

Notetagningshjemmeside

Projektets produkt er en hjemmeside, hvor man kan tage sine noter til skolearbejde, og så vil hjemmesiden selv sortere noten i det fag, noten er skrevet til, registreret med en machine learning algoritme.

NoteSort

Synopsis

Programmering

2020

Programmering

Eksamensprojekt

# Titelblad

## Vejledning

Søren Præstegaard (SPR)

## Fag

Programmering B

## Klasse og skole

3.D - Odense Tekniske Gymnasium

## Antal tegn

## Dato

16. april 2020

# Indholdsfortegnelse

[1 Titelblad 2](#_Toc37929471)

[1.1 Vejledning 2](#_Toc37929472)

[1.2 Fag 2](#_Toc37929473)

[1.3 Klasse og skole 2](#_Toc37929474)

[1.4 Antal tegn 2](#_Toc37929475)

[1.5 Dato 2](#_Toc37929476)

[2 Indholdsfortegnelse 3](#_Toc37929477)

[3 Indledning 5](#_Toc37929478)

[3.1 Opgaveformulering 5](#_Toc37929479)

[4 Kravspecifikation 5](#_Toc37929480)

[4.1 Brugerhistorier 5](#_Toc37929481)

[4.1.1 Log ind 5](#_Toc37929482)

[4.1.2 Opret ny note 6](#_Toc37929483)

[4.1.3 Rediger note 6](#_Toc37929484)

[4.2 Iterationer 7](#_Toc37929485)

[4.2.1 Pre-Iteration - skitser 7](#_Toc37929486)

[4.2.2 Iteration 1 - 9](#_Toc37929487)

[5 Programmets opbygning 9](#_Toc37929488)

[5.1 Tre-lags-modellen 9](#_Toc37929489)

[5.2 Beskrivelse af datalaget 9](#_Toc37929490)

[5.2.1 ER-diagram 9](#_Toc37929491)

[5.3 Beskrivelse af ”machine learning” algoritme 10](#_Toc37929492)

[5.4 Beskrivelse af centrale funktioner 10](#_Toc37929493)

[5.5 Kort om anvendte biblioteker 10](#_Toc37929494)

[6 Forslag til forbedringer 10](#_Toc37929495)

[7 Arbejdsprocessen 10](#_Toc37929496)

[8 Referencer 11](#_Toc37929497)

[9 Bilag 12](#_Toc37929498)

[9.1 Bilag 1 - Brugerhistorier 12](#_Toc37929499)

# Indledning

Det kan være fratage nogen lysten til at skrive noter, fordi det også indebærer en vis form for organisering. Dette er hvorfor jeg i dette projekt har sat mig for at lave en lettilgængelig notetagningshjemmeside, hvor man let kan lave en ny note. Programmet skal efterfølgende undersøge indeholder af notatet og vurdere, hvilket fag noten hører til og selv lægge noten det rigtige sted hen. Dette skulle forhåbentlig give en bedre oplevelse for f.eks. gymnasieelever, når de skal tage noter til fagene.

## Opgaveformulering

Herunder er mere specifikt beskrevet de områder, som jeg gerne vil gennemgå i dette projekt.

* Sammensætningen af en hjemmeside, der funger som en brugerflade til at interagere med en machine learning algoritmen. Herunder en forklaring af opdelingen mellem brugerflade, applikation og datalag (tre-lags-modellen).
* En machine learning algoritme (supervised learning), som givet nogle tekststykker med tilhørende mærkater vil kunne kategorisere nye tekststykker ud fra mønstre i de markerede tekststykker.

# Kravspecifikation

## Brugerhistorier

I dette afsnit er de mest essentielle funktioner af applikationen beskrevet ved brug af brugerhistorier. De resterende brugerhistorier ligger i Bilag 1 - Brugerhistorier.

### Log ind

1. Brugeren skriver sit brugernavn i det øverste inputfelt.
2. Brugeren skriver sit password i det nederste inputfelt.
3. Brugeren trykker på log-ind-klappen.
4. Programmet tjekker om begge felter er fyldt ud.
5. Programmet går ind i databasen og leder efter en bruger med det brugernavn.
6. Programmet tjekker, hvis brugeren er i databasen, og om den brugers kodeord er det samme, som brugeren har indtastet.
   1. Hvis kodeordet er det rigtige føres brugeren videre til brugerens startside.
   2. Hvis kodeordet ikke er rigtigt, siger programmet, at det er forkert, og brugeren kan indtaste igen.

### Opret ny note

1. Når brugeren er logget ind, trykker brugeren på ”New note”-knappen
2. Programmet navigerer til siden for at oprette nye noter.
3. Brugeren skriver et emne i ”Subject”-inputfeltet og indholdet af noten i tekstområdet nedenunder.
4. Brugeren klikker ”Submit”-knappen for at gemme noten.
5. Programmet tager indholdet af tekstområdet, kører det igennem ”machine learning” algoritmen og returnerer algoritmens gæt på det fag, som noten burde tilhøre.
6. Brugeren har nu tre valg
   1. Brugeren kan trykke krydset for at gå tilbage til noten og ændre noget, inden den skal gemmes.
   2. Brugeren kan vælge at trykke ”Yes”-knappen for at acceptere algoritmens placering af noten.
   3. Brugeren kan trykke ”No”-knappen, hvorefter programmet viser en mulighed for brugeren at vælge, hvilket fag noten i stedet skal lægges i.
7. Når noten gemmes, bliver den tilføjet til ”machine learning” algoritmens data, hvorefter algoritmen bliver opdateret (for at gøre den bedre over tid med tilføjelser af flere noter). Herefter bliver noten lagt i databasen.
8. Programmet navigerer tilbage til brugerens startside.

### Rediger note

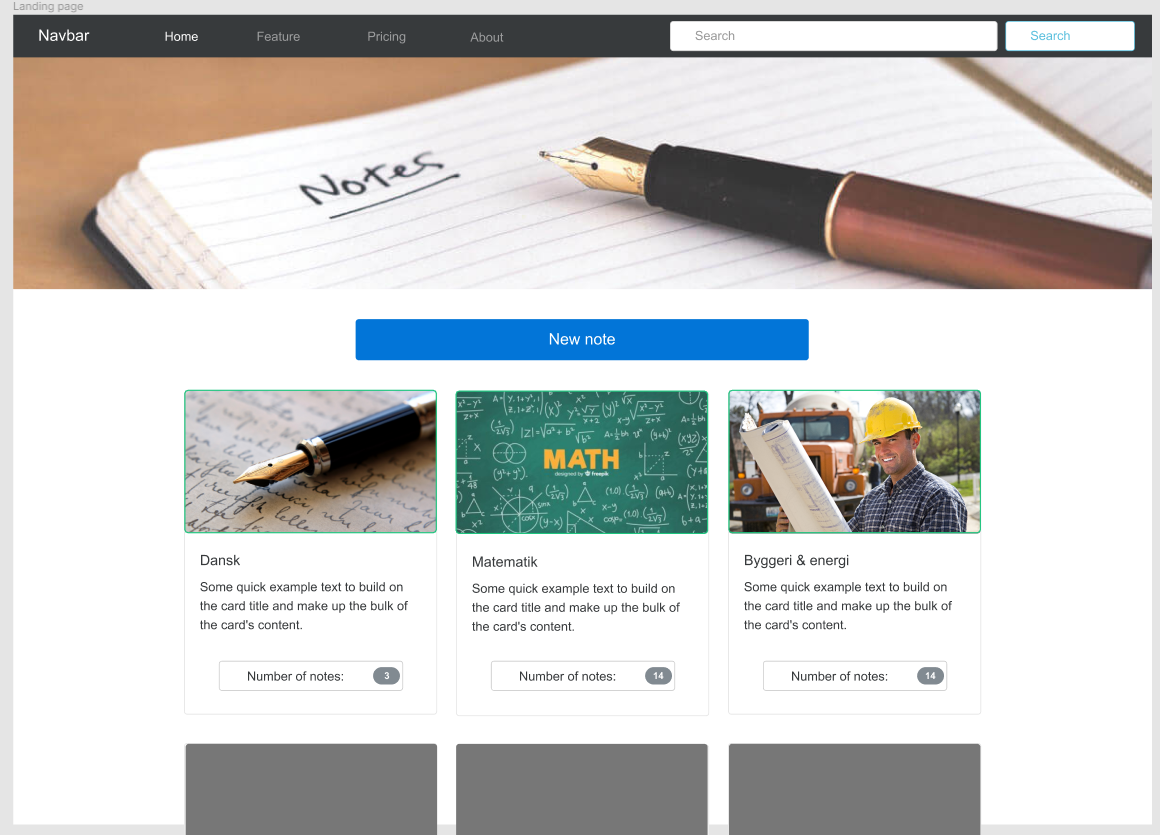
1. Når brugeren er logget ind, klikker brugeren på den klasse, hvor noten, som skal redigeres, ligger i.
2. Programmet navigerer til den klasses side.
3. Brugeren klikker på den note i listen, som skal redigeres.
4. Programmet navigerer til en side, hvor noten alene bliver vist.
5. Brugeren klikker på ”Edit”-knappen.
6. Programmet navigerer til en redigeringsside, hvor værdierne af henholdsvis emnet og tekst allerede er i felterne for dem (opbygget på samme måde som for siden, hvor man skriver en ny note).
7. Brugeren kan tilføje ændringer til noten og trykke ”Submit”-knappen for at gemme ændringerne til noten.
8. Programmet går ind i databasen og tilføjer ændringerne og navigerer tilbage til klasse, hvor den nye note også bliver vist.

## Iterationer

Arbejdet på programmer foregår i iterationer, hvor der er fokus på den enkelte funktions tilføjelse. I dette afsnit har jeg medtaget de mest essentielle iterationer for programmets funktion (her er ikke medtaget ”machine learning” algoritmen, da den bliver beskrevet senere).

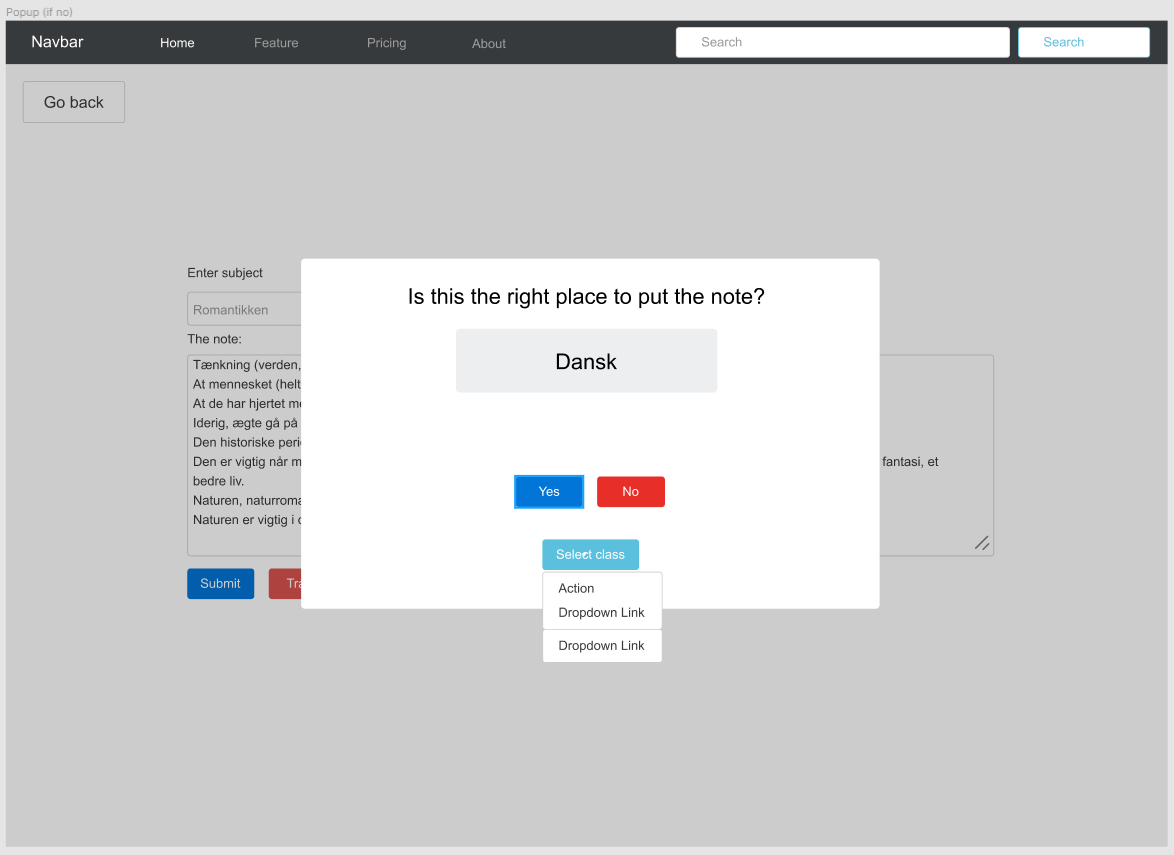
### Pre-Iteration - skitser

Inden jeg begyndte at kode, lavede jeg nogle skitser for at visualisere mit projekt og for at få et overblik over de elementer, som skulle indgå.



Figur 1 Egen illustration af designet af hjemmesidens forside.

På Figur 2 ses det første design af hjemmesidens forside. Her havde jeg tænkt mig at have en nem adgang til at skrive en ny note, men også en nem adgang til at komme til klasserne, som indeholder alle noterne for den pågældende klasse.



Figur 2 Egen illustration af designet af hjemmesidens forside.

\*Skriv videre herfra\*

### Iteration 1 -

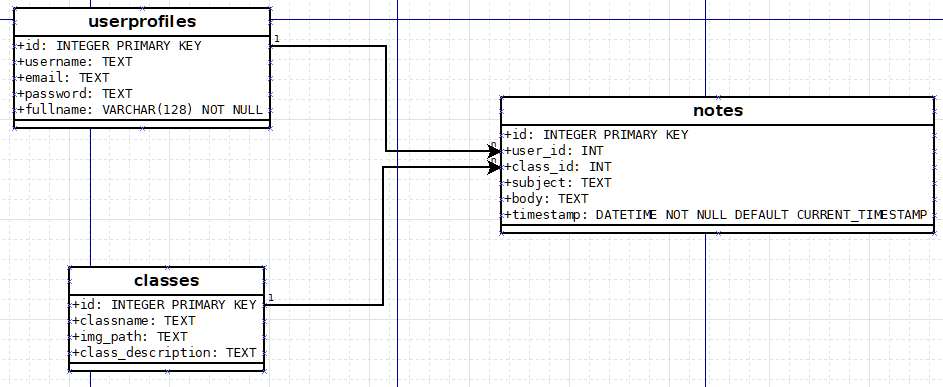
# Programmets opbygning

## Tre-lags-modellen

## Beskrivelse af datalaget

### ER-diagram

For at få et overblik over databasens sammensætning, har jeg lavet et ER-diagram (se Figur 3), som beskriver de tabeller og værdier deri, som jeg bruger i mit program.



Figur 3 Illustration af databasens opbygning (ER-diagram)

## Beskrivelse af ”machine learning” algoritme

## Beskrivelse af centrale funktioner

## Kort om anvendte biblioteker

# Forslag til forbedringer

# Arbejdsprocessen

# Referencer

**There are no sources in the current document.**

# Bilag

## Bilag 1 - Brugerhistorier

* Registrering af bruger
* Log ud
* Slet note