

Indledning til systemudvikling

Systemudvikling består af en kombination af at finde frem til kravene til det kommende system, analysere sammenhængene i, designe, programmere og teste det.

Groft deles systemudvikling op i tre discipliner:

- **Analyse**

Går ud på at beskrive, hvilke krav systemet skal opfylde. Dette indebærer dels en analyse af systemets problemområde/anvendelsesområde/område, dvs. den del af omverdenen, som systemet skal afspejle (indeholde data om), dels en analyse af systemets anvendelsesområde, dvs. den del af omverdenen, der skal bruge systemet (give input og modtage output).

Analysen udmøntes overvejende i en kravspecifikation og andre dokumenter som grundlag for designet.

- **Design**

Skal med baggrund i analysen finde frem til en hensigtsmæssig opdeling af systemet med hensyn til dels dataene om problemområdet (datamodel), dels grænsefladen til brugerne og andre systemer. I designet indgår både en beskrivelse af, hvilke komponenter systemet skal opdeles i, og hvordan de enkelte komponenters grænseflade med hinanden skal fungere.

Designet skal danne grundlag for programmeringen.

- **Programmering**

Går ud på at implementere designet i kode og teste den

Kilde: <https://da.wikipedia.org/wiki/Systemudvikling>

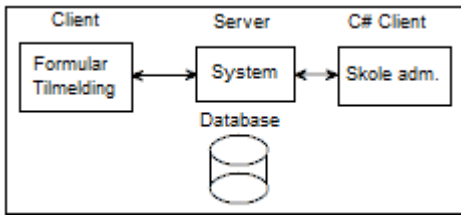
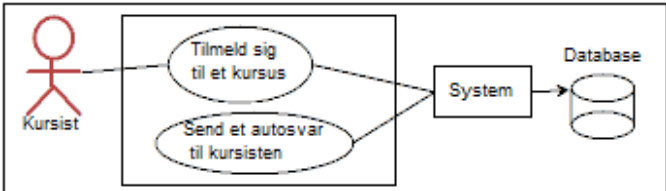
Aftenskole tilmeldings case:

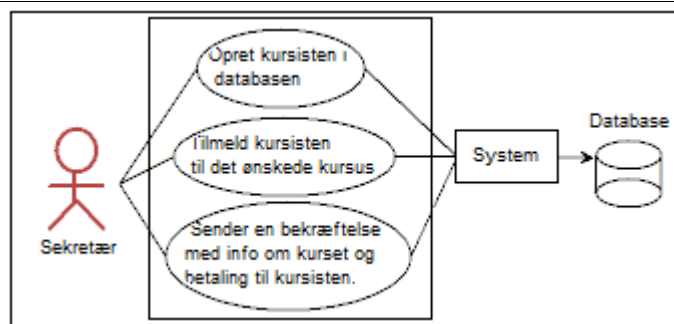
En nystartet aftenskole ønsker at få udviklet et simpelt tilmeldings system. De forslår selv at de vil bruge deres egen hjemmeside til at rumme den løsning.

De har talt om der skulle være en tilmeldings formular på hjemmesiden og en tilmeld knap.

Proceduren er når en kursist har sendt en tilmelding skal kursist modtage et autosvar at den er blevet modtaget. Sekretæren vil så oprette kursisten og sende et svar med information om tid og sted for kursuset og betaling. Der skal bruges en database til det system.

System udvikling trin for trin:

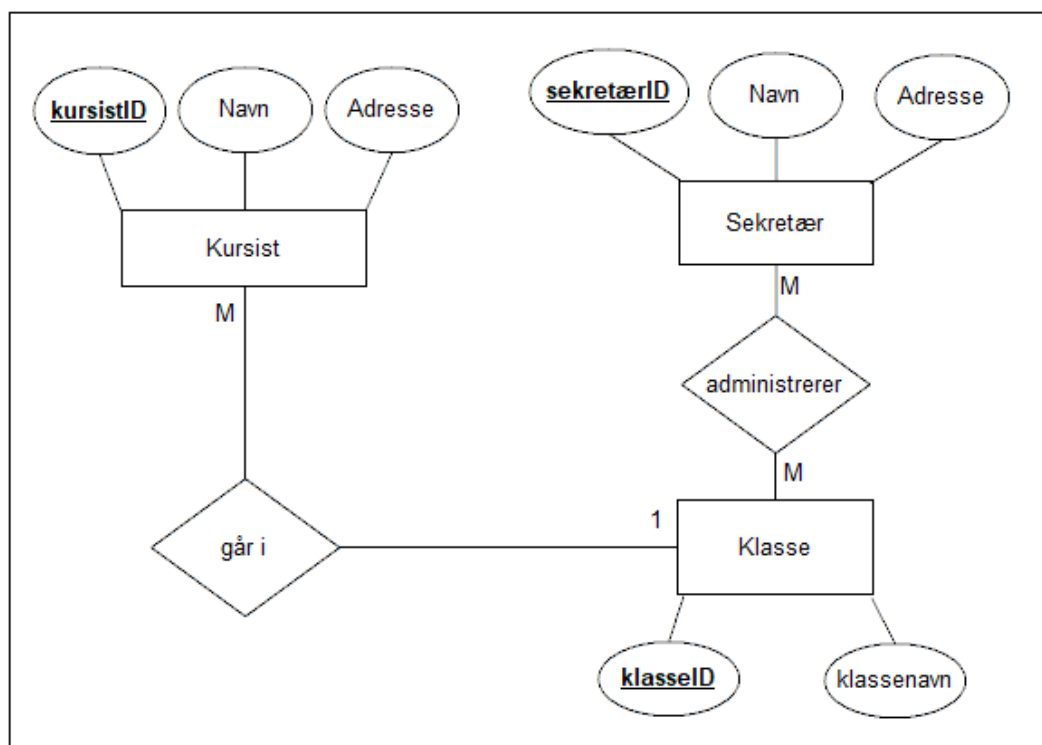
Case beskrivelse: (opgave beskrivelse)	<p>Afklaring: Læs og forstå hvad opgaven går ud, stil spørgsmål til kunden.</p> <p>Afgræns opgaven: Spørg ind til hvad kunden skal bruge af funktionalitet og hvilke informationer/data kunden/kursisten skal bruge.</p>
Hvordan kan kundens opgave løses?	<p>Lave en brainstorm hvor alle deltagere bidrager til en overordnet løsnings model.</p>  <p>The diagram shows a Client box containing 'Formular Tilmelding', a Server box containing 'System', and a C# Client box containing 'Skole adm.'. Arrows indicate bidirectional communication between Client and Server, and between Server and C# Client. Below the Server box is a Database cylinder icon.</p> <p>Løsning består af følgende:</p> <ul style="list-style-type: none">• Hjemmeside med en tilmeldingsformular• En administrativ grænseflade i C#• En server med en MSSQL database
Identificer ud fra den overordnet løsnings model de enkelte objekter.	<p>Objekter: Server med en database, en brugergrænseflade, en kursist, en sekretær</p>
Adfærd: Hvordan er processen med at en kursist tilmelde sig? Hvad skal sekretæren foretage sig? Hvad skal databasen udføre?	<p>Udarbejd Use Cases.</p> <p>Kursist: Udfylder og sender sin tilmelding vha. en formularen og modtager et autosvar.</p>  <p>The diagram shows an actor 'Kursist' connected to two use cases: 'Tilmeld sig til et kursus' and 'Send et autosvar til kursisten'. Both use cases are connected to a 'System' box, which is in turn connected to a 'Database' cylinder.</p> <p>Sekretæren: Opretter kursisten, tilmelder kursisten til det ønskede kursus og sender en bekræftelse med info om kurset og betaling.</p>



Systemet: Opretter en kursist record og kursus record under det ønskede kursus (klassenavn).

Database modellering:

E/R diagram.



Skrevet op som tabeller:

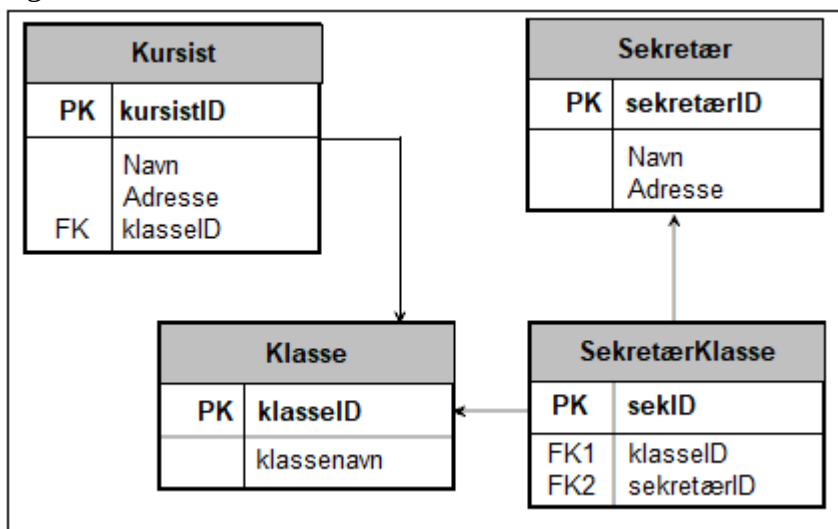
<u>kursistID</u>	Navn	Adresse	Status
int	varchar(40)	varchar(40)	Boolean

<u>KlasseID</u>	Klassenavn
int	varchar(40)

<u>sekretærID</u>	Navn	Adresse
int	varchar(40)	varchar(40)

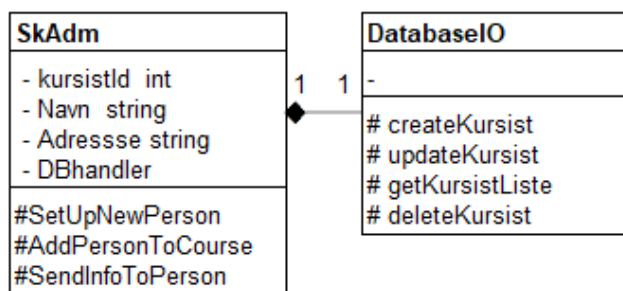
<u>sekID</u>	klasseId	sekretærID
int	FK1	FK2

Database relationsdiagram:



Til den administrative grænseflade skal der laves et klasse diagram.

Med udgangspunkt i de handlinger som sekretæren udfører kan følgende klasser skrives:



En administrativ klasse med handlinger sekretæren udfører (metoder) og en klasse der håndterer databasen (CRUD interface).

På skolens hjemmeside skal den nye tilmeldings formular kunne oploade data til databasen.

Det kan gøres ved at kalde et PHP script der læser det data der er oploade via data formularen og indsætter det i databasen.