

Namn: Bloggsystem 1.0

Licens: GPL

Applikationsstorlek: 68,2 MB

Nedladdat från: [Annice.se](http://Annice.se)

# Ett databasdrivet bloggsystem i C# med ASP.NET och Entity Framework

Av: Annice Strömberg

Skapat och publicerat: 2016

# Innehåll

<b>1 Allmän information</b>	1
<b>2 Systembeskrivning</b>	2
2.1 Entiteter	2
2.2 Funktioner och flödesbeskrivningar	4
<b>3 Implementation och installation</b>	8
<b>4 Gränssnitt</b>	11
<b>5 Applikationsexekvering och eventuella fel</b>	15

## 1 Allmän information

Webbapplikationen avser ett databasdrivet bloggsystem. I detta bloggsystem kan en admin tillika skribent logga in och skriva, redigera och radera blogginlägg, bloggkommentarer och bloggkategorier. Vidare har admin möjlighet att redigera sina användaruppgifter, såsom användarnamn, för- och efternamn, användarmejl samt användarlösenord. En admin har vid glömt lösenord också möjlighet att få ett nytt lösenord utmejlats. Gästbesökare kan i nästa tur läsa och söka fram publicerade blogginlägg. Ytterligare kan gäster skriva kommentarer till olika blogginlägg.

Webbapplikationen i fråga är vidare av licenstypen [GPL](#) (General Public License), vilket innebär att applikationen är gratis och får användas och modifieras helt efter egna behov. Vid eventuell vidareredistribuerings av skriptet gäller dock att det tillgängliggörs för andra under samma licensvillkor. Det ska även tilläggas att detta är med undantag för vad som gäller avseende den integrerade HTML-editorfunktionen som tillhandahålls via TinyMCE.

## 2 Systembeskrivning

*I detta kapitel beskrivs de entiteter som bloggsystemet innefattar med dessas relationer och attribut. Vidare beskriver kapitlet användargränssnittets huvudsakliga flöden avseende sidor och händelser. Kapitlet beskriver också applikationens olika funktioner som systemet tillhandahåller gästbesökare respektive admin.*

### 2.1 Entiteter

Bloggsystemet innefattar fyra relaterade entiteter: "User", "Entry", "Category" samt "Comment". "User" inkluderar uppgifter för en admin/skribent och består av attributen: "userID", "username", "firstname", "lastname", "mail" samt "password". Entiteten "Entry" berör i nästa tur blogginläggsuppgifter, liksom metainformation för varje inlägg, och består sålunda av attributen: "entryID", "title", "date", "keywords", "description", "entry", "categoryID" och "userID". Ytterligare utgörs entiteten "Category" av attributen "categoryID" samt "category". Avslutningsvis består entiteten "Comment" av kommentarsuppgifter, vilka utgörs av attributen: "commentID", "name", "mail", "date", "website", "comment", "answer", samt "entryID".

De regler som gäller för entitetsrelationerna handlar vidare om en-till-många-relationer (1:N-relationer) och lyder enligt följande: varje inlägg (entry) kan ha minst och max en skribent (user); en skribent behöver dock inte ha skrivit något inlägg. Varje inlägg har i nästa tur minst och max en kategori (category), men en kategori måste inte relatera till ett inlägg. Vidare kan varje kommentar (comment) tillhöra minst och max ett inlägg, men ett inlägg behöver inte ha några kommentarer.

Det givna ER-schemat för de beskrivna entiteterna är följande:

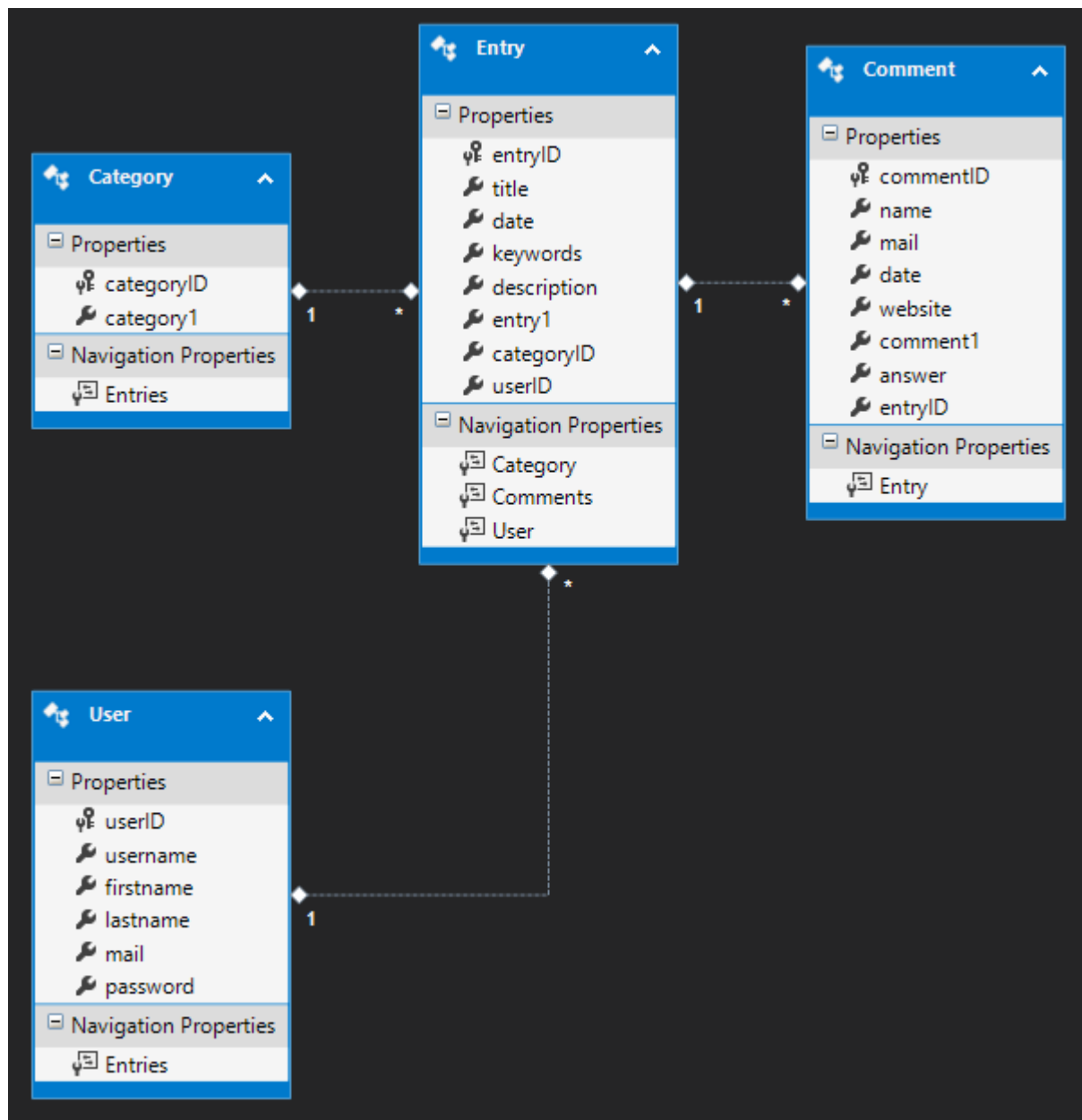


Bild 1: ER-schema avseende bloggsystemets entitetsrelationer.

## 2.2 Funktioner och flödesbeskrivningar

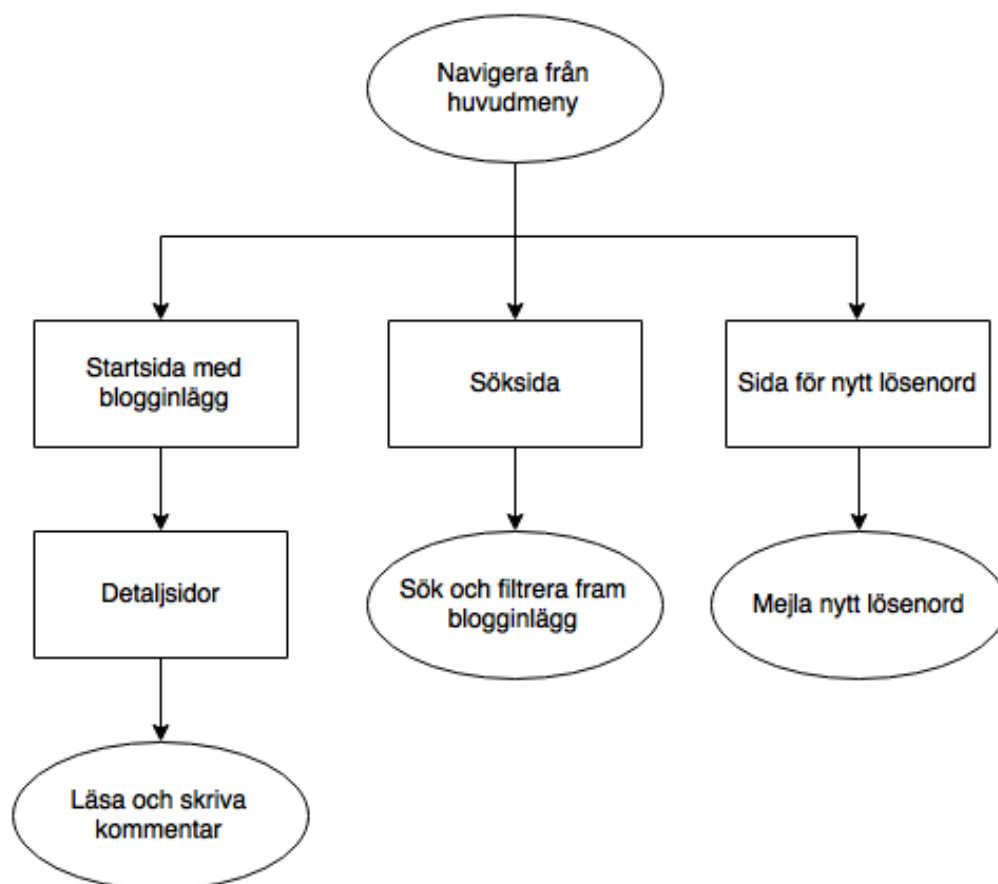
Webbapplikationen tillhandahåller följande generella funktioner:

- Bloggens layoutmall baseras på rutsystemet Bootstrap och tillhandahåller således ett responsivt användargränssnitt.
- Med hjälp av TinyMCE:s WYSIWYG HTML-editor ges skribenten full redigeringsmöjlighet över sina blogginlägg.
- För sökmotoroptimering används metataggar enligt Dublin Core avseende blogginläggens detaljsidor. Metataggarna innehåller vidare data avseende specifika blogginläggs titel, kategori, nyckelord, beskrivning samt datum. Titel, kategori, nyckelord och beskrivning kan anges manuellt av skribenten vid skapande och redigering av blogginlägg. Vid skapande av inlägg skapas automatiskt dagens datum, vilket sedan kan hanteras manuellt vid redigering av blogginlägg.
- Formulären är utformade i enlighet med WCAG med fieldset för textinnehåll och metadata.
- Validering är tillämpat på formulärsfält för att upprätthålla attributsregler och konsistent data.
- En välfungerande tabb-ordning finns på samtliga sidor som berör inmatning.
- Samtlig HTML-kod uppfyller W3C:s rekommendationer för valid HTML5.
- Inloggningssystemet baseras på sessioner.
- Användarlösenordet lagras krypterat enligt MD5 (Message-Digest Algorithm 5).
- Admin har vid bortglömt lösenord möjlighet att få ett nytt lösenord utskickat till sin användar mejl.
- Vid de fall det finns fler än tio blogginlägg pagineras sidor med länkar för första, sista, samt nästa och föregående blogginlägg.
- Systemet tillhandahåller en sökfunktion för att kunna söka och filtrera fram existerande blogginlägg, vilka vidare kan sorteras efter rubriker, kategorier, datum samt inläggsinnehåll.

Genom applikationens webbgränssnitt kan admin och gäster nå följande offentliga sidor och funktioner:

- Startside som visar de senaste blogginläggen i fallande ordning på datum.
- Detalsidor för varje blogginlägg med tillhörande kommentarer, där man också kan skriva och lägga till nya kommentarer.
- Sida för att söka och filtrera fram blogginlägg på titlar, kategorier, datum och inläggsinnehåll. Vidare kan blogginläggen sorteras i alfabetisk ordning på titlar och kategorier, samt i fallande ordning på datum.
- Admin har via offentlig vy också tillgång till en sida för att få ett nytt lösenord mejlat till den mejladress som är kopplad till användarkontot.

Hierarkin för offentliga sidor och dessas händelser utifrån GUI (Graphical User Interface) kan illustreras enligt följande flödesdiagram och bild 2:



*Bild 2: Flödesdiagram för offentliga sidor.*

Genom adminsektionen kan skribenten vidare nå följande fyra huvudsidor:

- Startside som listar befintliga blogginlägg med länkar för CRUD (Create Read Update Delete).
- Sida som listar befintliga kommentarer med UD-länkar.
- Sida som listar befintliga kategorier med CUD-länkar.
- Sida för att kunna redigera sina användaruppgifter avseende användarnamn, för- och efternamn, mejladress, samt lösenord.

Genom CRUD-länkarna för blogginläggen kan admin vidare nå följande sidor och funktioner:

- Sida för att skapa ett nytt blogginlägg.
- Sida för att se detaljer för ett specifikt blogginlägg.
- Sida för att redigera ett specifikt blogginlägg.
- Sida för att radera ett specifikt blogginlägg.

Genom sidan som listar befintliga kommentarer kan admin nå följande sidor och funktioner:

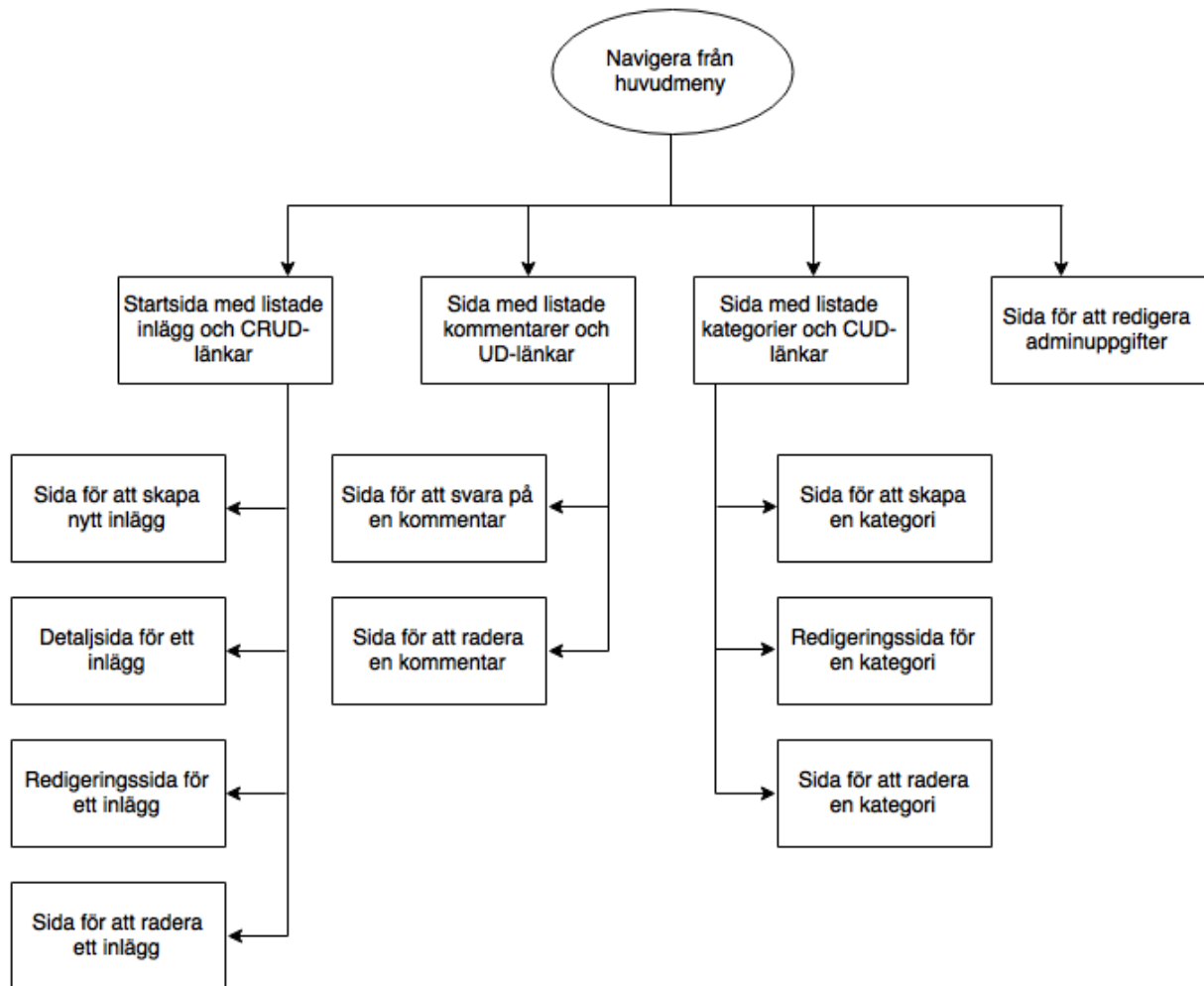
- Sida för att hantera svar till en specifik kommentar.
- Sida för att radera en specifik kommentar.

Genom sidan som listar befintliga kategorier kan admin vidare nå följande sidor och funktioner:

- Sida för att skapa en ny kategori.
- Sida för att redigera en specifik kategori.
- Sida för att radera en specifik kategori.



Hierarkin för adminsidor utifrån GUI kan illustreras enligt följande flödesdiagram och bild 3:



*Bild 3: Flödesdiagramm för adminsidor.*

### 3 Implementation och installation

Bloggsystemet är utvecklat i Windows-miljö med hjälp av programvarorna SQL Server 2014 samt Visual Studio Express 2015 for Web. Back-end-utvecklingen baseras vidare på databasspråket Transact-SQL, samt applikationsspråket C# utifrån designmönstret MVC baserat på ramverken ASP.NET och EF. Front-end-utvecklingen baseras i nästa tur på ramverket Bootstrap med kodspråken HTML5, CSS och JavaScript med jQuery.

För att köra applikationen förutsätter det att ni har ovannämnda programvaror installerade på en Windows-plattform. Om ni inte har dessa program installerade finns de tillgängliga att ladda ner för gratis användning via följande länkar:

- [Visual Studio Express 2015 for Web](#)
- [SQL Server Management Studio](#)

I detta kapitel kommer jag dock inte gå igenom hur man skapar en databas eller användarkonto via SQL Server eller Visual Studio. Jag ger dock en beskrivning av nödvändig SQL-kod för att ge bättre förståelse för vad som ligger till grund för applikationens entiteter, samt gällande installationskrav.

**Obs!** SQL-koden jag kommer beskriva på nästkommande sida finns för övrigt bifogat som fil, med ytterligare insert-satser för exempeldata, vid namn "SQL.sql". Den filen kan med fördel öppnas upp och exekveras direkt i SQL Server. Glöm dock inte att först ersätta värdena "YourDatabaseName" med ert eget databasnamn innan exekvering. I insert-satsen för att ange standardvärde för en admin/skribent ska förstås också värdena "YourUsername", "YourFirstname", "YourLastname", "yourmail@mail.se", samt "YourPassword" först ersättas med era egna önskade värden.

Den SQL-kod som gäller för att implementera databastabellerna via SQL Server är följande. Observera dock att "YourDatabaseName" alltså ska ersättas med ert eget databasnamn:

```
USE YourDatabaseName
GO

/* Create table for users. */
CREATE TABLE Users (
  userID int primary key identity(1,1),
  username VARCHAR(30) unique NOT NULL,
  firstname VARCHAR(30),
  lastname VARCHAR(30),
  mail VARCHAR(50) unique NOT NULL,
  password VARCHAR(32)
);

/* Create table for entry categories. */
CREATE TABLE Categories (
  categoryID int primary key identity(1,1),
  category VARCHAR(50) unique NOT NULL
);

/* Create table for entries. */
CREATE TABLE Entries (
  entryID int identity(1,1) primary key,
  title varchar(50) NOT NULL,
  date DATETIME,
  keywords VARCHAR(200),
  description VARCHAR(300),
  entry VARCHAR(8000) NOT NULL,
  categoryID int NOT NULL,
  CONSTRAINT fk_category FOREIGN KEY (categoryID) REFERENCES
Categories(categoryID)
  ON DELETE CASCADE,
  userID int NOT NULL,
  CONSTRAINT fk_username FOREIGN KEY (userID) REFERENCES Users(userID)
  ON DELETE CASCADE
);

/* Create table for entry comments. */
CREATE TABLE Comments (
  commentID int identity(1,1) primary key,
  name VARCHAR(30),
  mail VARCHAR(50) NOT NULL,
  date DATETIME,
  website VARCHAR(100),
  comment VARCHAR(300) NOT NULL,
  answer VARCHAR(300),
  entryID int NOT NULL,
  CONSTRAINT fk_entryID FOREIGN KEY (entryID) REFERENCES Entries(entryID)
  ON DELETE CASCADE
);
```

Efter att ovanstående SQL-kod exekverats ska följande insert-sats exekveras för att lägga till önskade standarduppgifter för en admin/skribent. Ni kan sedan logga in i bloggen med de användaruppgifter ni anger i nedanstående kod. Uppgifterna kan sedan ändras när applikationen körs via adminsidan "Redigera användaruppgifter".

```
/* Insert default value for a user. */
insert users values('YourUsername', 'YourFirstname', 'YourLastname',
'yourmail@mail.se', HASHBYTES('MD5', 'YourPassword'));
```

För att sedan få databasanslutningen att fungera behöver också applikationens webconfig-fil konfigureras enligt följande steg:

1. Öppna filen "Webconfig.XML" (i en lämplig keditor för bättre översikt och radbrytning). Filen kan hittas i nedladdad och upppackad applikationsmapp enligt följande sökväg: **Bloggsystem1.0>Bloggsystem>Web**
2. Leta upp följande XML-taggar under "connectionStrings" i filen och ändra markerade områden till ert eget datornamn samt databasegenskaper:

```
<!-- Start: Database connection settings -->
<add name="TechniqueContext" connectionString="Data
Source=YourComputername\SQLEXPRESS;Initial Catalog=databasename;User
id=databasename;Password=databaseword;Integrated Security=True"
providerName="System.Data.SqlClient" />

<add name="BlogContext"
connectionString="metadata=res://*/Models.BlogModel.csdl|res://*/Models.Blo
gModel.ssdl|res://*/Models.BlogModel.msl;provider=System.Data.SqlClient;pro
vider connection string=&quot;data
source=YourComputername\SQLEXPRESS;initial catalog=databasename;user
id=databasename;password=databaseword;MultipleActiveResultSets=True;App
=EntityFramework&quot;;" providerName="System.Data.EntityClient" />
<!-- End: Database connection settings -->
```

För att få applikationens mejlfunktion att fungera för att kunna skicka ut ett nytt lösenord till admin behöver ytterligare konfigurering göras avseende en utsändarmejl, vilket görs i samma webconfig-fil som nämnts ovan. Observera att utsändarmejl inte är samma som applikationens användarmejl. Utsändarmejl används alltså för att kunna skicka ett mejl till en användarmejl via utsändarmejlens SMTP-inställningar. Användarmejl är med andra ord den mejladress som angetts för ett adminkonto som alltså erhåller ett utmejl lösenord.

**Obs!** Applikationen i fråga har testkörts med en Gmail som utsändarmejl. Om ni själva vill använda en Gmail för ändamålet i fråga behöver ni bara ändra till er egen mejladress med berört lösenord i koden nedan, det vill säga låta övriga SMTP-värden stå kvar. I annat fall kan ni också behöva ändra dessa. Om en Gmail används som utsändarmejl måste dock också "[Less secure apps](#)" aktiveras för berört Gmail-konto. Webconfig-filen kan sedan konfigureras enligt följande steg:

1. Leta upp följande XML-taggar under "appSettings" i filen och ändra markerade områden till de mejlegenskaper som ska gälla för en utsändarmejl:

```
<!-- Start: Mail settings -->
<add key="MailSender" value="yourmail@gmail.com" />
<add key="MailSenderPass" value="yourmailpassword" />
<add key="MailSenderPort" value="587"/>
<add key="MailSenderHost" value="smtp.gmail.com"/>
<!-- End: Mail settings -->
```

## 4 Gränssnitt

För att illustrera webbgränssnittet för bloggsystemet har jag tagit några skärmdumpar på systemet i fråga. Nedanstående bild illustrerar den offentliga startsidan med inlagd exempeldata för bloggsystemet:

Bloggsystem   Sök inlägg      [Glömt lösenord?](#)

---

### Lorem ipsum

**Av:** admin | **Datum:** 2016-11-04 | **Kategori:** Allmänt

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[Kommentera](#)

---

### Data och IT

**Av:** admin | **Datum:** 2016-10-04 | **Kategori:** Data & IT

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

[Kommentera](#)

---

### Musikinlägg

**Av:** admin | **Datum:** 2016-09-15 | **Kategori:** Musik

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

*Bild 4: Skärmdump av bloggsystemets startsida.*

Ytterligare illustrerar nedanstående bild en offentlig sida för att söka och filtrera fram blogginlägg:

Bloggsystem   Sök inlägg      [Glömt lösenord?](#)

---

### Sök blogginlägg

Sök och filtrera fram blogginlägg på rubrik, inläggsinnehåll eller kategori genom att ange önskade sökord i formulärsfältet nedan.

Inläggsrubrik:	Inlägg:	Kategori:	Datum:
Musikinlägg	Lorem ipsum dolor sit amet, consectetur adipiscing...	Musik	2016-09-15

Ett bloggsystem i C# med ASP.NET och Entity Framework av Annice Strömberg, 2016.

*Bild 5: Skärmdump av offentlig sida för att söka och filtrera fram blogginlägg.*

Nedanstående bild illustrerar i nästa tur en offentlig detaljsida för ett blogginlägg:

Bloggsystem

Sök inlägg

Användare

Lösenord

Logga in

Glömt lösenord?

Rubrik: Lorem ipsum

Av: admin | Datum: 2016-11-04 | Kategori: Allmänt

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Skriv en kommentar

Av:

Mejladress: \*

Webbadress:

http://

Kommentar: \*

Posta

Kommentarer

Mejladress: [gast3@mail.se](mailto:gast3@mail.se) | Datum: 2016-11-06

Lorem ipsum dolor sit amet.

*Bild 6: Skärmdump av en offentlig detaljsida för ett blogginlägg.*

Nedanstående bild illustrerar i nästa tur en indexsida för admin för att hantera blogginlägg:

Bloggsystem

Hantera blogginlägg

Hantera kommentarer

Hantera kategorier

Redigera användaruppgifter

Sök inlägg

Logga ut

Hantera blogginlägg

Ange sökord.

Sök

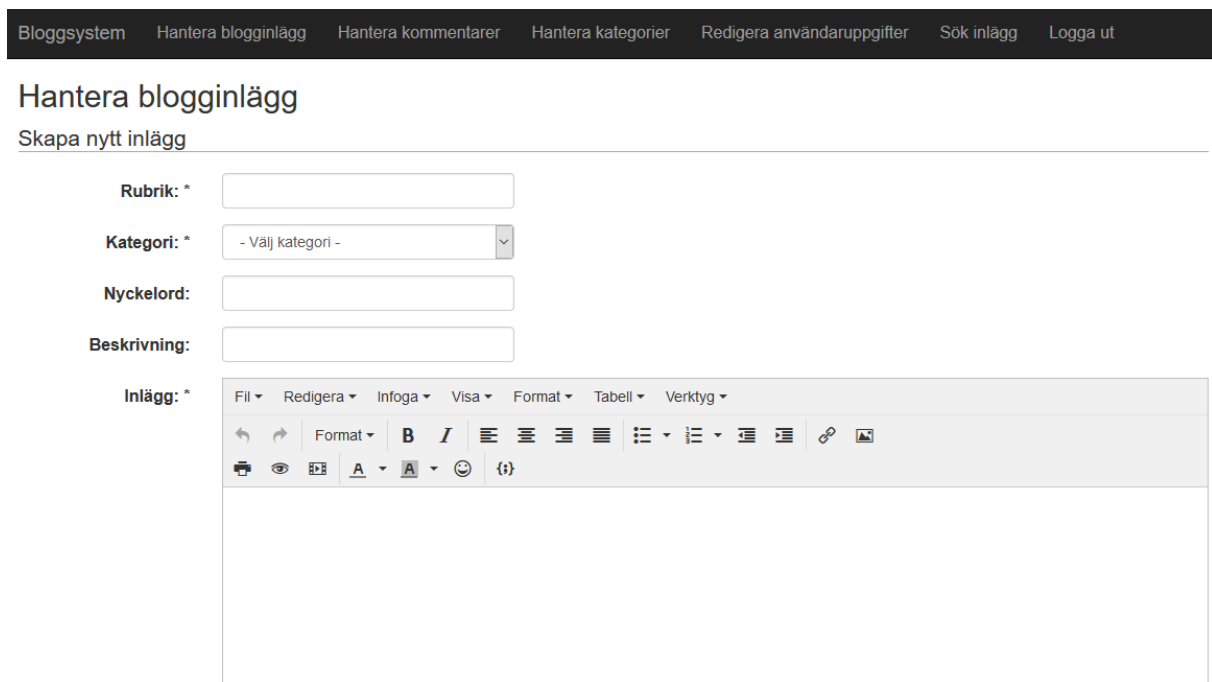
[Skapa nytt inlägg](#)

ID:	Rubrik:	Datum:	Nyckelord:	Beskrivning:	Kategori:	
1	Lorem ipsum	2016-11-04	Lorem, ipsum.	Ett lorem ipsum-inlä...	Allmänt	<a href="#">Redigera</a>   <a href="#">Detaljer</a>   <a href="#">Radera</a>
2	Data och IT	2016-10-04	Data, IT, internet.	Ett inlägg om data o...	Data & IT	<a href="#">Redigera</a>   <a href="#">Detaljer</a>   <a href="#">Radera</a>
3	Musikinlägg	2016-09-15	Musik.	Ett inlägg om musik.	Musik	<a href="#">Redigera</a>   <a href="#">Detaljer</a>   <a href="#">Radera</a>

Ett bloggsystem i C# med ASP.NET och Entity Framework av [Annice Strömberg](#), 2016.

*Bild 7: Skärmdump av en adminsida för att hantera blogginlägg.*

Nedanstående skärmdump illustrerar en adminsida för att skriva ett nytt blogginlägg:



*Bild 8: Skärmdump av en adminsida för att skapa ett nytt blogginlägg.*

Nedanstående bild illustrerar en adminsida för att svara/redigera svar till ett blogginlägg:

*Bild 9: Skärmdump av en adminsida för att hantera svar på en kommentar.*

Nedanstående bild illustrerar vidare en adminsida för att redigera användaruppgifter:

Bloggssystem   Hantera blogginlägg   Hantera kommentarer   Hantera kategorier   Redigera användaruppgifter   Sök inlägg   Logga ut

## Redigera användaruppgifter

Redigera användaruppgifter

Användarnamn: \*

Förnamn:

Efternamn:

Mejladress: \*

Ändra lösenord

Nytt lösenord:

Ett bloggssystem i C# med ASP.NET och Entity Framework av [Annice Strömberg](#), 2016.

*Bild 10: Skärmdump av en adminsida för att redigera användaruppgifter.*



## 5 Applikationsexekvering och eventuella fel

När implementationen av databastabellerna och nödvändiga konfigurationer är utförda är applikationen redo att användas. Jag kommer inte här gå igenom hur man gör applikationen publik via en publik webbserver som t.ex. [Azure](#), men för att se hur applikationen fungerar under körning lokalt kan den öppnas upp via Visual Studios menyval: **File>Open Project...** För att sedan läsa in applikationen kan ni navigera till den nedladdade, uppackade mappen "Bloggsystem1.0" och dubbelklicka på filen "Bloggsystem" av filtypen "Microsoft Visual Studio Solution". Avslutningsvis kan bloggsystemet exekveras via play-knappen för önskad webbläsare uppe i Visual Studios menypanel.

Om applikationen inte kan köras så kontrollera först att ni inte gjort några missar avseende databasimplementationen, samt konfigurationen i webconfig. Försäkra er också om att SQL Server körs under Windows-tjänsterna. Om ni för övrigt skulle upptäcka eventuella buggar i programmet under körning får ni gärna kontakta mig via mejl: [info@annice.se](mailto:info@annice.se)