

# Smidyo

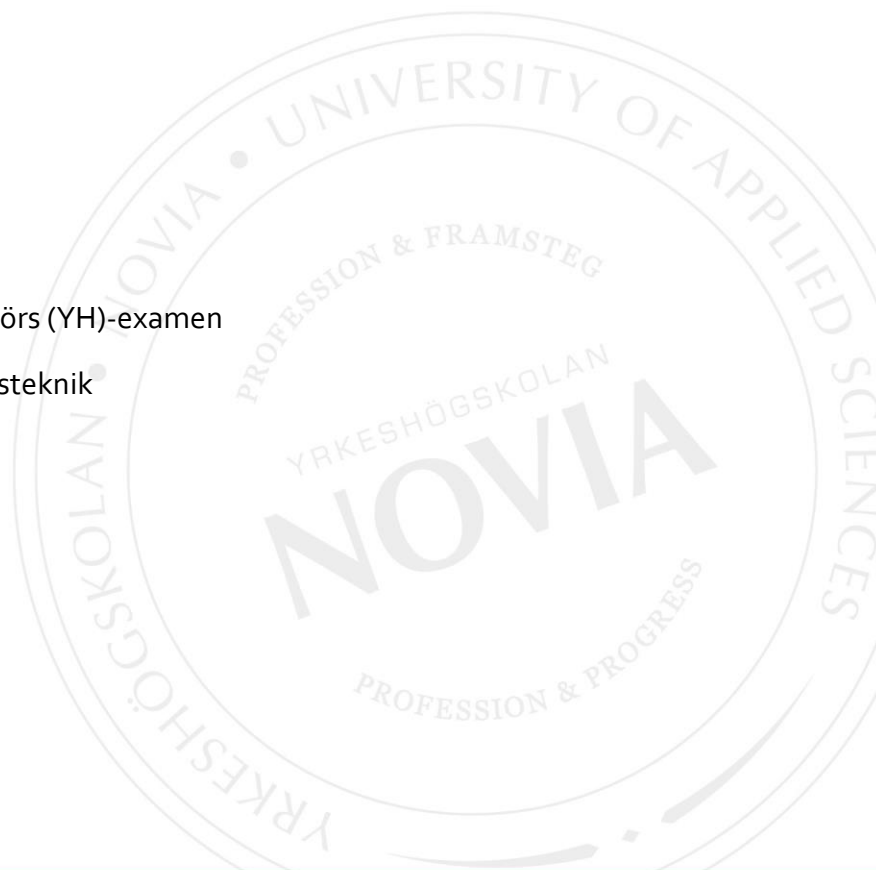
Från ritning till del

Joakim Segervall

Examensarbete för ingenjör (YH)-examen

Utbildningen Informationsteknik

Vasa 2017



## EXAMENSARBETE

Författare: Joakim Segervall  
Utbildning och ort: Informationsteknik, Vasa  
Handledare: Kaj Wikman

Titel: *Smidyo – Från ritning till del*

---

Datum 18.4.2017      Sidantal 24      Bilagor -

---

### Abstrakt

Examensarbetet utfördes på uppdrag åt CNC Design Öb. Uppdraget gick ut på att från grunden skapa en webbapplikation för direkt offertberäkning av laserskärning och gravering av olika material utgående från kundens egna uppladdade ritning. Smidyo blev skapad som grund och första implementation för vidareutveckling. Smidyo är därav designat för att senare kunna implementera flera andra tillverkningstekniker såsom UV- och 3D-printning.

Resultatet blev en ASP.NET MVC 5-applikation som använder sig av ett .NET CAD-bibliotek som bas för uträkningen av vektorernas längder och dess areor. Applikationen använder sig av SQL Server-databaser för olika material, priser och dylikt. För kundens enkelhet använder sig applikationen av PayPal, vilket gör det enkelt och snabbt för kunden att beställa sin del.

---

Språk: svenska      Nyckelord: ASP.NET, CAD, IT, C#, webshop

---

## OPINNÄYTETYÖ

Tekijä: Joakim Segervall  
Koulutus ja paikkakunta: Tietotekniikka, Vaasa  
Ohjaaja: Kaj Wikman

Nimike: *Smidyo – Piirustuksesta valmiiseen osaan*

---

Päivämäärä 18.4.2017      Sivumäärä 24      Liitteet -

---

### **Tiivistelmä**

Opinnäytetyön toimeksiantaja oli CNC Design Öb. Tehtävän tarkoituksena oli luoda web-sovellus erilaisten materiaalien laserleikkauksen ja kaiverruksen suoraa tarjouslaskentaa varten asiakkaan lähettämän piirustuksen pohjalta. Smidyo luotiin perustaksi ja ensimmäiseksi sovellukseksi jatkokehittelyä varten. Smidyo on suunniteltu siten, että siinä voidaan myöhemmin toteuttaa useita muita valmistustekniikoita, kuten UV- ja 3D-tulostus.

Tuloksena on ASP.NET MVC 5 -sovellus, joka käyttää .NET CAD -kirjastoa perustana vektoripituuksien ja -alojen laskennalle. Sovellus käyttää SQL Server -tietokantoja erilaisille materiaaleille, hinnoille jne. Asiakasystävällisyyttä lisää PayPal, joka helpottaa ja nopeuttaa tilauksen tekemistä.

---

Kieli: ruotsi      Avainsanat: ASP.NET, CAD, IT, C#, webshop

---

## BACHELOR'S THESIS

Author: Joakim Segervall  
Degree Programme: Information Technology, Vasa  
Supervisor: Kaj Wikman

Title: *Smidyo – From drawing to part*

---

Date 18.4.2017      Number of pages 24      Appendices -

---

### **Abstract**

This thesis was made on behalf of CNC Design Öb. The thesis' aim was to create a web application from scratch, which offers direct quotation on laser cut and engraved parts of different materials based on drawings uploaded by customers. Smidyo was created as the first implementation for further development. Smidyo is thereby designed to implement other manufacturing techniques later, such as UV-printing and 3D-printing.

The result of this thesis was an ASP.NET MVC 5 application which uses a .NET CAD library as a base for calculating the length and areas of the vectors. The application uses SQL Server databases for storing for example different materials and prices. PayPal is used as payment for simplicity, which makes it fast and easy for the customer to order his part.

---

Language: swedish      Key words: ASP.NET, CAD, IT, C#, webshop

---

# Innehållsförteckning

1	Inledning.....	1
1.1	Arbetsgivare .....	1
1.2	Uppdrag.....	1
2	Tekniker.....	1
2.1	Microsoft Visual Studio .....	2
2.2	C#.....	2
2.2.1	CLI.....	3
2.3	WoutWare .....	3
2.3.1	CadLib 4.0 .....	3
2.3.2	WW.Math .....	3
2.4	Vektorfiler .....	3
2.4.1	Uppbyggnad.....	4
2.4.2	DXF.....	5
2.4.3	DWG.....	5
2.5	SQL Server Express.....	5
2.6	ASP.NET .....	6
2.6.1	ASP.NET MVC 5.....	6
2.7	ADO.NET .....	7
2.8	Entity framework .....	7
2.9	Microsoft Azure.....	8
3	Applikationen .....	9
3.1	Planering.....	9
3.1.1	Krav och specifikationer .....	9
3.1.2	Val av databashanterare .....	10
3.1.3	Val av CAD-bibliotek.....	10
3.1.4	Orderstatus .....	11
3.2	Design .....	11
3.2.1	Make.....	12
3.2.2	Cart.....	13
3.2.3	Checkout.....	14
3.3	Hjälpbibliotek med WoutWare CadLib 4.0 som grund.....	15
3.3.1	Exportering av bilder .....	17
3.4	Databaser.....	18
3.5	PayPal API .....	18
3.6	Fraktberäkning.....	19
3.7	E-post.....	19

3.8	Lagring.....	20
3.9	Administrationspanel.....	21
3.9.1	Loggning av fel.....	21
4	Resultat och diskussion .....	22
4.1	Resultat .....	22
4.2	Vidareutveckling.....	22
4.3	Diskussion .....	22
5	Källförteckning.....	23

# 1 Inledning

Examensarbetet gjordes på uppdrag av CNC Design Öb. Uppdraget var att skapa en webbapplikation för online offertberäkning av laserskurna delar, baserat på kundens ritning.

## 1.1 Arbetsgivare

CNC Design Öb är ett företag i Nykarleby. Företaget grundades 2011 av två bröder, och sysslar med att tillverka olika saker, bl.a. genom laserskärning och gravering, UV digital flatbäddspritning och 3D-printning. De är kända från saker som Deal with it-glasögon, "Nintendo NX fake leak", och sin Nintendo Switch-replika. (CNC Design)

## 1.2 Uppdrag

Uppdraget var att skapa en slags webbtjänst som automatiskt kan beräkna priser på beställda delar utgående från kundens ritningar, och låter kunden beställa dessa delar direkt utan personlig interaktion med företaget. För tillfället e-postar de med kunden, och detta kan vara tidskrävande och kräver manuell offertberäkning. Webbtjänsten ska också innehålla förhandsvisningar på hur kundens del kommer att se ut, och på så vis undviks missförstånd eftersom att kunden själv godkänner förhandsvisningen.

Första delen av denna webbtjänst är laserskärning och gravering, som blev mitt uppdrag tillsammans med själva botten för webbtjänsten. Webbtjänsten ska senare kunna expandera för att tillåta andra tillverkningstekniker såsom till exempel 3D-printning. Styling av själva hemsidan görs med CSS av CNC Design.

# 2 Tekniker

För att utveckla webbtjänsten användes ASP.NET MVC 5 och ett .NET CAD-bibliotek som hjälp. Utvecklingsverktyget som användes för programmering var Microsoft Visual Studio, och Microsoft SQL Server Management Studio för databashantering. Ritningar att testa applikationen med gjordes av uppdragsgivaren i CAD-programmet CorelDraw.

## 2.1 Microsoft Visual Studio

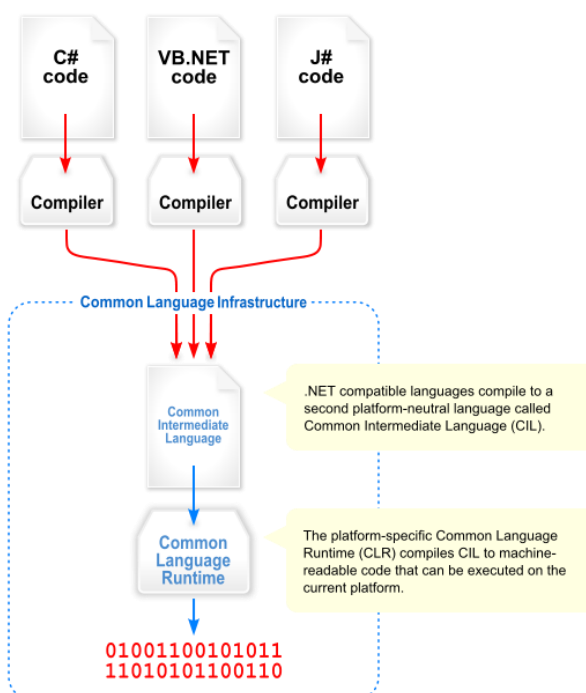
Visual Studio är ett utvecklingsverktyg skapat av Microsoft. Första versionen utgavs i februari 1997. Visual Studio används för att skapa datorprogram, hemsidor, webbapplikationer och webbtjänster. Det använder en kodredigerare med IntelliSense, som är Microsofts svar på Intelligent Code Completion. Visual Studio har också en del andra inbyggda verktyg, såsom visuell designer för design av användargränssnitt och hemsidor.

Visual Studio 2015 finns i 5 olika utgåvor. Medan Professional, Test Professional och Enterprise är för professionellt bruk finns också Community och Express, som är mindre gratisversioner för hobbybruk och självständiga utvecklare. (Microsoft Visual Studio, 2016)

## 2.2 C#

C# är ett objektorienterat och typsäkert programmeringsspråk. Det kan bland annat användas till att utveckla datorprogram för Windows och klient-server applikationer. C# påminner om C++ och Java, och är således inte ett svårt språk för de som redan är bekanta med C, C++ eller Java.

Eftersom att C# är objektorienterat stöder det inkapsling, arv och polymorfism. C# program använder .NET Framework, som i sin tur använder Common Language Runtime (CLR) vilket är Microsofts implementation av Common Language Infrastructure. Detta betyder att en kod skriven i C# kompileras till Intermediate Language (IL) kod, som sparas på disken som en körbar fil. När denna fil exekveras laddas IL-koden in till CLR-komponenten som utför en Just in Time-kompilering för att konvertera IL-koden till den använda maskinens maskinkod. Kod som körs av CLR brukar kallas för hanterad kod, till motsats från ohanterad kod som direkt kompileras till maskinkod. (MSDN, 2015)



Figur 1, Överblick av CLI..



### **2.2.1 CLI**

CLI står för Common Language Infrastructure och är en specifikation som beskriver körbar kod och körtidsmiljö. Detta gör att man kan använda flera s.k. högnivåspråk på flera olika datorer utan att behöva skriva om koden för den specifika arkitekturen. Exempel på olika implementationer av CLI är .NET Framework, Mono, dotGnu och Portable.NET. (Common Language Infrastructure, 2017)

## **2.3 WoutWare**

WoutWare är en självständig programutvecklare som fokuserar på CAD, grafik och matematiska lösningar i .NET (WoutWare, u.d.)

### **2.3.1 CadLib 4.0**

CadLib 4.0 är ett .NET-bibliotek för CAD-funktionalitet från WoutWare. CadLib kan läsa och skriva DWG och DXF-filer i samtliga .NET språk. CadLib kan också t.ex. exportera ritningar som bilder, PDF, och SVG. Biblioteket stöder också rendering av 2D och 3D-ritningar, både som trådmodell och yta. Biblioteket finns i tre utgåvor, Basic, Standard samt Professional. Basic-utgåvan kan endast läsa och skriva DXF-filer, medan Standard kan exportera dessa som bl.a. bilder och PDF. Professional-utgåvan kan också läsa och skriva DWG-filer, samt stöder rendering.

### **2.3.2 WW.Math**

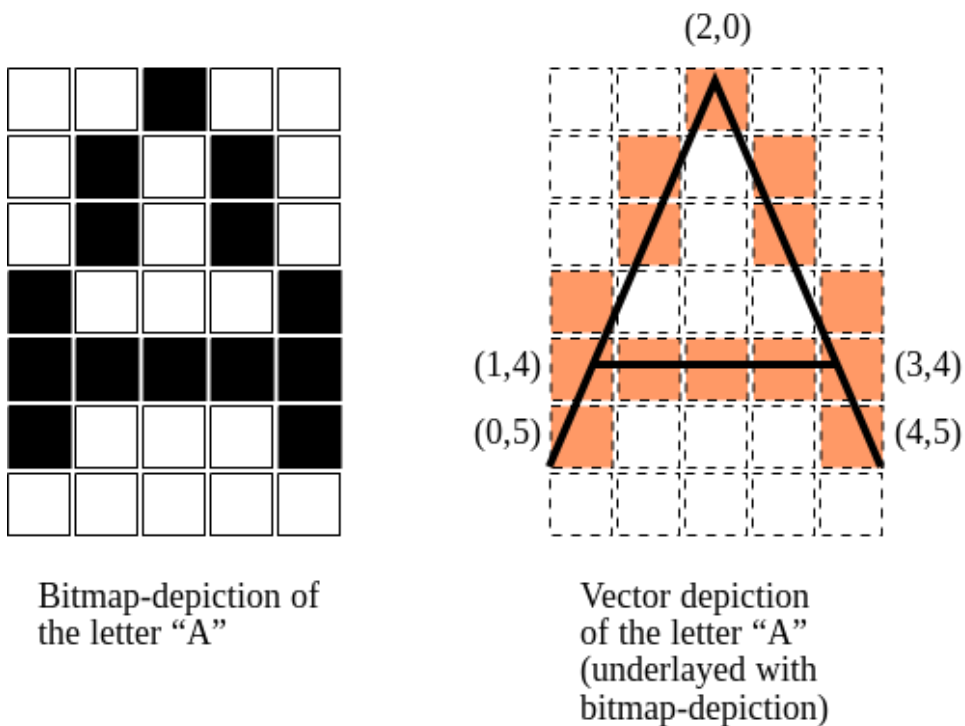
WW.Math är en .NET komponent för matematik som stöder 2D/3D-punkter, vektorer, matriser, linjer med mera. Denna komponent kan köpas enskilt, men följer också med CadLib 4.0. Detta bibliotek kan användas t.ex. för beräkning av areor i en tvådimensionell ritning.

## **2.4 Vektorfiler**

Vektorgrafik är uppbyggd av punkter, linjer, cirklar och andra liknande objekt. Dessa filer består inte av bildpunkter, och kan på det sättet skalas upp och ner hur som helst utan att kvalitén blir lidande. I stort sett all CAD använder sig av vektorbaserad teknik, både tvådimensionell och tredimensionell. (Vektorgrafik, 2015)

### 2.4.1 Uppbyggnad

För att förstå vektorgrafik lättare kan man tänka sig att man ritar en cirkel, för att denna cirkel ska kunna skapas behöver den två olika faktorer, radien och koordinater för mittpunkten. I en vektorfil sparas endast dessa värden, och inte alla bildpunkter som utgör cirkeln. Detta ger också den sparade filen en mindre filstorlek. På detta sätt kan man också zooma in på en linje utan att den blir bredare, p.g.a. att endast start och slutpunkt för linjen skalas men ej linjen i sig själv.



Figur 2, Illustration av bitmap och vektorgrafik.

## 2.4.2 DXF

DXF står för Drawing Interchange Format, och är ett CAD-dataformat skapad av Autodesk för att kunna växla ritningar mellan AutoCAD och andra CAD-program. En DXF fil kan läsas av vilken texteditor som helst, t.ex. Notepad. En sådan fil innehåller olika sektioner, som t.ex. HEADER och ENTITIES. I sektionen ENTITIES så finns all information om de olika objekten som finns i ritningen. Exempel på olika s.k. entities är LINE, CIRCLE, SPLINE och POLYLINE. Dessa objekt innehåller alla värden som behövs för att kunna rita upp objektet på skärmen. Se exempel i figur 3. (AutoCAD DXF, 2016)

```
SECTION
2
ENTITIES
0
SPLINE
5
70|
330
1F
100
AcDbEntity
8
Layer 1
370
9
100
AcDbSpline
210
0.0
220
0.0
230
1.0
70
11
71
3
```

Figur 3, Exempel på DXF-fil.

## 2.4.3 DWG

DWG kommer från det engelska ordet ”drawing” och är ett standardformat för flera olika CAD-program, såsom AutoCAD, Intellicad och PowerCAD. Det stöds också av många andra CAD-program. DWG utvecklades av Mike Riddle i slutet av 1970-talet och blev licensierat av Autodesk 1982. Autodesk underhåller filformatet och designar, definierar och utvecklar nya versioner. Autodesk säljer ett bibliotek kallat RealDWG som kan användas för att läsa och skriva DWG-filer i icke-konkurrensbetonade applikationer. (.dwg, 2016)

## 2.5 SQL Server Express

SQL Server är en databasserver utvecklad av Microsoft. SQL Server Express är en mindre utgåva av SQL Server som är gratis under vissa omständigheter. Genom ett plugin kallat SQL Server Data Tools (SSDT) kan man direkt jobba med SQL Server-databaser i Microsoft Visual Studio. (Microsoft, 2016)

## 2.6 ASP.NET

ASP.NET är en webbplattform som bygger på .NET Framework för att skapa serverbaserade webbapplikationer. Man kan koda ASP.NET med vilket CLR-kompatibelt programmeringsspråk som helst, såsom C# och Basic.

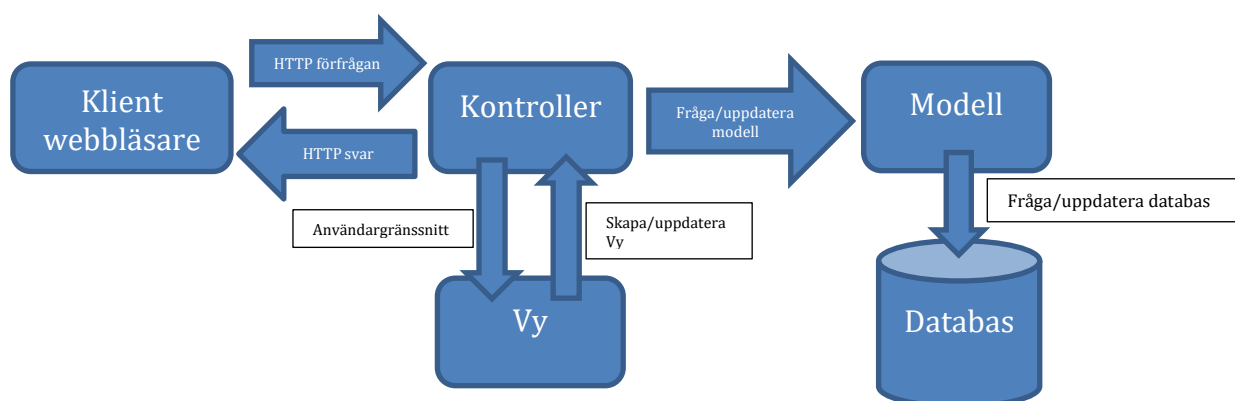
ASP.NET innehåller tre olika ramverk för att skapa webbapplikationer, Web Forms, ASP.NET MVC och ASP.NET Web Pages. Man kan basera sitt val av ramverk på bl.a. utvecklingsstil och typen av applikation man tänker skapa.

Web Forms är sidor som skapas med hjälp av HTML, klient-script, serverkontroller och serverkod. När en användare begär en sida kompileras denna på servern och ramverket genererar HTML som skickas tillbaka användaren. Man kan enkelt designa sidorna med dra-och-släpp teknik. (Microsoft, u.d.)

Web Pages är en annan teknologi som fokuserar på att skapa dynamiska webbsidor, med hjälp av kod och databaser på serversidan. (Microsoft, u.d.)

### 2.6.1 ASP.NET MVC 5

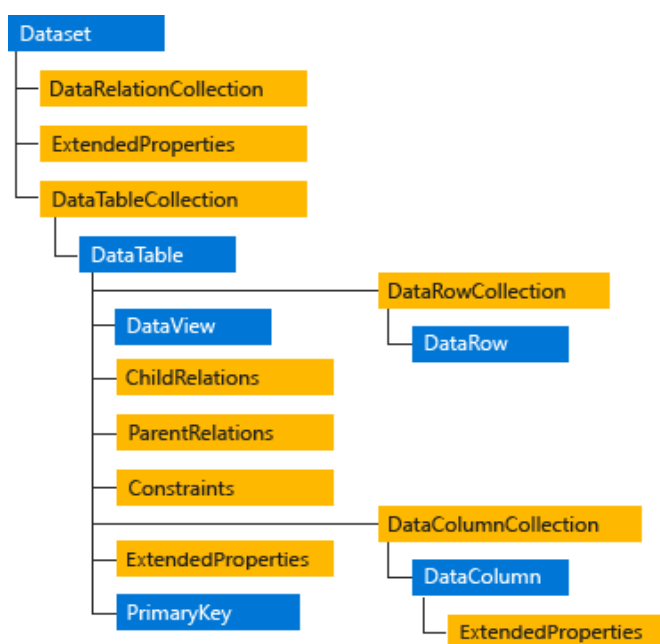
MVC står för Model-View-Controller som är ett designmönster som bryter upp applikationen i tre delar. Modell är ett objekt som kan innehålla data som visas på sidan. Kontroller hanterar interaktionen mellan användaren och servern, samt uppdaterar modellen och skickar vidare informationen till vyn. Vy tar emot informationen och genererar ett användargränssnitt utgående från den. (ASP.NET MVC, 2016)



Figur 4, MVC.

## 2.7 ADO.NET

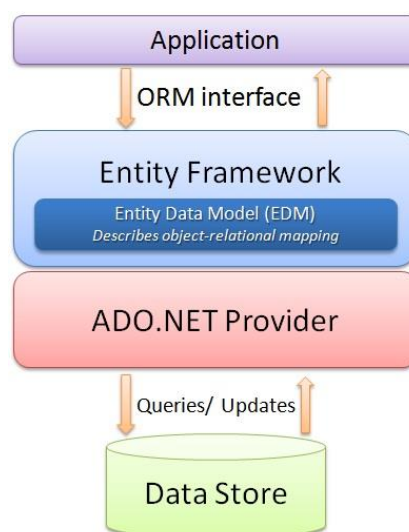
ADO.NET är en samling klasser som ger funktionalitet för att komma åt datakällor i .NET Framework. Med ADO.NET kan man koppla ihop sin applikation med t.ex. SQL server, XML, och datakällor via OLE DB och ODBC. Detta gör att man kan tillverka en applikation vars användare kan komma åt, uppdatera och hantera data i en datakälla. Data man får tag på via ADO.NET kan lagras i ett ADO.NET DataSet, och med hjälp av detta DataSet kan programmeraren använda data som finns där, såsom sätta in nya data, ändra eller ta bort befintliga data. (Microsoft, 2017)



Figur 5, ADO.NET arkitektur.

## 2.8 Entity framework

Entity Framework, förkortat EF, är en öppen källkod Object-Relational Mapper (ORM) som tillåter utvecklaren att hantera data från en datakälla som ett objekt. Med hjälp av detta ramverk behöver utvecklaren inte oroa sig över hur databasens tabeller och kolumner ser ut, utan kan direkt jobba med objekt som sedan lagras som en rad i en tabell i databasen. EF översätter objekten till motsvarande databasoperationer. Detta gör också att utvecklaren behöver skriva mindre kod än för traditionella applikationer. (Microsoft, 2013)



Figur 6, EF arkitektur.

Entity Framework kan användas på flera sätt, kod först och databas först är två av dessa metoder. Med kod först-modellen måste utvecklaren först själv skapa de domänklasser han vill kunna spara och använda i databasen, och EF skapar sedan en databas utgående från de klasserna och relationerna. Med databas först-modellen kan utvecklaren designa databasen först på valfritt sätt, och sedan låta EF ta hand om att skapa motsvarande klasser utgående från databasen.



Figur 7, Databas först.



Figur 8, Kod först.

## 2.9 Microsoft Azure

Microsoft Azure är en molnplattform som har många olika funktioner. I detta examensarbete är det främst deras SQL Server, lagring samt App Service-tjänster som är intressant. I samtliga av dessa finns det flera olika prisnivåer att välja mellan. Den billigaste App Service-tjänsten är gratis, men är då s.k. delad processor tid, med 60 CPU-minuter/dag. Azure gör att man också enkelt kan expandera och uppgradera sin tjänst till en snabbare variant ifall det behovet uppstår. (Microsoft, 2017)

INSTANS	KÄRNOR	RAM	STORAGE	PRISER
<b>S1</b> Standard	1	1,75 GB	50 GB	€0,085/per timme
<b>S2</b> Standard	2	3,50 GB	50 GB	€0,169/per timme
<b>S3</b> Standard	4	7 GB	50 GB	€0,338/per timme

Figur 9, App Service Standard, specifikationer och prisuppgifter.

## 3 Applikationen

Före planeringen och utvecklingen kunde börja krävdes först forskning om hur stort jobb det var och ifall det ens var möjligt att lösa problemet. Forskningen gjordes bland annat om hur vektorfiler fungerar. Det framkom att applikationen nog var möjlig, men att ett tredje parts bibliotek skulle behövas.

### 3.1 Planering

För att applikationen skulle kunna genomföras krävdes planering, tidigt satta krav på hur applikationen skulle fungera, samt förståelse för hur tillverkningstekniken utförs. Under utvecklingen uppstod ändå flera problem som var svåra att förutse under planeringsfasen. Med hjälp av fria händer och god kommunikation med uppdragsgivaren löstes dessa problem i de flesta fall snabbt.

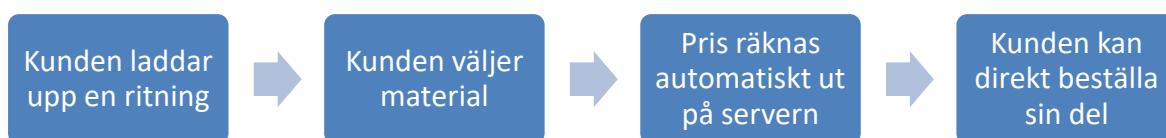
De flesta av de stora valen bestämdes redan före utvecklingen alls började, såsom val av databashanterare och CAD-bibliotek.

#### 3.1.1 Krav och specifikationer

Det största kravet var att applikationen skulle vara lättanvänd, till motsats från de få liknande lösningar som redan existerar, vilka är invecklade och kan på så sätt avskräcka användaren.

Vilka slags filformat applikationen ska kunna fungera med undersöktes tillsammans med uppdragsgivaren, och det framkom att de flesta CAD-program idag klarar av att exportera till DXF. Detta passade också bra vid val av CAD-bibliotek, eftersom att de bibliotek som klarade av flera filtyper var betydligt dyrare.

Förhandsvisning av kundens specialbeställda del, med valt material och illustration av hur lasern reagerar på det valda materialet, var ett löst krav som behövde realiserats åtminstone till en viss del.



Figur 10, Arbetsflöde.

### 3.1.2 Val av databashanterare

Först övervägdes Microsoft Access som databashanterare, främst på grund av att det är enkelt att skapa formulär och rapporter direkt i Access vilket gör det lätt för uppdragsgivaren att använda applikationen. Men vid lite efterforskning så framkom det att Microsoft Access inte var lämplig, och att det gav mindre valmöjligheter för distribuering.

På grund av detta valdes Microsoft SQL Server Express, som är en gratisversion som sedan kan uppgraderas vid behov. En SQL Server-databas kan man också distribuera till Microsoft Azure.

Under utvecklingen användes SQL Server Express lokalt på utvecklingsdatorn, men databasen exporterades sedan till en Azure SQL Server.

### 3.1.3 Val av CAD-bibliotek

På grund av vektorfilers upplägg skulle det varit komplicerat och tagit lång tid att själv koda ett bibliotek som behandlar dessa filer och klarar av att exportera dem som bildfiler. Hade man själv kodat ett bibliotek hade det också krävt mera jobb om man i framtiden vill sätta in fler funktioner. Därför undersöktes och testades ett antal CAD-bibliotek som redan hade de funktioner som behövdes.

Utgående från detta bibliotek kodas ett antal hjälpklasser, som sedan används i ASP.NET-applikationen, på detta vis går det smidigare att byta CAD-bibliotek ifall det behovet uppstår i framtiden.

CAD.NET var ett alternativ, som hade allt som behövdes och förmodligen lite mera. Detta skulle annars ha varit det bästa alternativet, men vid en offertförfrågan framkom det att biblioteket var alldeles för dyrt.

netDxf var ett alternativ som dessutom var öppen källkod. På grund av för få funktioner och osäkerhet inom stöd och nyare versioner valdes inte detta bibliotek.

WoutWares CadLib hade de funktioner som behövdes, och dessutom till ett överkomligt pris. WoutWare hade också ett matematikbibliotek för olika slags beräkningar på två och tredimensionella vektorer som skulle vara till hjälp. På grund av dessa fördelar valdes detta bibliotek.



### 3.1.4 Orderstatus

En order ska kunna ha fem olika statusar:

- Betalningen lyckades, beställningen klar.
- Betalningen lyckades inte, väntar på betalning.
- Paketet är större än vad tillgängliga fraktmetoder tillåter, kunden kontaktas manuellt.
- Ordern ivägskickad, kunden meddelas.
- Kunden avbryter ordern.

När en kund lägger in en order skickas han till betalningen, om betalningen inte lyckas av någon anledning, så får ordern statusen ”betalning lyckades ej”. Om detta händer, sätts ordern på is och uppdragsgivaren kan kontakta kunden för att höra hur han vill betala, eller om han vill avbryta sin order.

Applikationen har två olika fraktmetoder, om ingen av dem är tillgängliga för kundens order, t.ex. på grund av fysisk storlek eller vikt, får ordern statusen ”väntar på fraktmetod”. Uppdragsgivaren kan då kontakta kunden och komma överens om ett fraktsätt.

Om allt gick som det ska och betalningen lyckades, får ordern statusen ”betalning lyckades”. Ordern kan då tillverkas och ivägskickas enligt normal procedur.

När ordern är ivägskickad, får ordern statusen ”ivägskickad” och kunden kontaktas med försändelsekod.

Ifall kunden av någon anledning vill avbryta sin order, ändras statusen till ”avbruten”, men ordern försvinner ej automatiskt ur databasen.

Oavsett vilken status en order får, ska kunden meddelas detta via e-post. På så sätt kan kunden lätt se om det lyckades eller inte, och i så fall vad som var problemet. Orderstatusar i denna text är inte applikationens namn på statusar, utan endast pseudostatusar.

## 3.2 Design

Designen blev mer eller mindre helt gjord av uppdragsgivaren, eftersom att de var mer vana vid design och redan hade på klart hur de ville att applikationen skulle se ut. Efter att uppdragsgivaren designat grunden i HTML och CSS översattes detta till en ASP.NET-applikation, och all logik för uträkning och databashantering gjordes.

### 3.2.1 Make

På Make sidan kan användaren ladda upp sin fil, välja material och få se en förhandsvisning på hur biten kommer att se ut i valt material. Här kan användaren också få se ungefärligt pris och eventuell mängdrabatt.

När användaren för pekaren över de olika raderna i listan till höger så förstärks dessa linjer på förhandsvisningsbilden, så att användaren kan se att allt är korrekt. Användaren kan godkänna detta genom att trycka på knappen "Add to part cart". Biten hamnar då i kundvagnen och ett meddelande om detta visas.

**SMIDYO BETA** Laser Express Quote Contact 0 parts Login

#### MAKE DIRECT: LASER

✓ Upload your drawing

Upload DXF Export guide

SMIDYOTestbit.dxf  
73.3 mm ↔ 103.5 mm

This is looking from the front of the piece. In the preview, reverse engraving and scoring can be seen through the material, even though the material selected isn't transparent. Make sure to flip the reverse artwork if needed!

✓ Pick material

Metal Others Paper Plastic Wood

Acrylic colored Black

3 mm 5 mm 8 mm

In stock Material guide

No Add 3M adhesive to backside

Get it made

Quantity 1 Discount 0.0%

Unit price 6.63

Base price 5.00

Total price (incl. 24% VAT) 11.63 €

Commit to Part Cart

#### Summary

SMIDYO  
MAKE DIRECT: LASER  
Clear acrylic, extruded  
Rev. engrave Engrave Cut Score Rev. score  
Smidyo is an all-in-one manufacturing platform. For more info, please visit make.direct

For rough reference only, not exact indication of final result. Please refer to the Material Guide to see exactly how the material reacts to the laser.

Item	Amount	Subtotal
Cutting	592.58 mm	1.05
Engraving	3,200.00 mm <sup>2</sup>	4.80
Scoring	430.66 mm	0.18
Reverse Engraving	176.00 mm <sup>2</sup>	0.26
Reverse Scoring	71.15 mm	0.03
Material	0.01 m <sup>2</sup>	0.30

Figur 11, Make-sidan.

### 3.2.2 Cart

På sidan Cart räknas användarens alla bitar upp i en lista, samt priser för var och en av dessa. Användaren kan här också ändra kvantitet och beräkna om mängdrabatten. Genom att välja ett land kan användaren se fraktpriset. Om användaren har lagt in de bitar och de antal han vill ha, kan han gå vidare till Checkout.

The screenshot shows the Smidyo cart interface. At the top, there's a navigation bar with the Smidyo logo, a 'Laser' dropdown menu, and links for 'Express Quote' and 'Contact'. On the right, there's a shopping cart icon with '1 part' and a 'Log in' link.

The main content area is titled 'PART CART' and includes a disclaimer: 'For rough reference only, not exact indication of final result. Please refer to the Material Guide to see exactly how the material reacts to the laser.'

Part	Specification	Subtotal	
<p>SMIDYOtestbit.dxf</p> <p>Acrylic colored Black 3mm</p> <p>103 mm x 73 mm x 32 g</p>	<b>LASER</b>	Quantity: 1	
	Engraving	4.80	
	Scoring	0.18	
	Cutting	1.05	
	Reverse Engraving	0.26	
	Reverse Scoring	0.03	
	3M adhesive	0.00	
	0.01 m <sup>2</sup> Material - In stock	0.30	
	<b>Total one-off price</b>	<b>6.63</b>	Discount: 0.0%
			Unit price: 6.63
		Base price: 5.00	
		Subtotal: 11.63	
		Price: 11.63 €	
		Price: 9.38 €	

At the bottom left, there's a 'Calculate shipping' button and a 'Select country' dropdown menu.

At the bottom right, there's a section titled 'Accounts coming soon!' with the following text: 'Having a Smidyo account makes checkout not only fast, but also easy to re-order parts you've ordered previously. We offer 0% VAT intra-community purchases for registered companies in the EU. We issue post-delivery invoices to Finnish companies, and also to foreign companies per request.' Below this text are three buttons: 'Login', 'Register', and 'Checkout without account'.

Figur 12, Cart-sidan.

### 3.2.3 Checkout

Checkout sidan är sista sidan vid beställning, här sätter användaren in sin adress och kontaktuppgifter. Användaren kan också välja att sätta olika adresser för betalning och leverering. Just nu är PayPal det enda betalningssättet för en vanlig kund.

SMIDYO BETA
Laser
Express Quote
Contact
Admin
🛒 1 part
Hello info@cnc-design.fi!

#### CHECKOUT

➤ Contact info

➤ Delivery address

Different billing address

➤ Billing address

➤ Payment

I would like to pay via

Paypal / Credit card

Invoice (contact us to enable)

Part	Specification	Qty	A	Total
	LASER SMIDYOtestbit.dxf Acrylic colored Black 3mm <span style="background-color: green; color: white; padding: 2px;">Material in stock</span>	1	6.63	11.63
<p>📌 Shipping is calculated based on the measurements and weight of your parts.</p>				
	SHIPPING DHL Globalmail PLUS		32 g	6.16
				17.79 €
				14.34 €

➤ Finalize your order

📌 The total delivery time depends on the material you've picked and where you live. Check the cart for the material delivery times.

📅 Material time: 7-12 days

🕒 Manufacturing time: 1-3 days

🚚 Shipping time: 3-6 days

Expect your parts within 12 to 21 days.

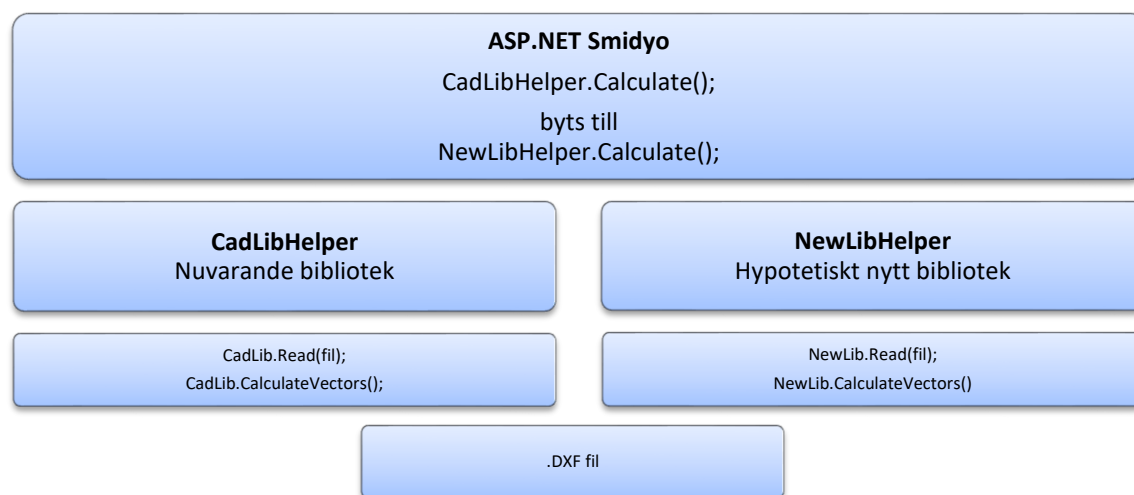
Feel free to leave a comment!

Check out with

Figur 13, Checkout-sidan.

### 3.3 Hjälpbibliotek med WoutWare CadLib 4.0 som grund

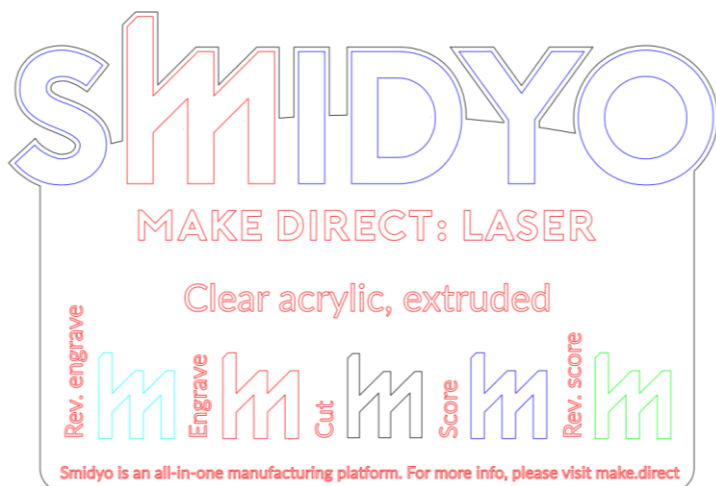
För att i framtiden vid behov kunna byta CAD-bibliotek skapades ett hjälpbibliotek som använder CadLib. Tanken är att man ska kunna uppgradera till ett annat CAD-bibliotek, och skriva ett nytt liknande hjälpbibliotek som använder det uppgraderade biblioteket. På detta vis skulle man slippa att göra stora förändringar i själva ASP.NET-applikationen.



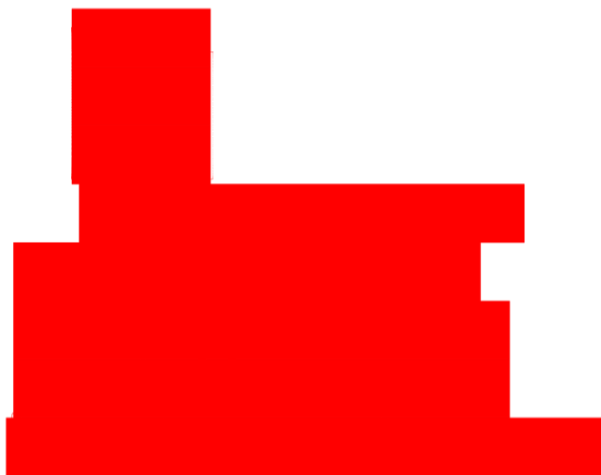
Figur 14, Illustration med pseudokod.

Biblioteket fungerar så att det öppnar en ritning med förbestämda färger för lasergraving, skärning och ristning på fram och baksida av biten. På grund av hur laserskäraren kör när den graverar kan man inte räkna priset direkt efter arean av gravingen. Man kan beskriva laserhuvudets färd som horisontella linjer uppifrån och ner, detta betyder att ifall det ska graveras en lodrät linje på höger och vänster sida av biten, måste lasern ändå köra hela vägen från graveringslinje till graveringslinje, för varje horisontell linje. Detta beskrivs bäst med bilder.

Figur 15 illustrerar kundens insatta ritning, de svarta linjerna är skärningslinjer, blå är ristningslinjer på framsidan, grön är ristningslinjer på baksidan, cyan är graving på baksidan och röd är graving på framsidan. I figur 16 har biblioteket plockat ut det som ska graveras, och räknat ut hur lasern kommer att köra när den graverar denna design, i figur 17 är bilderna på varandra.



Figur 15, Insatt ritning.



Figur 16, Illustration av hur lasern kommer att köra.

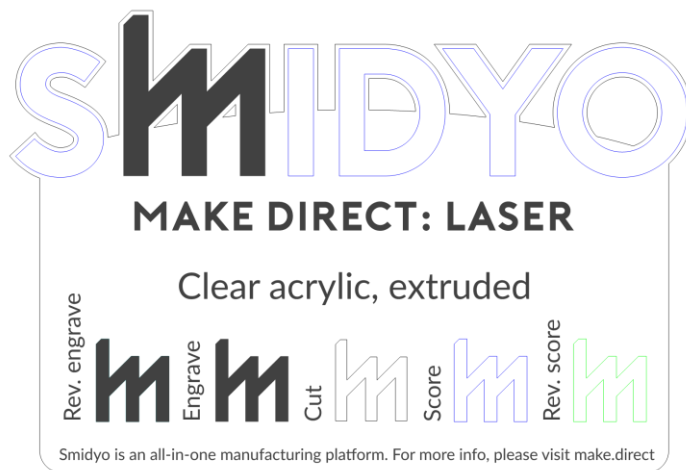


Figur 17, Illustration med transparens.

Detta är implementerat så att biblioteket delar upp designen i ett antal horisontella linjer, och mellan dessa linjer kan man sedan se vilken punkt som är längst till höger och vänster. Utgående från dessa linjer och punkter kan man beräkna en rektangulär area. Med hjälp av att ändra antal linjer får man mer eller mindre exakta areor. På detta vis kan man räkna ut nästan exakt hur lång tid det kommer att ta för lasermaskinen att gravera designen.

För beräkningen av längden på de vektorer som ska skäras används en del av WW.Math som är ett matematikbibliotek som följer med CadLib. Denna funktion konverterar alla DXF-entiteter till matematiska polygoner, och beräknar sedan längden på dessa.

För att kunden ska få se en förhandsvisning på hur designen kommer att se ut när den är graverad och skuren så konverteras alla graveringslinjer till en ifyllnad, som man sedan kan byta färg på med hjälp av CSS i ASP.NET-applikationen. Resultatet av ifyllningen blir då som figur 18.



**Figur 18, Ifylld gravering.**

### 3.3.1 Exportering av bilder

För att man med CSS kan kunna ändra färgen på de olika linjerna måste de exporteras som skilda PNG-filer där alla linjer och ifyllningar är röda. Totalt exporteras sex stycken bilder, en för framsidegravering, en för baksidegravering, en för framsideristning, en för baksideristning, en för skärning och slutligen en med alla linjer och ifyllnader som på Figur 18. Den sistnämnda används bl.a. i kundens kundvagn som en miniatyrbild (se figur 12 och 18).

För exporteringen används CadLibs inbyggda ImageExporter-klass. Nedan är ett exempel på hur en PNG-exportering kan se ut i kod.

#### Kodexempel 1. PNG-exportering med CadLib.

```
Bitmap bitmap = ImageExporter.CreateAutoSizedBitmap(EngravingModel, graphics,
Matrix4D.Identity, System.Drawing.Color.Transparent, maxSize);

using (stream = File.Create(path + ".png"))
    ImageExporter.EncodeImageToPng(bitmap, stream);
```

### 3.4 Databaser

Applikationen har två databaser, en för material och kundorderna samt en autogenerated av ASP.NET för framtida kundkontosystem. Material- och orderdatabasen innehåller data om materialens egenskaper, priser samt skär och graveringshastigheter. Den innehåller också samtliga uppgifter om en order, såsom kunduppgifter, priser och materialval samt information om de olika delarna kunden har beställt.

Att använda två databaser istället för en enda var ett medvetet val, på grund av framtida kundkontosystem lämnades ASP.NETs autogeneratede databas ifred för att senare kunna utvecklas skilt.

### 3.5 PayPal API

För tillfället är den enda betalningsmetoden i applikationen PayPal. PayPals API är relativt simpel, man skapar ett objekt och får en länk som man sedan dirigerar kunden till. När betalningen är utförd skickas kunden tillbaka till applikationen och applikationen får information om betalningen från PayPal, såsom transaktions ID och huruvida betalningen lyckades eller ej.

I framtiden ska företagskunder kunna beställa på faktura, men det implementeras inte i detta examensarbete.



### 3.6 Fraktberäkning

De två fraktmetoder som implementerades var DHL globalmail och Posti Priority. Både Posti och DHL har priszoner för olika länder, Posti har fyra och DHL har två. Dessa finns lagrade i databasen i en tabell, och utgående från vilket land kunden valt så räknas frakten ut.

Kraven för DHL globalmail är att paketet väger under 2kg, sammanlagda längden av bredd, höjd och längd är under 90cm, och längsta sidan maximalt 60cm.

Ifall paketet inte klarar av dessa krav så väljs istället Posti. Posti har liknande krav, med maximalvikt 30kg, längd plus omkrets högst 300cm och maximalt 200cm lång.

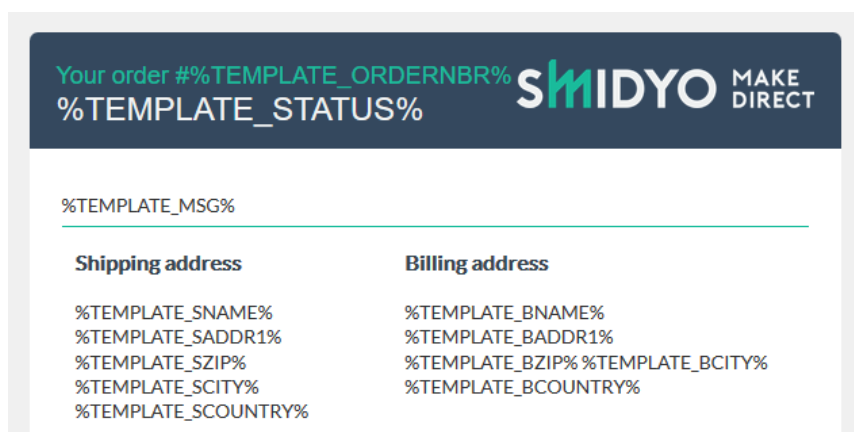
Om paketet inte går att skicka med någon av dessa fraktmetoder så ombeds kunden vänta med sin betalning, och kontaktas sedan av företaget för att komma överens om en annan fraktmetod.

### 3.7 E-post

När ordern är insatt får kunden ett e-postmeddelande med all information om ordern, såsom ordern status, t.ex. betalning mottagen eller betalningen misslyckades.

Ifall allt gick som det skulle får kunden ett meddelande att ordern är insatt, och när ordern har blivit ivägskickad får kunden ett till e-postmeddelande med orderns försändelsekod.

För att sända e-post via applikationen används klassen SmtpClient som finns i System.Net.Mail. För att generera meddelandet används en mall med en simpel Replace-funktion på informationen som ska skickas.



Figur 19, Del av e-postmall.

### 3.8 Lagring

För lagring av kunders ritningar och miniatyrbilder används Microsoft Azure's blob-lagring. Blob-lagring är ett enkelt sätt att lagra data i molnet, både privat och offentligt. Nedan är ett kodexempel på hur man kan ladda upp en fil till en blob i C#.

#### Kodexempel 2. Filuppladdning.

```
CloudStorageAccount storageAccount =  
CloudStorageAccount.Parse(CloudConfigurationManager.GetSetting("StorageConnecti  
onString"));  
  
CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();  
CloudBlobContainer container =  
blobClient.GetContainerReference(blobcontainername);  
  
CloudBlockBlob blob = container.GetBlockBlobReference(blobname);  
  
using (var fileStream = System.IO.File.OpenRead(filepath))  
{  
    blob.UploadFromStream(fileStream);  
}
```

Totalt exporteras sex stycken bilder från applikationen, men endast en av dem sparas på Azure Blob-lagringen. Denna sparas för att uppdragsgivaren enkelt ska kunna se hur biten ser ut utan att behöva ladda ner och öppna ritningen först.

I administratörspanelen för kundorder kan uppdragsgivaren ladda ner kundens ritningar från Azure Blob-lagringen.

#### Kodexempel 3. Nerladdning av fil från Blob.

```
CloudStorageAccount storageAccount =  
CloudStorageAccount.Parse(CloudConfigurationManager.GetSetting("StorageConnecti  
onString"));  
  
CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();  
CloudBlobContainer container =  
blobClient.GetContainerReference(blobcontainername);  
CloudBlockBlob blob = container.GetBlockBlobReference(blobname);  
  
var fileStream = new MemoryStream();  
blob.DownloadToStream(fileStream);
```

## 3.9 Administrationspanel

För att uppdragsgivaren ska kunna se order justera priser och parametrar o.s.v. krävs en slags administrationspanel. Denna är gjord så att uppdragsgivaren kan logga in med ett redan insatt administratörskonto. När man är inloggad som administratör syns ett extra menyalternativ, på den sidan kan man ändra olika parametrar för applikationen, samt ändra priser, sätta in och ta bort material, skicka iväg en order m.m.

Här kan administratören också ladda ner ritningarna från Azure Blob-lagringen för en order, så att tillverkningen kan börja. När tillverkningen av en orders alla bitar är klar och ordern ivägskickad, kan en försändelsekod läggas in. När administratören lagt in en försändelsekod, skickas automatiskt ett e-postmeddelande till orderns kund med koden och annan eventuell information.

### 3.9.1 Loggning av fel

För att enklare kunna reda ut eventuella buggar i systemet vid körning på Azure, så har applikationen ett loggningssystem som simpelt sparar alla ohanterade undantag (eng. Exception) till en textfil på Azure Blob-lagringen. Denna textfil innehåller klockslag och information om ett undantag. ”ExceptionLogging” klassen är en egen hjälpklass, som skapar en textfil utgående från undantaget, och sparar det till Azure. Se figur 20.

#### Kodexempel 3. Modifiering i Global.asax.cs.

```
protected void Application_Error(object sender, EventArgs e)
{
    HttpException lastErrorWrapper = Server.GetLastError() as HttpException;
    Exception lastError = lastErrorWrapper;
    ExceptionLogging.SendErrorToBlob(lastError);
}
```

## Logs

log 23-03-2017--13-42-36-768

```
-----Exception Details on 23/03/2017 13:42:36-----
-----
Log Written Date: 23/03/2017 13:42:36
Error Line No : ine 207
Error Message: NullReferenceException
Exception Type: System.NullReferenceException
Error Location : Object reference not set to an instance of an object.
Error Page Url: http://localhost:1935/Laser/Make/th
User Host IP:
-----*End*-----
Back
```

Figur 20, Exempel på log-fil.

## 4 Resultat och diskussion

Resultatet blev en fullt fungerande ASP.NET-webbapplikation, som förvisso kan behöva lite testning och finslipning innan den tas helt i bruk.

### 4.1 Resultat

Uppdragsgivaren är så här långt nöjd med resultatet, även om det inte satts i bruk ännu. Eftersom att en stor del av denna applikation förlitar sig på att användaren följer anvisningarna för hur han ska rita sin design, kommer det förmodligen att uppstå en del fel i programvaran ännu som skulle varit svåra att förutse.

### 4.2 Vidareutveckling

Tanken var också att en kund ska kunna registrera sig på hemsidan och på så sätt kunna se sina gamla order, och vid behov då enkelt kunna beställa samma sak på nytt. Men eftersom att detta skulle ha blivit för mycket för ett examensarbete så kommer detta att implementeras senare. Detta är egentligen endast grunden och första delen till en större plattform för liknande tillverkningsmetoder, i framtiden ska det också finnas tillval för UV-printning på de laserskurna delarna, 3D-printning m.m. Detta är något som jag förmodligen kommer att jobba på i framtiden.

### 4.3 Diskussion

Detta var ett väldigt bra och intressant examensarbete. Att kunna vara med och se på hur tillverkningen fungerar var också till stor hjälp, och gav en insikt i hur applikationen kunde förbättra beställningssystemet samt hur den behöver fungera. Nära och snabb kommunikation med uppdragsgivaren var också en viktig del, det gjorde att de flesta problem och frågor man stötte på kunde lösas nästan direkt, och inte behövde spendera så mycket tid på att fundera.

Först tyckte jag att det lät som ett ganska litet och relativt lättlöst problem, men med tiden visade det sig att det fanns flera, lite mer invecklade problem att lösa. Sist och slutligen är jag nöjd med vad jag åstadkommit, mitt examensarbete kommer att finnas på internet för alla att se och använda världen över, för åtminstone ett antal år framåt.

## 5 Källförteckning

.dwg, 2016. [Online]

<https://en.wikipedia.org/wiki/.dwg>

[Använd 20 12 2016].

ASP.NET MVC, 2016. [Online]

[https://en.wikipedia.org/wiki/ASP.NET MVC](https://en.wikipedia.org/wiki/ASP.NET_MVC)

[Använd 21 12 2016].

AutoCAD DXF, 2016. [Online]

[https://en.wikipedia.org/wiki/AutoCAD DXF](https://en.wikipedia.org/wiki/AutoCAD_DXF)

[Använd 20 12 2016].

CNC Design, u.d. *CNC Design*. [Online]

<http://www.cnc-design.fi/wp/>

[Använd 20 12 2016].

Common Language Infrastructure, 2017. [Online]

[https://en.wikipedia.org/wiki/Common Language Infrastructure](https://en.wikipedia.org/wiki/Common_Language_Infrastructure)

[Använd 13 3 2017].

Help:Vector graphics tutorial, 2017. [Online]

[https://commons.wikimedia.org/wiki/Help:Vector\\_graphics\\_tutorial](https://commons.wikimedia.org/wiki/Help:Vector_graphics_tutorial)

[Använd 18 2 2017].

Microsoft Azure, u.d. *App Service Priser*. [Online]

<https://azure.microsoft.com/sv-se/pricing/details/app-service/>

[Använd 20 2 2017].

Microsoft Visual Studio, 2016. [Online]

[https://en.wikipedia.org/wiki/Microsoft Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio)

[Använd 20 12 2016].

Microsoft, 2013. *Entity Framework*. [Online]

[https://msdn.microsoft.com/en-us/library/gg696172\(v=vs.103\).aspx](https://msdn.microsoft.com/en-us/library/gg696172(v=vs.103).aspx)

[Använd 18 2 2017].

Microsoft, 2016. *SQL Server 2016 : Microsoft*. [Online]

<https://www.microsoft.com/en-us/sql-server/sql-server-2016>

[Använd 21 12 2016].

Microsoft, 2017. *ADO.NET Overview*. [Online]

[https://msdn.microsoft.com/en-us/library/h43ks021\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/h43ks021(v=vs.110).aspx)

[Använd 18 2 2017].

Microsoft, 2017. *Microsoft Azure*. [Online]

<https://azure.microsoft.com/sv-se/>

[Använd 27 3 2017].

Microsoft, u.d. *ADO.NET Architecture*. [Online]

[https://msdn.microsoft.com/en-us/library/27y4ybxw\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/27y4ybxw(v=vs.110).aspx)

[Använd 18 2 2016].

Microsoft, u.d. *Introduction to ASP.NET Web Forms*. [Online]  
<https://www.asp.net/web-forms/what-is-web-forms>  
[Använd 21 12 2016].

Microsoft, u.d. *Learn About ASP.NET Web Pages*. [Online]  
<https://www.asp.net/web-pages/overview/getting-started>  
[Använd 21 12 2016].

MSDN, 2015. *Introduction to the C# Language and the .NET Framework*. [Online]  
<https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx>  
[Använd 20 12 2016].

Vektorgrafik, 2015. [Online]  
<https://sv.wikipedia.org/wiki/Vektorgrafik>  
[Använd 20 12 2016].

WoutWare, u.d. *WoutWare*. [Online]  
<https://www.woutware.com>  
[Använd 20 12 2016].