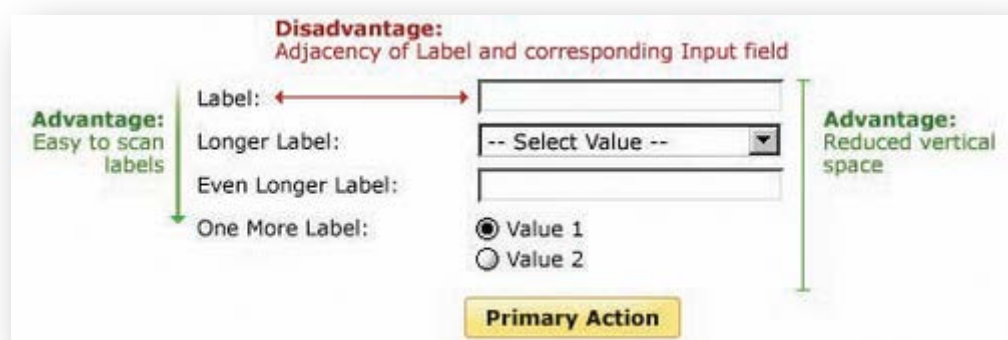
Högerjusterad etikett har samma fördel som top-placerad etikett, den ligger nära inmatningsfältet. En annan fördel med detta alternativ är att utrymmet reduceras vertikalt. Nackdelen är att det är svårt att läsa etiketterna. Speciellt för användare i västvärlden som läser från vänster till höger [8]. Se figur 5.



Figur 5 Högerjusterad etikett [8]

- **Vänsterjusterad etikett**

Med vänsterjusterad etikett är det enkelt att läsa etiketten. Detta alternativ har samma fördel som med högerjusterad att det reducerar utrymmet vertikalt. Nackdelen är att det kan vara svårt att urskilja vilken etikett som hör till vilket inmatningsfält, eftersom etiketten ibland ligger långt ifrån inmatningsfältet [8]. Se figur 6.



Figur 6 Vänsterjusterad etikett [8]

De tre alternativen har alla olika fördelar och nackdelar. Men det är att föredra att använda top-placerad etikett när etiketterna behöver ha olika längd och man vill ha ett formulär som går snabbt för användaren att slutföra. Högerjusterad etikett och vänsterjusterad etikett är bra att använda när det finns ett begränsat utrymme. Ifall användaren enkelt ska kunna läsa igenom etiketterna för att få en snabb överblick över vilka frågor som skall besvaras, skall vänsterjusterad etikett användas [8].

### **2.4.2 Inmatningsfält**

För att kunna besvara de frågor användaren har läst igenom i form av etiketter används inmatningsfält. Det finns olika typer av inmatningsfält; textfält, kryssrutor, radioknappar och rullgardinsmenyer.

När man designar ett formulär gäller det att välja rätt typ av inmatningsfält till rätt typ av fråga. T.ex. Ska svaret vara antingen ja eller nej, rekommenderas att använda radioknappar.

Man ska försöka undvika inmatningsfält som är valfria. Om man använder en asterisk \* för att visa vilka fält som är obligatoriska så måste man någonstans skriva vad asterisken betyder. En del användare kan uppfatta att fältet istället är valfritt.

Används textfält i formuläret skall dessa vara tillräckligt långa för att användaren ska få utrymme för ett svar [8].

### 2.4.3 Åtgärdsknappar

När användaren har läst igenom etiketterna och svarat på frågorna kan en åtgärdsknapp användas för att slutföra formuläret. Oftast innehåller ett formulär fler än en åtgärdsknapp. Vanligtvis en primär och en sekundär. Exempel på en primär åtgärdsknapp kan vara; Spara och Skicka. En sekundär åtgärdsknapp kan vara; Tillbaka och Avbryt.

För att skapa ett bra formulär bör man undvika sekundära åtgärdsknappar. Ifall man ändå behöver använda sekundära åtgärdsknappar bör man se till att det är en tydlig skillnad mellan de primära och sekundära. Se figur 7.

Använder man sig av sekundära åtgärdsknappar som ”Ta bort” eller ”Rensa” måste man alltid ge användaren ett ytterligare alternativ att antingen ändra sig eller fortsätta [8].



Figur 7 Åtgärdsknapp [8]

### 2.4.4 Misstag och fel

Fastän att man lägger ner mycket tid på att designa ett bra formulär finns det risker för att användare gör misstag. Det kan vara förödande. Därför måste man förebygga och överväga de fel och misstag som kan inträffa. Detta gör man genom att kontrollera alla fält i formuläret och skriva ut felmeddelande ifall något fel inträffat. Med ett felmeddelande kan användaren enkelt och snabbt lösa felet. Genom att t.ex. fylla i ett obligatoriskt textfält som var tomt. Röd text är att föredra när man skriver ut felmeddelande eller varningar i ett formulär [8].

## 3 Metod och genomförande

I detta avsnitt får man en inblick i hur genomförandet av examensarbetet gick till och vilka metoder som har använts. Första delen går igenom kravspecifikation för examensarbetet, utvecklingsmiljöer och testmiljöer. Sedan följer en genomgång av vilken metod som användes för att lösa problemet att överföra information från en PDF-fil till en databas. Efter det en genomgång av den programmering som ligger till grund för att kunna hantera databasen via en webbapplikation och en genomgång av databasens struktur. Till sist en genomgång av tankesätt vid formulärdesign och en genomgång av implementationen.

### 3.1 Kravspecifikation

Tillsammans med Familyfood togs en kravspecifikation fram för examensarbetet.

- I databasen skall titeln på varje recept vara unik. Det får inte finnas dubletter av samma titel.
- Varje recept skall ha en URL-länk, så att man enkelt skall kunna dela med sig av ett recept genom att kopiera länken.
- Databasen skall innehålla en kategori för varje recept.
- Det skall finnas möjlighet att kunna ladda upp en bild till varje recept.
- Någon form av statistik skall även finnas för att kunna läsa av hur många besökare det har varit på varje recept.

### 3.2 Utvecklingsmiljö

#### 3.2.1 Adobe Dreamweaver

Adobe Dreamweaver är ett redigeringsprogram för att underlätta vid skapande och redigering av webbplatser. Programmet har flera fördelar som underlättar när man utvecklar nya webbplatser. Ett av dessa är att Dreamweaver har ett designläge och ett kodningsläge. I designläget kan man utveckla och designa sin hemsida utan att behöva se koden. Koden skapas automatiskt i bakgrunden av Dreamweaver. I kodningsläget kan man skriva direkt i koden och man får en bra överblick p.g.a. att koden är färglagd i olika färger för att underlätta när man skall läsa koden.

Detta program har varit mycket användbart vid studiet för programmering och design av webbapplikationen. Programmet är enkelt att arbeta med, man sparar mycket tid och det finns olika verktyg för att felsöka fel i sin kod.

### 3.2.2 phpMyAdmin

Som det angavs i avsnittet *Teoretisk bakgrund* är phpMyAdmin ett populärt webbinterface för att administrera en MySQL-databas. phpMyAdmin är mycket användbart vid utveckling. Man kan enkelt via webbinterfacet skapa nya tabeller och dess struktur. Det är även enkelt att lägga till, ta bort och uppdatera rader i de tabeller som finns.

phpMyAdmin har varit mycket användbart i detta examensarbete. Med hjälp av applikationen har det varit enkelt att skapa databasstrukturen och tabellernas kopplingar mellan varandra. Det är även enkelt och snabbt att kunna hantera den information som finns i databasen.

### 3.2.3 Webbplats

Familyfood har redan en utvecklad webbplats, ett eget domännamn och utrymme på en webbserver. Webbplatsen är utvecklad i WordPress som är ett populärt och gratis innehållshanteringssystem, även kallat CMS (Content Management System). När webbapplikationen är klar och redo för att läggas ut i skarpt läge, kommer den att integreras och läggas upp på Familyfoods webbplats.

## 3.3 Testmiljö

Förutom en bra utvecklingsmiljö krävs även att man tillhandahåller en bra testmiljö. Vid utveckling och testning av webbapplikationen är en testmiljö att föredra. En dålig idé är att köra i skarpt läge på t.ex. en kunds server. Därmed används i detta examensarbete en webbserver med MySQL och Apache installerat. Apache är en fri programvara för att kunna köra en http-server.

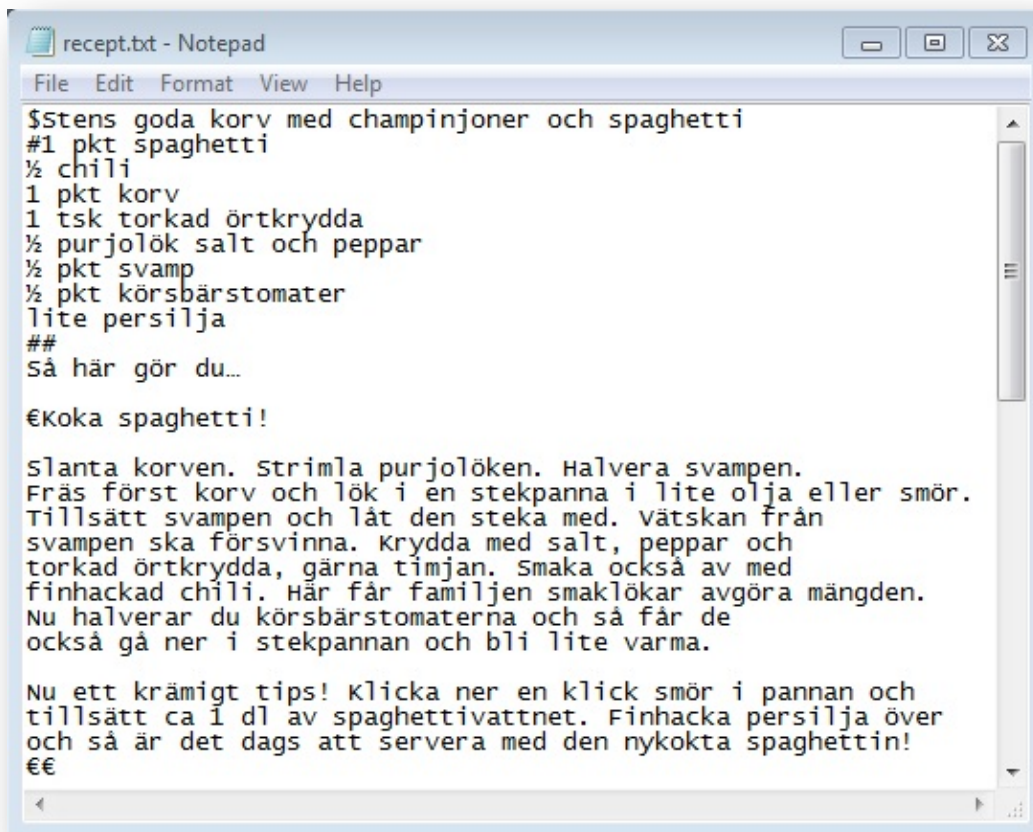
### 3.4 PDF till databas

Som man kan läsa om i inledningen, består examensarbetet i att hitta ett tillvägagångssätt för att överföra informationen från en PDF-fil till en databas.

Den metod som användes var att läsa facklitteratur och söka information på internet. Enligt boken *Adobe Acrobat 9 PDF Bible* kan man exportera en PDF-fil till ett flertal andra format. T.ex. XML-, HTML- och textfil. [9]

Efter en hel del testande och resonering valdes textfil som det bästa formatet att arbeta med. Detta pga. att det är enkelt att läsa in en textfil med PHP, det programmeringsspråket som används i examensarbetet. De andra filformaten hade ingen fördel jämfört med textfil. Det gick inte att utvinna vad som var vad i filen. Det fanns t.ex. inga speciella taggar i XML-filen som berättade vad som var titel, ingrediens eller beskrivning.

Nästa problem som behövdes lösas är hur man i textfilen kan skilja på vad som är vad. Hur ska datorn förstå vad som är en titel, ingrediens eller beskrivning i textfilen? Man måste på något sätt tala om för datorn vad som är vad. Detta löstes genom att man manuellt får mata in tecken i textfilen som indikerar de olika delarna. En titel börjar med ett dollar-tecken (\$). Ingredienser börjar med en fyrkant (#), och avslutas med två. Beskrivningen fungerar likadant som ingredienserna fast använder istället ett euro-tecken (€). Se figur 8.



```
recept.txt - Notepad
File Edit Format View Help
$Stens goda korv med champinjoner och spaghetti
#1 pkt spaghetti
½ chili
1 pkt korv
1 tsk torkad örtekrydda
½ purjolök salt och peppar
½ pkt svamp
½ pkt körsbärstomater
lite persilja
##
så här gör du...

€koka spaghetti!

slanta korven. strimla purjolöken. Halvera svampen.
Fräs först korv och lök i en stekpanna i lite olja eller smör.
Tillsätt svampen och låt den steka med. Vätskan från
svampen ska försvinna. Krydda med salt, peppar och
torkad örtekrydda, gärna timjan. Smaka också av med
finhackad chili. Här får familjen smaklökar avgöra mängden.
Nu halverar du körsbärstomaterna och så får de
också gå ner i stekpannan och bli lite varma.

Nu ett krämigt tips! Klicka ner en klick smör i pannan och
tillsätt ca 1 dl av spaghettivattnet. Finhacka persilja över
och så är det dags att servera med den nykokta spaghettin!
€€
```

Figur 8 Textfil



Med hjälp av PHP-kod kan man sedan läsa rad för rad i textfilen. Beroende på vilket tecken som raden börjar på, sparas radens data i olika variabler. Se kodexempel 7.

```
$textfile = fopen("recept.txt", "rb");
while (!feof($textfile) ) {
    $row = fgets($textfile);
    if ($row[0] == '$') {
        $row = substr($row, 1);
        $titel = $row;
    }
}
fclose($textfile);
```

Kodexempel 7

Med detta tillvägagångssätt sparar man mycket tid jämfört med att klippa och klistra in data från en PDF till ett formulär på en webbplats.

## 3.5 PHP för hantering av databas

För att en användare utan kunskap av programmering och databaser skall kunna visa, lägga till, uppdatera och ta bort information från en databas krävs ett användarvänligt webbinterface med kod som körs i bakgrunden. För att skapa detta krävs en hel del programmering och det finns många olika faktorer man måste ta hänsyn till. I detta examensarbete används programmeringsspråket PHP för detta ändamål.

### 3.5.1 Lägg till recept

Via inloggning på Familyfoods hemsida skall man kunna administrera databasen. Bland annat skall man kunna lägga till nya recept via ett formulär. I formuläret skall man enkelt kunna mata in receptets titel, ingredienser, beskrivning, kategori och ladda upp en bild. När sedan användaren är färdig och trycker på spara skall PHP-kod köras igång och mata in all data i databasen. I kodexempel 8 är ett exempel på PHP-kod som lägger till en ny rad i databasen. I detta fall är det en ny rad i tabellen *tbl\_recipe*. Data hämtas från variabler som i sin tur har hämtat sin data från det skickade formuläret. Informationen skickas från formuläret med hjälp av en metod som heter POST.

```
$sql_query = "INSERT INTO tbl_recipe (title,description,categoryID)
              VALUES('$title','$description','$category')";
if (!mysql_query($sql_query))
{
    die('Error: ' . mysql_error());
} else {
    echo 'Receptet är nu tillagt';
}
```

Kodexempel 8

I kravspecifikationen anges att inte flera recept får ha samma titel. För att lösa detta används PHP-kod som kollar igenom databasen och jämför ifall det finns en liknande titel som den inskickade från formuläret.

```
$numrows_recipe = 0;
$title = trim($title);
$numrows_recipe = mysql_num_rows(mysql_query("SELECT title FROM
tbl_recipe WHERE title LIKE '$title%'"));
if ($numrows_recipe == 0)
{
}
}
```

Kodexempel 9

När sedan ingredienserna skall matas in i databasen måste det göras en kontroll ifall ingrediensen redan finns, eller det behövs läggas till en ny rad. När kontrollen är genomförd skall en koppling skapas mellan ingrediensen och receptet. Se figur 9.

Med liknande kod som i kodexempel 9 så görs en kontroll ifall en liknande ingrediens finns i databasen, i tabellen *tbl\_ingredientDetails*. I så fall hämtas ingrediensens id. Annars läggs en ny rad till i tabellen. För att hämta dess id används funktionen `mysql_insert_id()`. Denna funktion hämtar id från den sista raden som lagts till.

Det sista steget är att skapa kopplingen mellan receptet och ingrediensen. Detta görs genom att lägga till nya rader i tabellen *tbl\_ingredient*. Se figur 9. Med hjälp av föregående steg i processen har receptets id och ingrediensernas id sparats i variabler och kan enkelt matas in i tabellen. Detta görs likadant som i kodexempel 8 förutom att SQL-frågan är annorlunda utformad.

### 3.5.2 Hämta recept

På Familyfoods hemsida skall deras kunder och dem själva kunna söka efter recept och sedan skickas vidare till en sida med resultatet av deras sökning. För att hämta ett recept och skriva ut dess titel på en webbsida används PHP-kod som i kodexempel 10. En variabel hämtar informationen från det skickade formuläret. Sedan med hjälp av en SQL-fråga hämtas alla de rader som innehåller ordet eller den mening som är skickad. Efter det skrivs alla rader ut. I detta fall innehåller varje rad en titel och en länk som skickar med dess id.

```
$title = $_POST['title'];
$sql_query = mysql_query("SELECT recipeID,title FROM tbl_recipe
                          WHERE title LIKE '%" . $title . "%' ") or die();
while ($row = mysql_fetch_array($sql_query))
{
    echo '<p>
    <a href="show_db.php?title=id=' . $row['recipeID'] . '">' . $row['title'] . '</a>
    </p>';
}
```

Kodexempel 10

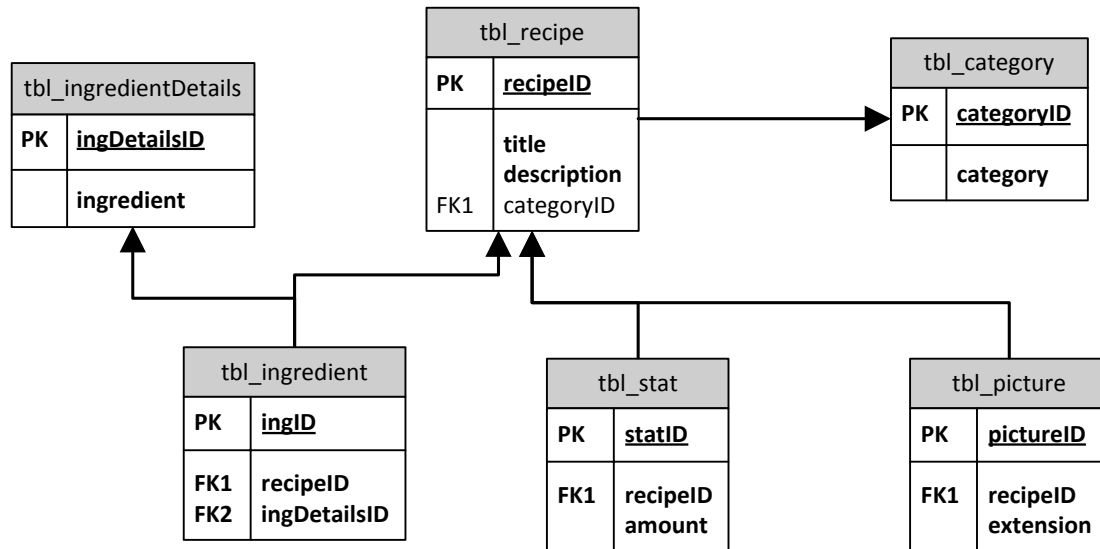
Användaren skall nu kunna klicka på ett recept och skickas vidare till en ny sida där information om det specifika receptet visas. För att veta vilket recept som användaren har klickat på skickas ett id med i länken. Som sedan kan hämtas med hjälp av GET metoden, som är en annan metod för att skicka med data. Receptet skrivs ut med hjälp av liknande PHP-kod som i kodexempel 10. Däremot behövs en längre och mer avancerad SQL-fråga. Se kodexempel 11.

```
$sql_query = mysql_query("SELECT tbl_recipe.recipeID AS ID, tbl_recipe.title
                          AS title, tbl_recipe.description AS description
                          FROM tbl_recipe, tbl_ingredient, tbl_ingredientdetails
                          WHERE tbl_recipe.recipeID = tbl_ingredient.recipeID
                          AND tbl_ingredient.ingDetailsID =
                          tbl_ingredientdetails.ingDetailsID
                          AND tbl_recipe.recipeID = ".$id."
                          GROUP BY tbl_recipe.recipeID") or die();
```

Kodexempel 11

### 3.6 Struktur av databas

För att kunna spara data och omvandla den till meningsfull information krävs en databas med olika tabeller och attribut, en struktur av databasen. Det gäller att man tänker igenom strukturen, t.ex. vilken slags data som skall sparas och vilka kopplingar som skall finnas mellan tabellerna. I figur 9 kan du se strukturen för databasen för detta examensarbete. Nedan följer sedan en kort förklaring av varje tabell.



Figur 9 Struktur av databas

#### 3.6.1 tbl\_recipe

I tabellen *tbl\_recipe* lagras receptets titel och beskrivning. Denna tabell är navet i strukturen och har på ett eller annat sätt en koppling till alla tabeller.

#### 3.6.2 tbl\_ingredient

*tbl\_ingredient* är en kopplingstabell för att koppla ihop tabellen *tbl\_recipe* och *tbl\_ingredientDetails*. Tabellen innehåller tre kolumner, en primärnyckel och två främmande nycklar. Dessa kolumners värden kommer endast vara ett heltal.

#### 3.6.3 tbl\_ingredientDetails

Ingredienserna lagras i tabellen *tbl\_ingredientDetails*. Varje ingrediens är unik i tabellen och det kan inte finnas några dubletter.

### 3.6.4 **tbl\_category**

För att kunna kategorisera ett recept används *tbl\_category*. Här sparas ett ID-nummer och ett namn på kategorin. I *tbl\_recipe* sparas ett värde på kategorins id. Därmed kan man sedan läsa från databasen vilken kategori ett recept har.

### 3.6.5 **tbl\_stat**

I kravspecifikationen var ett krav att antal besökare på varje recept skulle registreras. I tabellen *tbl\_stat* sparas antal besökare i kolumnen *amount*.

### 3.6.6 **tbl\_picture**

Ett annat krav i specifikationen var även att kunna ladda upp bilder till receptet. Bildens id och filtyp sparas i tabellen *tbl\_picture*. Med hjälp av denna information skall man sedan kunna hämta den bild som är kopplad till receptet.

## 3.7 Design av formulär

För att en användare snabbt och enkelt skall kunna fylla i ett formulär med relevant data och hitta de recept användaren söker efter, krävs en genomtänkt formulärdesign. I detta examensarbete utformades formulärdesignen utifrån Luke Wroblewskis bok *Web Form Design* [8]. Detta kan man läsa mer om i avsnitt 2.4 *Formulärdesign*.

I detta avsnitt görs en generell genomgång av tänkesättet när formulären designades med hjälp av ett par exempel.

### Receptsök

Titel:

[Detaljerad sökning >>](#)

Figur 10 Design receptsökning

### Receptsökning

Titel:

Ingrediens:

Kategori:  ▾

Figur 11 Design detaljerad receptsökning

### 3.7.1 Receptsökning

På Familyfoods hemsida är tanken att det skall ligga ett enkelt sökformulär till höger i designen som kommer finnas tillgänglig på alla sidor. Designen för detta formulär är designat med vänsterjusterad etikett, för att spara så mycket utrymme vertikalt som är möjligt. Eftersom det endast finns ett textfält så behöver man inte tänka på att etiketterna kommer bli olika långa. Utifrån detta valdes vänsterjusterad etikett. Se figur 10.

### 3.7.2 Detaljerad receptsökning

För att användaren skall kunna göra en detaljerad sökning behövs ett annat formulär. Detta formulär innehåller tre olika typer av inmatningsfält. Ett för titel, ingrediens och kategori. Eftersom etiketterna har olika längd är det att föredra top-placerad etikett. Med top-placerad etikett är det snabbt och enkelt för användaren att slutföra formuläret fort. I detta fall är inte vertikalt utrymme ett bekymmer. Därmed valdes top-placerad etikett. Se figur 11.

### 3.8 Implementation

Fram till implementering av webbapplikationen och receptdatabasen på Familyfoods webbplats har webbapplikationen körts och testats i en testmiljö. Detta kan man läsa mer om i avsnitt 3.3.

När väl webbapplikationen var redo för implementering kontaktades Peter Püschel, som har utvecklat den webbplats Familyfoods tillhandahåller i nuläget. Med hjälp av honom kunde implementering ske snabbare och smidigare eftersom han redan är insatt i hur webbplatsen är uppbyggd. Familyfoods webbplats är utvecklat med hjälp av WordPress. WordPress kan man läsa mer om i avsnitt 3.2.3.

Sökfunktionen med tillhörande sidor för att visa recept skapades som ett tillägg i WordPress. Detta tillägg kan sedan anropas på en sida och enkelt använda dess funktioner. Att göra på detta sätt är det bästa och snyggaste sättet, men inte det enklaste. Det krävdes en del ändring i den koden som var skriven. Arbete för ändring av koden utfördes av Peter Püschel.

Administrationen av receptdatabasen lades in i Familyfood Center som är en separat applikation på Familyfoods hemsida. Familyfood Center används för hantering av information som rör Familyfoods verksamhet, t.ex. kundinformation. Administrationen och dess olika sidor lades till i Familyfood Center som en iframe. En iframe är en html-tag som placeras på en sida som sedan i sin tur visar en annan sida. Kort sagt infogas i iframen en separat sida. Detta är det enklaste tillvägagångssättet. Med hjälp av detta slipper man ändra något i koden.

Ett stort problem under implementeringen var de svenska bokstäverna å, ä och ö. Vid utveckling av webbplatser anges meta-taggar i huvudet av html-dokumentet. I en av dessa meta-taggar anger man vilken teckenkodning som skall användas. Denna teckenkodning skall stämma överens med den som används i databasen. På Familyfoods webbplats används UTF-8 som teckenkodning. Webbapplikationen som har skapats i detta examensarbete är däremot utvecklat med den västerländska teckenkodningen iso-8859-1. Detta skapade problem under implementeringen. Istället för att skriva ut å, ä och ö skrevs ett annat tecken ut som ser ut som en trekant med ett frågetecken i. Se figur 12.



1 dl vatten  
1 pose sallad  
1 dl ströbröd  
ca 500 g lammfärs  
1 rödlök  
lite timjan  
4 tomater  
salt och peppar  
4 bakpotatisar

Figur 12 Teckenkodning

Detta problem löstes genom att ändra så att alla sidor använder samma teckenkodning. Det ända alternativet var att använda UTF-8 eftersom Familyfoods hemsida är utvecklat utifrån det.

## 4 Resultat och analys

I detta avsnitt beskrivs resultatet av examensarbetet. Här beskrivs hur slutresultatet av webbapplikationen blev, med sökfunktion och administration.

### 4.1 Sökfunktion

Som angetts tidigare skall det finnas ett enkelt sökformulär till höger i designen på Familyfoods hemsida som kommer finnas tillgänglig på alla sidor. Se figur 10. Det skall även finnas en sida för detaljerad sökning. Se figur 11.

När besökaren angett sökord i något av formulären skickas den vidare till en resultatsida, se figur 13. Här visas de träffar som stämmer överens med sökorden. Endast receptens titel visas. Ifall det blir många träffar används en pagineringsfunktion där man kan bläddra mellan sex recept i taget.

Besökaren kan sedan klicka på en av titlarna och sedan skickas vidare till en ny sida där det utvalda receptet visas. Se figur 14. Här visas receptets titel, ingredienser, beskrivning, eventuell kategori och bild. Ifall det inte finns någon bild knuten till receptet visas en default bild som är Familyfoods logotype.



Figur 13 Sökresultat



Figur 14 Visa recept



## 4.2 Administration

Via inloggning på Familyfoods hemsida kan man som användare komma åt webbapplikationen för att kunna administrera receptdatabasen.

### 4.2.1 Lägg till recept

För att kunna lägga till ett nytt recept går användaren igenom två steg. I det första steget matas receptets titel, ingredienser, beskrivning och kategori in. Men först av allt måste en kontroll göras som kontrollerar ifall det finns en titel i databasen som stämmer överens med den användaren matat in. Detta är ett viktigt steg eftersom alla titlar i databasen skall vara unika. Se figur 15.

#### Lägg till recept

Steg 1 av 2

The screenshot shows a form titled 'Lägg till recept' at 'Steg 1 av 2'. There is a single input field labeled 'Titel' with a 'Check' button to its right.

Figur 15 Lägg till recept del1

Ifall kontrollen blir godkänd kommer användaren vidare och kan börja fylla i resterande inmatningsfält. Här skall användaren på ett användarvänligt sätt kunna mata in flera ingredienser. Detta löstes genom att användaren trycker på knappen *Lägg till*. Då läggs ett nytt inmatningsfält till som användaren sedan också kan ta bort.

Användaren är tvungen att fylla i alla fält förutom kategori. Ifall ett obligatoriskt fält inte är ifyllt kan användaren inte spara och de icke ifyllda fälten får en röd ram som indikerar för användaren vilka fält som måste fyllas i för att komma vidare. Se figur 16.

#### Lägg till recept

Steg 1 av 2

The screenshot shows the 'Lägg till recept' form at 'Steg 1 av 2'. The 'Titel' field contains 'En unik titel' and has a green border. The 'Ingreienser' field has a red border. The 'Beskrivning' field is empty. The 'Kategori' field is a dropdown menu with '-Välj kategori-' selected. A 'Spara' button is at the bottom.

Figur 16 Lägg till recept del2

## 4.2.2 Hantera recept

En väsentlig del av administrationen är att kunna hantera de recept som ligger lagrade i databasen. Först av allt måste användaren ange vilket recept som skall ändras. Här kan användaren filtrera med hjälp av ett sökord för att enklare hitta det receptet som skall ändras. I detta steg har användaren även möjlighet att ta bort hela receptet genom att trycka på en knapp till höger om titeln. När knappen är intryckt dyker en ruta upp där användaren är tvungen att ange ifall den verkligen vill ta bort receptet. Se figur 17.

När användaren har hittat det recept som skall ändras, klickar användaren på titeln varifrån användaren sedan skickas till en ny sida. Härifrån kan användaren ändra det angivna receptet. Användaren kan ändra titel, beskrivning, ingredienser, kategori och ladda upp en bild. Se figur 18.

### Hantera recept

Info: För att ändra ett recept, klicka på titeln.

13 inlägg på 3 sidor

[<< Första](#) | [< Förra](#) | [1](#) [2](#) [3](#) | [Nästa](#) > | [Sista](#) >>

<input type="button" value="Ta bort"/>	<a href="#">kvckling i ugn med buketter, klyftor och en dipp</a>
<input type="button" value="Ta bort"/>	<a href="#">lammfärsbiffar med bakade potatis- &amp; tomathalvor</a>
<input type="button" value="Ta bort"/>	<a href="#">örtmarinerade kvcklingspett med avokadosallad</a>
<input type="button" value="Ta bort"/>	<a href="#">fiskgröta med fiskkrydda från Stigs</a>
<input type="button" value="Ta bort"/>	<a href="#">Stens goda korv med champinjoner och spaghetti</a>
<input type="button" value="Ta bort"/>	<a href="#">mustig färsgröta med ölandsbönor och goa grejer</a>

Figur 17 Hantera recept

### börjes goda dragonkorv med potatismos

[<< Tillbaka](#)

Titel: börjes goda dragonkorv med potatismos

Beskrivning: Skala och dela potatisen. Koka i lättsaltat vatten i cirka 20 minuter. Korven steker du bara i en stekpanna i lite smör. Vi skickar med dragonkorv och så prinskorvar till barnen.

När potatisen kokat klart håller du av vattnet. Mosa pota

Kategori:

#### Ingredienser

Info: Tar inte bort ingrediensen, bara kopplingen till receptet.

1 knippe rädisor

1 pikt Börjes dragonkorv

2-3 morötter

salt och peppar

½ gurka

3 dl mjölk

1 pikt prinskorv

ca 1 kg potatis

Figur 18 Redigera recept

### 4.2.3 Hantera kategori

Som angivits tidigare kan ett recept innehålla en kategori. För detta behövs administration för att kunna lägga till, ta bort och redigera kategorier.

För att lägga till en kategori används ett inmatningsfält. Inmatningsfältet får inte vara tomt och en kontroll görs som kollar så att den inmatade kategorin inte överensstämmer med en kategori som redan finns i databasen. Se figur 19.

För att ta bort och redigera en kategori används en rullgardinslista. Användaren markerar en kategori i listan. Ifall en kategori skall tas bort görs en kontroll så att ett recept inte har en koppling till den markerade kategorin. Se figur 19.

#### Lägg till kategori

Kategori

Spara

#### Hantera kategori

-Välj kategori-

Ta bort Ändra

Figur 19 Hantera kategori

Ifall kategorin skall ändras skickas användaren vidare till en ny sida där kategorins information finns i ett formulär. Härifrån kan sedan användaren enkelt ändra kategorins namn i inmatningsfältet och uppdatera.

### 4.2.4 Statistik

Ett av kraven i kravspecifikationen var att kunna se statistik för hur många besökare som har varit inne och läst varje enskilt recept. På administrationssidan kan man se genom en lista topp 15 av de recept som flest besökare har varit inne och läst. Se figur 20.

Statistiken sparas i en egen tabell i databasen med namnet *tbl\_stat*. Se figur 9.

#### Statistik

##### Topp 15

	Antal:
test1	5
test4	3
test	3
test3	2
börjes goda dragonkorv med potatismos	2
test2	2

Figur 20 Statistik

## 5 Diskussion och slutsatser

I detta avsnitt diskuteras resultatet av examensarbetet. Även slutsatser och rekommendationer. Till sist en beskrivning av eventuell vidareutveckling av webbapplikationen.

### 5.1 Resultatdiskussion

Här diskuteras resultatet av de olika frågeställningarna som var fastställda för examensarbetet.

#### 5.1.1 Hur överför man med ett enkelt tillvägagångssätt information från en PDF-fil till en databas?

Efter en hel del förstudie skapades till slut en applikation som kunde läsa textfiler och läsa av textfilen rad för rad. PDF dokumenten är tvungna att konverteras till textfil och sedan i textfilen mata in tre olika tecken som indikerar vad som är titel, ingrediens och beskrivning. Med hjälp av detta sparas mycket tid jämfört med att klippa ut från ett PDF dokument och sedan klistra in i ett formulär. Det är inte säkert att det vart den bästa lösningen, men den sparar tid och är enkel att använda.

#### 5.1.2 Hur skapar man användarvänliga formulär, ur synpunkt av interaktionsdesign?

Med hjälp av boken *Web Form Design* skriven av Luke Wroblewski [8], skapades användarvänliga formulär. Det är viktigt att tänka igenom designen av formuläret. Genom det hjälper man besökare och användare att enkelt och på ett snabbt sätt kunna fylla i ett formulär.

### 5.2 Slutsatser och rekommendationer

Som är nämnt tidigare var det stora problem vid implementationen med de svenska bokstäverna å, ä och ö. Detta pga. att det användes olika teckenkodningar på Familyfoods hemsida och den webbapplikation som har utvecklats. I detta fall gäller det att innan man börjar utveckla en applikation på internet att man noterar vilken teckenkodning som används och bör användas. Det viktigaste är att det är samma teckenkodning överallt.

Användning av testmiljö i någon form är att rekommendera vid utveckling av nya applikationer. Det är en stor fördel att enkelt och smidigt kunna testa sin applikation för att hitta lösningar på problem och hitta eventuella buggar.

En annan rekommendation är att använda någon form av versionshanteringsprogram. För att kunna hantera olika versioner av sin kod. Har man lyckats förstöra något som har fungerat i koden är det skönt att enkelt kunna gå tillbaka till en tidigare version.

### **5.3 Vidareutveckling**

En vidareutveckling av webbapplikationen är helt klart en möjlighet. I början av examensarbetet arbetades olika idéer fram (brainstorming) för hur applikationen skulle fungera. Pga. avgränsningar fick en hel del idéer plockas bort. Men några av idéerna är helt klart relevanta att implementera i framtiden. Bland dessa fanns möjligheten för en besökare att kunna lägga en kommentar till varje recept. En annan idé är en lista som visar de populäraste recepten utifrån statistiken som redan finns lagrad i databasen. En länk till utvalt recept i ett blogg inlägg är också en funktion som skulle kunna implementeras i framtiden.

## 6 Referenser

- [1] Ted Padova (2011), *Adobe Acrobat X PDF Bible*, ISBN13 9780470612910
- [2] Adobe PDF – historia (2012)  
<http://www.adobe.com/se/products/acrobat/adobepdf.html>  
(Acc. 2012-03-01)
- [3] Matt Doyle (2009), *Beginning PHP 5.3*, ISBN13 9780470413968
- [4] Thomas Connolly (2009), *Database Systems*, ISBN13 9780321523068
- [5] W3Schools – SQL Tutorial (2012) <http://www.w3schools.com/sql/>  
(Acc. 2012-03-07)
- [6] Sheeri K Cabral, Keith Murphy (2009), *MySQL Administrator's Bible*, ISBN13 9780470416914
- [7] M Delisle (2010), *Mastering phpMyAdmin 3.3.x for Effective MySQL Management*, ISBN13 9781849513548
- [8] Luke Wroblewski (2008), *Web Form Design*, ISBN 1-933820-24-1
- [9] Ted Padova (2008), *Adobe Acrobat 9 PDF Bible*, ISBN13 9780470379

## 7 Sökord

Formulärdesign .....	16, 28	phpMyAdmin.....	15, 21
MySQL .....	15, 21	Relationsdatabas .....	12
PDF .....	10, 22	SQL.....	12, 13, 14, 15
PHP .....	10, 15, 22, 23		