

Gruppe:

Andreas Rømer, Nicklas, Fardin og Jonas

Antal sider total: 11

## The Crosswords - 3 Letter

$\mathcal{K}$	$\mathcal{R}$	$\mathcal{O}$
$\mathcal{L}$	$\mathcal{A}$	$\mathcal{S}$
$\emptyset$	$\mathcal{V}$	$\mathcal{E}$

Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

# Indholdsfortegnelse

<b>The Crosswords - 3 Letter</b>	<b>1</b>
<b>Opgavebeskrivelse</b>	<b>3</b>
<b>Krav</b>	<b>4</b>
<b>Use Cases</b>	<b>5</b>
<b>Use Case Diagram</b>	<b>9</b>
<b>Domænemodel</b>	<b>10</b>
<b>SD Diagram</b>	<b>11</b>
<b>Klassediagram</b>	<b>11</b>

Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

## Opgavebeskrivelse

I vores opgave har vi til hovedformål at oprette en krydsords-generator, der kan tælle alle mulige kombinationer af ord i et krydsord.

Krydsordet skal bestå af ord på tre bogstaver, hvor ingen af disse ord må være ens. Derudover er der andre pre-definerede krav, som vi uddyber i vores liste af krav(Se KR004).

### Vores opgave kan deles op i 5 opgaver:

1.	Første opgave består i at lave en database, tabel og ordliste importeres hertil.
2.	Læse ordlisten ind i en Java ADT (set, map eller list). Lave ordene om så der er en standard(eks. no special chars, alle lowercase.) Skriv de redigerede ord ind i ny database eller tabeller.
3.	Design og implementér et java program der bruger den nye database eller tabeller fra opgave 2 til at lave crosswords med <b>unikke</b> bogstaver.
4.	Gentag opgave 3 men denne gang med klasser, oprettet af os gennem inheritance.
5.	Tag tid og tæl hvor mange crosswords vores program kan lave ud fra vores database.

I slutningen af denne rapport vil være en samlet konklusion, der opsummerer vores forløb, hvad vi fandt frem til, samt muligheder vi har/ikke har fuldført.

Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

## Krav

Krav listen er udarbejdet efter opgaven vi har fået udleveret. Vi har tilføjet lidt til opgaven i form da vi gerne vil kunne styre databasen med CRUD principperne.

ID	Krav	Kommentar
KR001	CRUD på Database	Create, Read, Update & Delete på databasen.
KR002	Udleveret ordliste implementeres	Ordlisten skal filtreres af vores program. Så vi opretholder en standard i vores nye tabel. Se KR004
KR003	Programmet skal generere crosswords i str. 3x3.	
KR004	Retningslinjer for ord i databasen: <ul style="list-style-type: none"><li>• 3 bogstaver</li><li>• lower case</li><li>• a-zæøå</li><li>• ingen specialtegn</li><li>• unikke ord og bogstaver</li><li>• minimum én vokal</li></ul>	Programmet skal sørge for at der ikke tilføjes ord til databasen der ikke overholder retningslinjerne.
KR005	Programmet skal kunne tælle hvor mange crosswords der kan sammensættes ud fra de ord der er i databasen.	
KR006	Programmet skal kunne tage tid på hvor lang tid det tager at tælle kombinationerne.	
KR007		

Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

## Use Cases

<b>Use case 1.</b>	
<b>Title:</b>	<b>Opret Nyt Krydsord</b>
<b>Level:</b>	User goal
<b>Primary Actor:</b>	User
<b>Precondition:</b>	Vi har forbindelse til vores SQL database i vores Java program
<b>Stakeholders and interests:</b>	User skal kunne oprette et nyt krydsord i vores program.
<b>Main success scenario:</b>	<ol style="list-style-type: none"> <li>1. User vælger "Opret nyt krydsord" fra menuen i programmet</li> <li>2. Systemet opretter et nyt krydsord der opfylder vores givne krav. (Se KR004)</li> <li>3. Systemet bekræfter, at et nyt krydsord er oprettet og returnerer til menuen.</li> </ol>
<b>Extensions:</b>	<p><b>1a.</b></p> <ul style="list-style-type: none"> <li>- User kan til hver en tid vælge at gå tilbage/stoppe programmet.</li> </ul> <p><b>2a.</b></p> <ul style="list-style-type: none"> <li>- Programmet kan ikke oprette et krydsord der opfylder kravene, men vil da præsentere en fejlkode som beskriver årsagen.</li> </ul> <p><b>3a.</b></p> <ul style="list-style-type: none"> <li>- Programmet kan ikke bekræfte at et nyt krydsord er oprettet, præsenterer en beskrivende fejlkode og returnerer til menuen.</li> </ul>

Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

<b>Use case 2</b>	
<b>Title:</b>	<b>Slet Ord</b>
<b>Primary Actor:</b>	User
<b>Precondition:</b>	Vi har forbindelse til vores SQL database i vores Java program.
<b>Stakeholders</b>	Vi som udviklere vil gerne have mulighed for at slette ord fra databasen, det kunne f.eks. være ord som forekommer flere gange ved et tilfælde, eller vi skal fjerne ord som ikke skal bruges i vores kryds og tværs længere. Udover det, så er det en feature vi skal have med for at overholde CRUD.
<b>Main success scenario:</b>	<ol style="list-style-type: none"> <li>1. User vælger "Slet ord" fra programmenuen.</li> <li>2. User specificerer hvilket ord, der ønskes slettet.</li> <li>3. Programmet søger i vores database, finder ordet og sletter dette.</li> <li>4. Programmet bekræfter specificerede ord er slettet fra databasen og returnerer til menuen.</li> </ol>
<b>Extensions:</b>	<b>3a.</b> <ul style="list-style-type: none"> <li>- Programmet kan ikke finde det specificerede ord i vores database. Programmet giver besked om at ordet ikke er fundet, og beder om at specificere et nyt ord.</li> </ul>

<b>Use case 3</b>	
<b>Title:</b>	<b>Tilføj Ord</b>
<b>Primary Actor:</b>	User
<b>Precondition:</b>	Vi har forbindelse til vores SQL database i vores Java program.
<b>Stakeholders</b>	User er interesseret i at kunne tilføje ord til databasen, således at flere kombinationer af krydsord vil være tilgængelige.
<b>Main success scenario:</b>	<ol style="list-style-type: none"> <li>1. User vælger "Tilføj ord" fra programmenuen.</li> <li>2. User indtaster et ord, der skal tilføjes til databasen.</li> <li>3. Systemet validerer indtastede ord i forhold til krav(KR004) og tilføjer til databasen.</li> </ol>

Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

	<ol style="list-style-type: none"> <li>4. Systemet giver en besked om, at det indtastede ord er tilføjet korrekt til databasen.</li> <li>5. Systemet returnerer til menuen.</li> </ol>
<b>Extensions:</b>	<b>3a.</b> <ul style="list-style-type: none"> <li>- Det indtastede ord opfylder ikke kravene (KR004). Systemet giver en fejlmeddelelse der beskriver, at ordet ikke kan tilføjes da kravene ikke er opfyldt og returnerer til menuen.</li> </ul>

<b>Use case 4</b>	
<b>Title:</b>	<b>Redigér Ord</b>
<b>Primary Actor:</b>	User
<b>Precondition:</b>	Vi har forbindelse til vores SQL database i vores Java program.
<b>Stakeholders</b>	User er interesseret i at kunne redigere ord f.eks. hvis de er stavet forkert, ikke er gyldige ord og for vedligeholdelse.
<b>Main success scenario:</b>	<ol style="list-style-type: none"> <li>1. User vælger "Redigér ord" i programmenuen.</li> <li>2. User specificerer et ord, der ønskes redigeret.</li> <li>3. Programmet søger i databasen efter indtastede ord.</li> <li>4. User kan nu indtaste ordet igen.</li> <li>5. Systemet validerer indtastede ord i forhold til krav(KR004) og erstatter forhenværende ord med nye input fra User</li> <li>6. Systemet giver en bekræftende besked om, at ordet nu er redigeret/opdateret og returnerer til menuen.</li> </ol>
<b>Extensions:</b>	<b>3a.</b> <ul style="list-style-type: none"> <li>- Systemet kan ikke finde indtastede ord. Der gives meddelelse om at indtastede ord ikke findes i databasen og returneret til menuen.</li> </ul> <b>5a.</b> <ul style="list-style-type: none"> <li>- Indtastede ord fra User opfylder ikke kravene, fejlmeddelelse omkring uopfyldte krav specificeres og der returneres til menuen. Ordet erstattes ikke.</li> </ul>

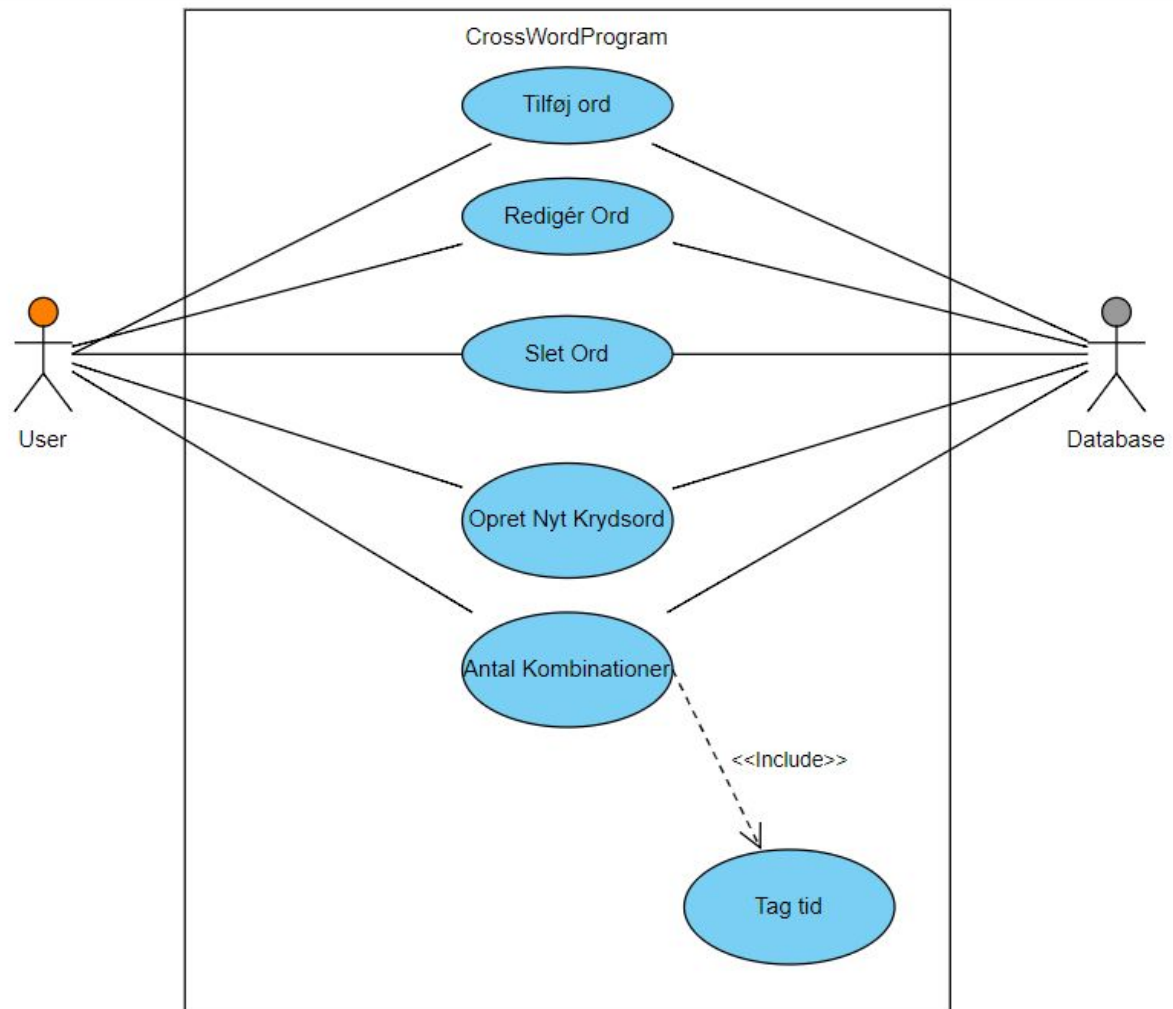
Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

<b>Use case 5</b>	
<b>Title:</b>	<b>Antal Kombinationer</b>
<b>Primary Actor:</b>	User
<b>Precondition:</b>	Vi har en funktionel database og kan filtrere vores ordliste.
<b>Stakeholders</b>	User skal kunne køre denne funktion i vores program, da user ønsker at vide hvor mange kombinationsmuligheder der er, med nuværende ordliste(r) i databasen.
<b>Main success scenario:</b>	<ol style="list-style-type: none"><li>1. User vælger "Antal Kombinationer" i programmenuen.</li><li>2. Programmet tæller alle mulige kombinationer der opfylder kravene(KR004) baseret på ordlisten i vores database. Dernæst returneres et tal der repræsenterer alle mulige kombinationer og tiden det tog.</li><li>3. Programmet returnerer til menuen.</li></ol>
<b>Extensions:</b>	<b>2a.</b> <ul style="list-style-type: none"><li>- Det er ikke muligt for programmet at tælle kombinationerne. En specificerende fejlmeddelelse præsenteret og der returneres til menuen.</li></ul>



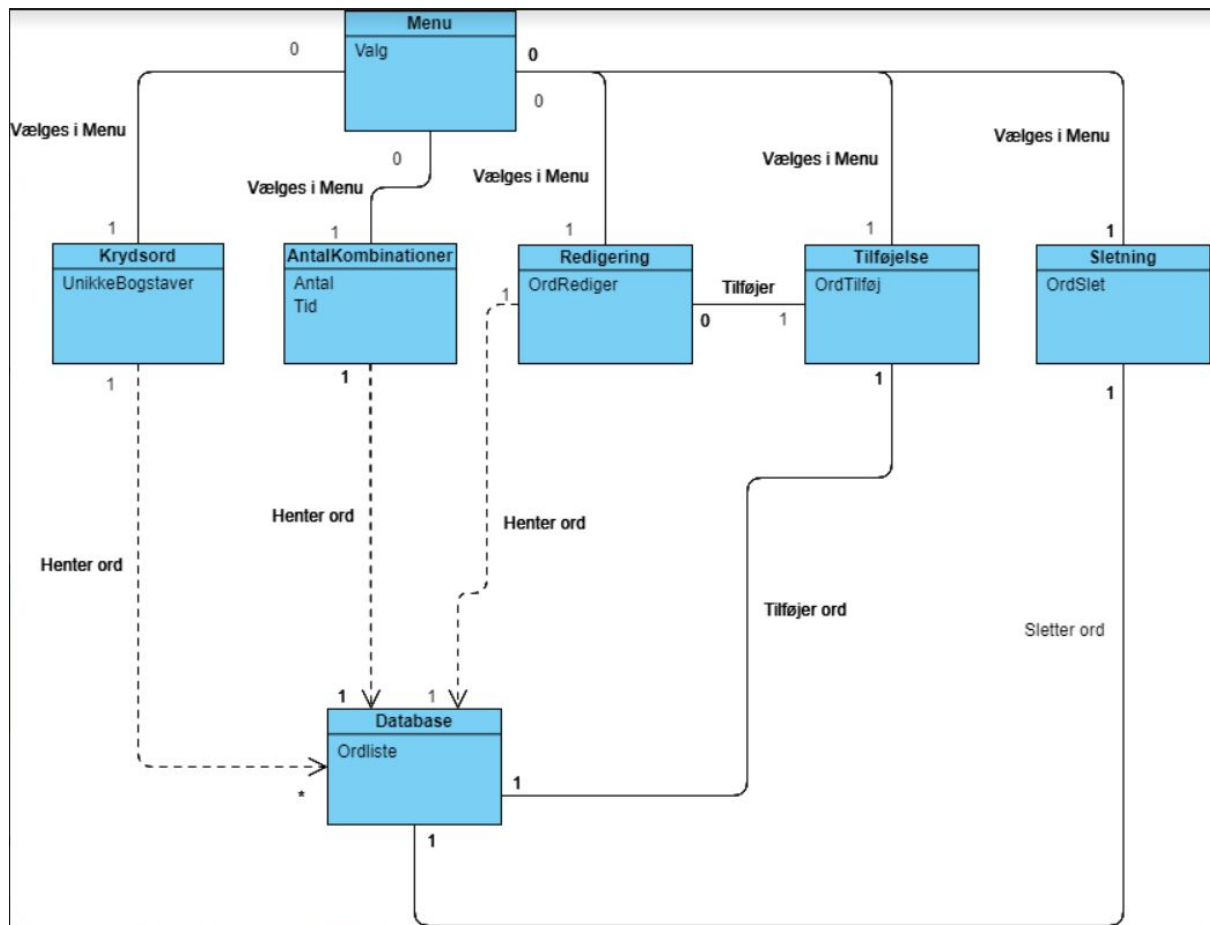
Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

## Use Case Diagram



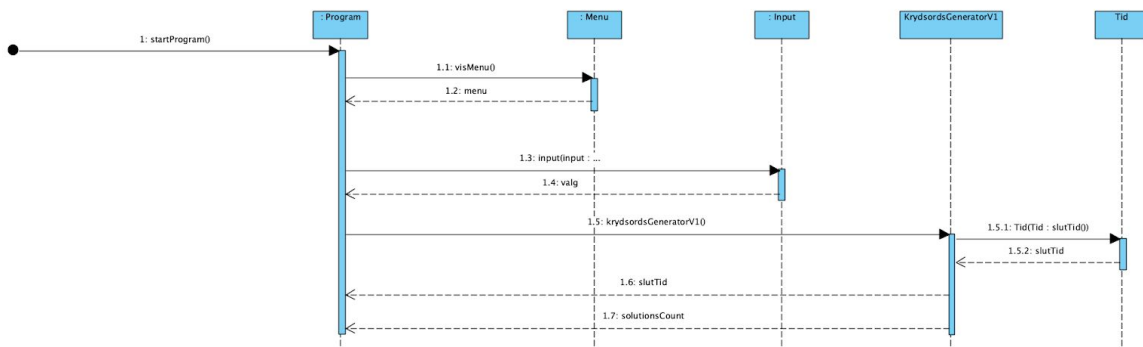
Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

# Domænenemodel

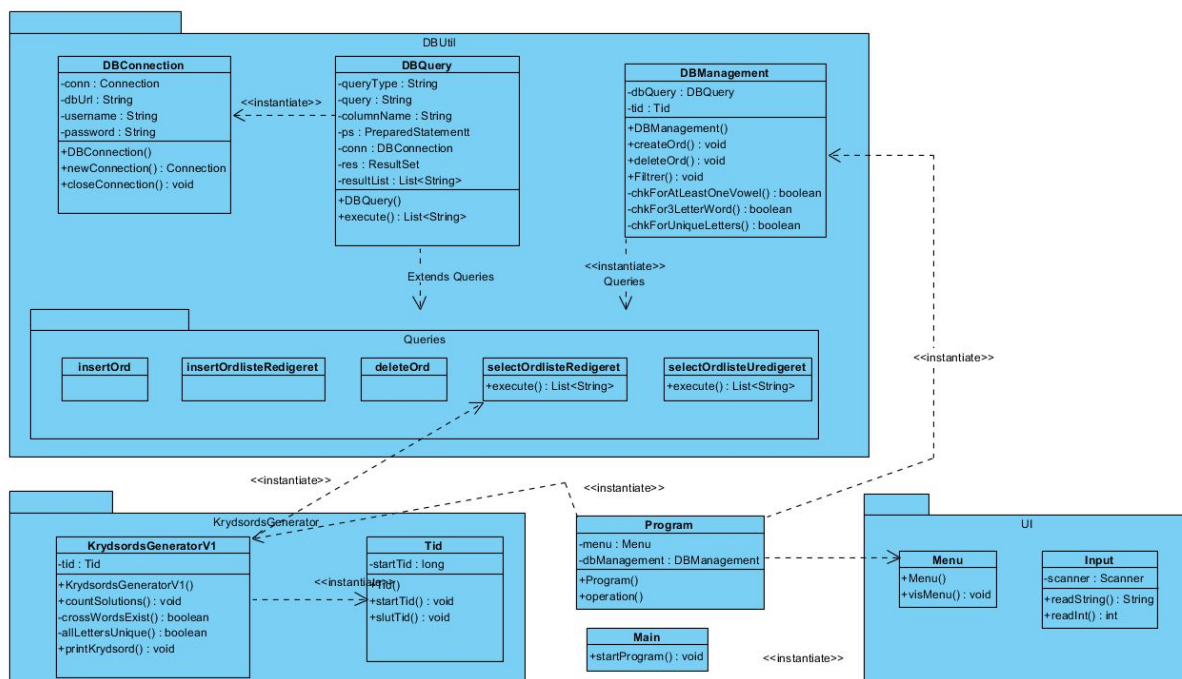


Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

## SD Diagram



## Klassediagram



Gruppe: Andreas Rømer, Nicklas, Fardin og Jonas

**Beskrivelse af vores algoritme til at finde crosswords med:**

1. Vi sætter 3 ord vandret ind i vores crossword fra vores ordliste.
2. Vi tager eksempelvis tre ord; **“dog, god, køn”** de sættes ind vandret:
3. Først tjekker vi om bogstaver er unikke. Hvis de ikke er unikke går vi tilbage til step 1.
4. Hvis alle bogstaver er unikke; Så tjekkes de 3 lodrette ord der nu er opstået; **“dgk, ooø, gdn”** om ordene findes i vores ordliste. Hvis de findes har vi en crossword, hvis ikke går vi til step 1.

d	o	g
g	o	d
k	ø	n