



# Eksamen

## Generell info

Gruppeeksamen

Veiledning

Discord

Plagiat

GitHub

Eksamensmappen

Prosjektdokument

Karakter

Innlevering

## Oppgave 1

Deloppgave 1

Deloppgave 2

Deloppgave 3

Deloppgave 4

## Oppgave 2

Deloppgave 1

Deloppgave 2

Deloppgave 3

Deloppgave 4

## Oppgave 3

## Oppgave 4

Stepper

ColorPicker

useCounter

## Generell info

### Gruppeeksamen

- Gruppene består av 3-4 personer og må opprettes selv i Canvas
- Innlevering skjer i Inspera innen eksamensfristen
- Gruppen må levere refleksjonsnotat

## Veiledning

Det vil som nevnt være mulig å få veiledning underveis i eksamensperioden. Link og instruksjoner knyttet til veiledningen finner du på denne [linken](#). Veiledning vil foregå på Discord.

## Discord

Det har blitt opprettet en kanal i Discord (#prosjekt-spørsmål) for eksamen. Her kommer det informasjon som er relevant for alle.

- I denne kanalen kan dere også få hjelp til eksamensrelevante spørsmål. Forutsetningen for å få hjelp er:
  - At problemet forklares til oss på Discord
  - At siste versjon av koden er tilgjengelig på Github slik at vi har tilgang til koden
  - At vi får screenshot av feilmeldingen
  - At du har prøvd å løse feilen

## Plagiat

Kode du låner / kopierer fra diverse ressurser, må kommenteres. Dette for å unngå at hele gruppen stryker grunnet plagiat.

## GitHub

Her er et dokument som kan brukes hvis gruppen har glemt hvordan bruke Git / er usikker på hvordan bruke Git som team. Link til dokumentet finner du [her](#).

## Eksamensmappen

I [eksamensmappen](#) ligger

- Zip av koden (uten min lokale .git)
- Video om eksamen
- Eksamensoppgavene

- Mal prosjektdokument
- Postmancollection
- Dokumenter som viser hvordan jobbe i team på GitHub

## Prosjektdokument

Dette dokumentet skal fylles ut av gruppen og leveres sammen med eksamen. Link til dokumentet finner du [her](#).

## Karakter

Oppgavene står i prioritert rekkefølge. Sensor setter karakter etter hvilken grad akseptansekriteriene er oppfylt. Helhetsvurderingen legges til grunn hvis det er få mangler på et akseptansekriterie.

### Absolutte krav

- Applikasjonen må være kjørbart (at det fungerer å starte prosjektet)
- Masterbranchen inneholder alle deltakerne sin kode (at alt er merget med master eller main)
- At det er få eller ingen "røde" errors i Google Chrome consolen
- At alt som skal leveres inn er levert inn til fristen
- At sensor har tilgang til GitHub-prosjektet

### Eksempel på akseptansekriterie

| Det skal være mulig å legge til en kommentar til en henvendelse

Dette kriteriet har flere krav som må oppfylles

- Du kan ha en `frontend` med `dummydata`, men ingen `API`. Kriteriet er da delvis oppfylt.

- Du kan ha en `backend` som kan testes med `Postman`, men ingen `frontend`. Kriteriet er da delvis oppfylt.
- Du kan ha en `frontend` og `API`, men de kommuniserer ikke sammen. Kriteriet er da delvis oppfylt.
- Du kan ha en `frontend` og `API`, men oppfyller ikke krav til `design`, `validering` og `errorhåndtering`. Kriteriet er da delvis oppfylt.

## Innlevering

Når dere skal levere gruppeeksamen må dere

- Lage en gruppe i Inspira
  - Inviter alle deltakerne til denne gruppa
- Last opp en .zip fil med
  - Grupperapporten / refleksjonsnotat som baserer seg på malen dere har fått. Ved behov kan også individuelle rapporter leveres.
    - Denne rapporten skal inneholde link til GitHub-repoet deres
  - (Hvis relevant) tekstdokumenter som inneholder:
    - Antagelser dere har gjort i oppgaven som sensor bør kjenne til
    - Linker til kode dere har kopiert fra eksterne kilder (stack overflow, google, ++ ) (for å unngå plagiat) og hvor i koden dere har brukt det
    - Informasjon sensor må kjenne til for å starte prosjektet (.env variabler e.l.)
    - Annet du tenker sensor bør kjenne til (f.eks Wireframes fra IA e.l.)
  - Hele prosjektet **UTEN** node\_modules (dette er kun en nødløsning)
- Inviter [hiofkonto@gmail.com](mailto:hiofkonto@gmail.com) til deres **GitHub** repo med **alle rettigheter**

## Oppgave 1

I denne oppgaven har jeg laget et frontend for å lage supporthenvendelser.

Applikasjonen er ikke ferdig, og det er flere features kunden trenger. Utviklingen av

applikasjonen er delt opp i 4 deloppgaver. Koden til oppgaven ligger i mappen `helpdesk`.

### Dette er akseptansekriteriene kunden har satt:

- Det skal være mulig å navigere til en side for å lage supporthenvelser. Kunden har ingen krav til styling av skjema som brukes på denne siden.
  - Informasjonen som sendes skal valideres på frontend og backend.
    - Tittel er required, minimum 25 bokstaver langt og maks 150
    - Beskrivelse er required og maks 250 bokstaver langt
    - Navn er required og skal bestå av `Fornavn Etternavn`. Kravet er mellomrom og stor forbokstav i fornavn og etternavn.
    - Viktighet og avdeling er required
- Det skal være mulig å navigere til en side for å se alle supporthenvelser som har blitt laget.
  - Denne siden skal ivareta designet gitt av kunden
- Det skal være mulig å navigere til en side for å se en enkelt supporthenvelse og alle kommentarer som hører til den. Kunden har ingen absolutte krav til design av denne siden, men forventer at designet fra `alle henvendelser`-siden gjenbrukes.
- Det skal være mulig å filtrere listen med alle supporthenvelser. Filtrene er `viktighet` eller `avdeling`.
- Det skal være mulig å kommentere en supporthenvelse.
  - Tekst til kommentar er required og maks 250 bokstaver langt
  - Det skal være mulig å legge til kommentarer til en henvendelse på siden for alle henvendelser (listervisningen)
  - Det skal være mulig å legge til kommentarer på siden for den enkelte supporthenvelsen
  - Antall kommentarer skal vises i henhold til design
- Det skal være mulig å avslutte / lukke en supporthenvelse. Den skal da merkes som `løst` i grensesnittet. F.eks `Tittel på henvendelsen (løst)`.

- Det skal være mulig å gjøre både på listevisningen og den enkelte supporthenvendelsen.
- Ved eventuelle feil skal brukeren få riktig feilkode og en beskrivende feilmelding

## **Deloppgave 1**

I første runde ønsker kunden å oppgradere designet slik at det blir enklere å se alle henvendelsene. Din oppgave er å implementere et design som ligner designet du ser nedenfor.

Gjelder **kun** designet for å se alle henvendelser.

# Henvendelser

Viktighet ▼

Avdeling ▼

Salg

Høy ●

## Tittel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

Trude Trudesen

05.12.12

[Legg til kommentar](#) [Avslutt](#)

It

Medium ●

## Tittel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

Trude Trudesen

05.12.12

[Legg til kommentar](#) [Avslutt](#)

Design

Lav ●

## Tittel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

Trude Trudesen

05.12.12

[Legg til kommentar](#) [Avslutt](#)

# Henverdeler

Viktighet ▼

Avdeling ▼

Salg

Høy ●

## Tittel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

Trude Trudesen

05.12.12

[Legg til kommentar](#) [Avslutt](#)

## Legg til kommentar

X

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

[Send](#)

It

Medium ●

## Tittel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

Trude Trudesen

05.12.12

[Legg til kommentar](#) [Avslutt](#)

Design

Lav ●

## Tittel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

Trude Trudesen

05.12.12

[Legg til kommentar](#) [Avslutt](#)



# Henvelndelser

Viktighet ▼

Avdeling ▼

Salg

Høy ●

## Tittel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

Trude Trudesen

05.12.12

[Se kommentarer \(2\)](#)

[Legg til kommentar](#)

[Avslutt](#)

## Kommentar 1

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

## Kommentar 2

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

It

Medium ●

## Tittel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

Trude Trudesen

05.12.12

[Legg til kommentar](#)

[Avslutt](#)

Design

Lav ●

## Tittel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero

Trude Trudesen

05.12.12

[Legg til kommentar](#)

[Avslutt](#)

## Deloppgave 2

I prototypen du fikk, mangler en del funksjonalitet. I denne oppgaven skal du ferdigstille `frontend` slik at dette oppfylles:

- I skjema for å lage en supporthenvendelse mangler
  - Mulighet til å velge grad av viktighet (høy, medium eller lav)
  - Mulighet til å velge mellom avdelingene (it, design eller salg)
  - Validering av skjema
- Navigasjon
  - Egen side for å lage supportsaker (f.eks `/issues/create`)
  - Egen side for å se alle saker (f.eks `/issues`)
  - Egen side for å se en sak og alle kommentarer tilhørende saken (f.eks `/issues/123`)
- Filter
  - Mulighet til å se saker med en gitt viktighet (høy, medium og lav)  
eller
  - Mulighet til å se saker for en avdeling (it, design og salg)
- Kommentarer
  - Mulighet til å kommentere en sak som har blitt opprettet
- Avslutte / lukke saker
  - Mulighet til å avslutte / lukke en sak

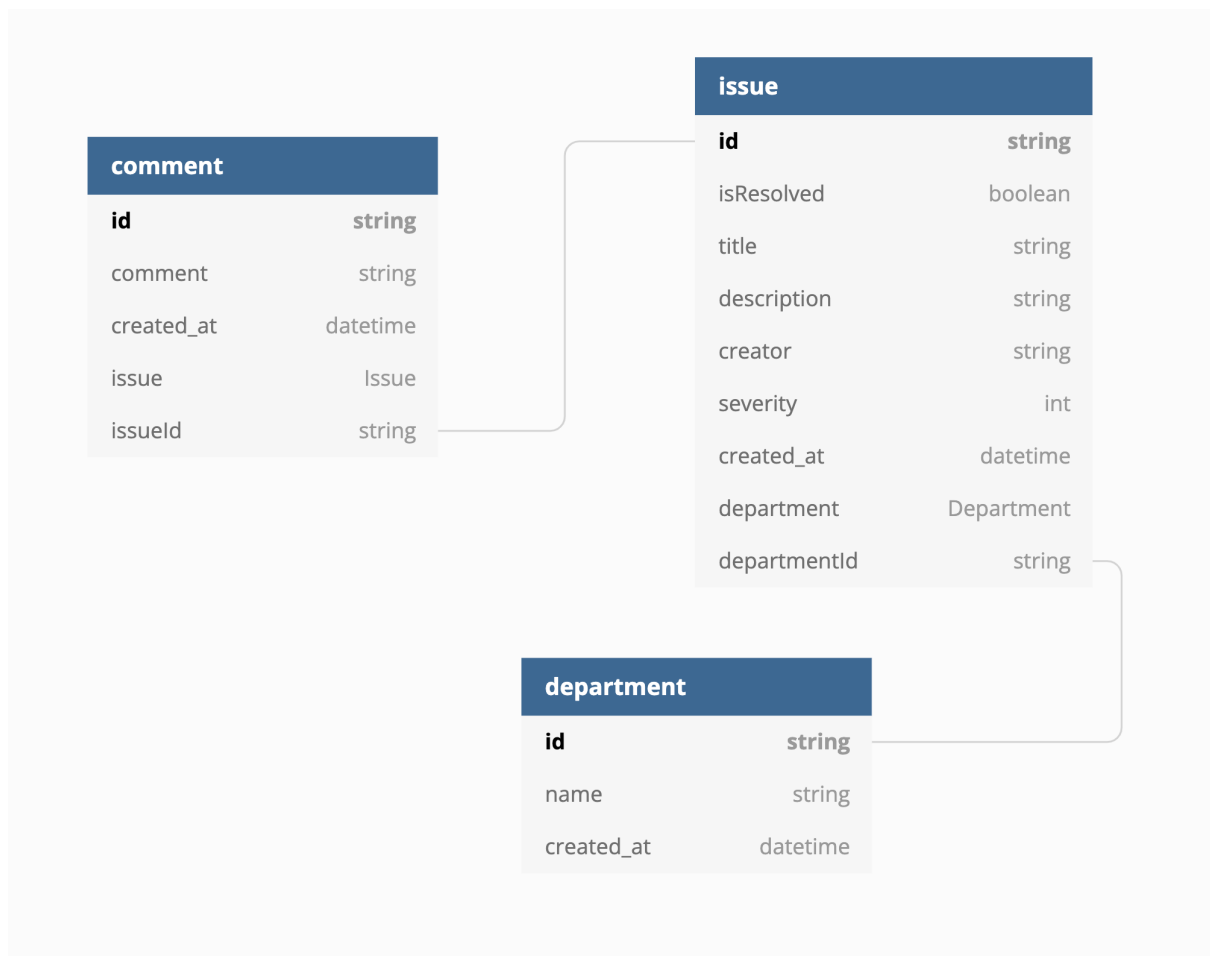
## Deloppgave 3

Kunden er veldig fornøyd med den nye funksjonaliteten og designet, men savner en mulighet til å lagre informasjonen. Kunden trenger derfor et `API` som kan brukes til å oppfylle de kravene kunden har til `CRUD`, validering og errorhåndtering.

I denne oppgaven er det kun fokus på API. API kan testes via f.eks `Postman`. I neste oppgave kobles alt sammen.

Dette diagrammet har kunden utarbeidet

- En `Issue` kan ha mange `Comments`
- En `Comment` kan kun høre til en `Issue`
- En `Issue` kan kun høre til en `Department`
- En `Department` kan ha mange `Issues`



## Deloppgave 4

Koble sammen `API` og `frontend` slik at akseptansekriteriene oppfylles.

## Oppgave 2

Det nærmer seg jul og kunden har veldig lyst til å få på plass en julekalender. Utviklingen av applikasjonen er delt opp i 4 deloppgaver. Koden til denne oppgaven ligger i mappen `christmas`.

### Dette er akseptansekriteriene kunden har satt:

- Startsidene skal vise en liste med X antall luker. Denne siden skal ivareta designet gitt av kunden.
- Luker som du allerede har åpnet skal markeres med fargekoden `#96E8C5` (grønnaktig farge). Slik identifiserer du at luken har blitt åpnet av deg.
- Luker som ikke er åpnet, men som er tilgjengelig skal være hvite
- Luker som ikke er tilgjengelig skal være dus grå og markert med `Åpner om X dager` fra dagens dato
- Det skal være mulig å åpne en luke i henhold til reglene nevnt under
  - Ikke åpne før luken er tilgjengelig
  - Kun åpne luken (få kode) en gang
  - Koden som skjuler seg bak luken skal være en random streng bestående av 4 tall og bokstaver
- Det skal være mulig å navigere til siden `dashboard` av `admin`.
  - På denne siden skal admin se en liste over alle deltakere som har åpnet en gitt luke, og deres kode. Denne siden skal ivareta designet gitt av kunden.
  - By default skal det kun vises 3 deltakere
  - Hvis noen som ikke er `admin` besøker siden skal det vises en melding. Meldingen kan f.eks være `Du har ikke tilgang til å se dette innholdet`.
  - På denne siden skal det til hver luke kunne trekkes en tilfeldig superbons-vinner. Det er kun `admin` som kan bruke denne funksjonaliteten fra APllet.
    - Trenger ikke å lagre resultatet fra trekking da kunden i fremtiden vil maile den som vinner.
- Det skal i applikasjonen gis hensiktsmessige feilmeldinger og statuskoder.

### Hjelpesfunksjoner du kan benytte deg av:

- `seed.js` har funksjon for seeding av user, slot og kalender. Databasen må ha `users` for at applikasjonen skal fungere.
- `/lib/utills/user` har en metode som heter `userInfo` som kan brukes til å hente ut informasjon om brukeren. Se `/api/dummy.js` for eksempel.

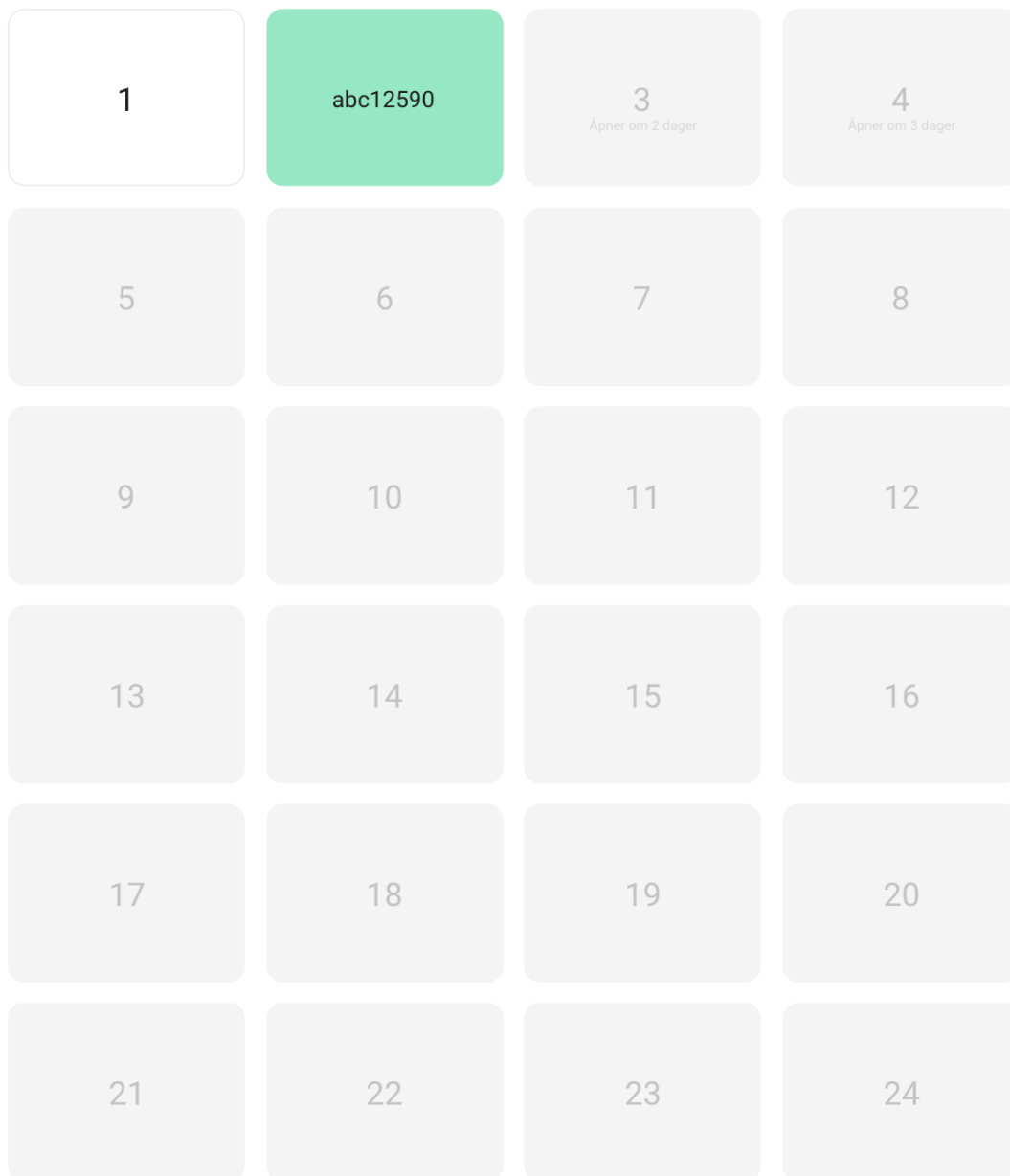
- `postman-collection` du kan bruke til å kommunisere med `API`.
  - OBS: `userInfo` hjelpefunksjonen fungerer ikke via `Postman`
  - `collection` ligger i mappen

## Deloppgave 1

Kunden hadde sett for seg et grensesnitt av denne typen for å vise lukene. Din oppgave er å implementere dette. Kunden har noen ønsker (ikke krav) til animasjoner og ser for seg at:

- Det å trykke på en uåpnet luke trigger en animasjon
- Det å trykke på en luke som ikke er tilgjengelig, trigger en annen type animasjon
- En luke som allerede er åpnet, skal ikke trigge noen animasjon når den trykkes på

## Julekalender eksamen 2021



Kunden hadde sett for seg et grensesnitt av denne typen for å vise `dashboard`-siden. Din oppgave er å implementere dette.

# Admin

## Luke 1

Tilgjengelig fra: 05.12.12

**Se alle deltakelser (44)**   **Trekk superbonus**

#	Brukernavn	Dato for deltakelse	Kode
1	Trude	05.12.21	ac12bc90
2	Simen	05.12.21	89kr9012
3	Åse	06.12.21	ok77cc99

## Luke 2

Tilgjengelig fra: 06.12.12

**Se alle deltakelser (44)**   **Trekk superbonus**

#	Brukernavn	Dato for deltakelse	Kode
1	Trude	05.12.21	aq12bc90
2	Simen	05.12.21	19kr9012
3	Åse	06.12.21	ro77cc99

# Admin

Luke 1

Tilgjengelig fra: 05.12.12

[Se alle deltakelser \(44\)](#) [Trekk superbonus](#)

#	Brukernavn	Dato for deltakelse	Kode
1	Trude	05.12.21	aq12bc90

## Superbonus, luke 1

X

Trude

#	Brukernavn	Dato for deltakelse	Kode
1	Trude	05.12.21	aq12bc90
2	Simen	05.12.21	19kr9012
3	Åse	06.12.21	ro77cc99

## Deloppgave 2

APIet er litt rufsete satt sammen og alle relasjoner er ikke satt opp.

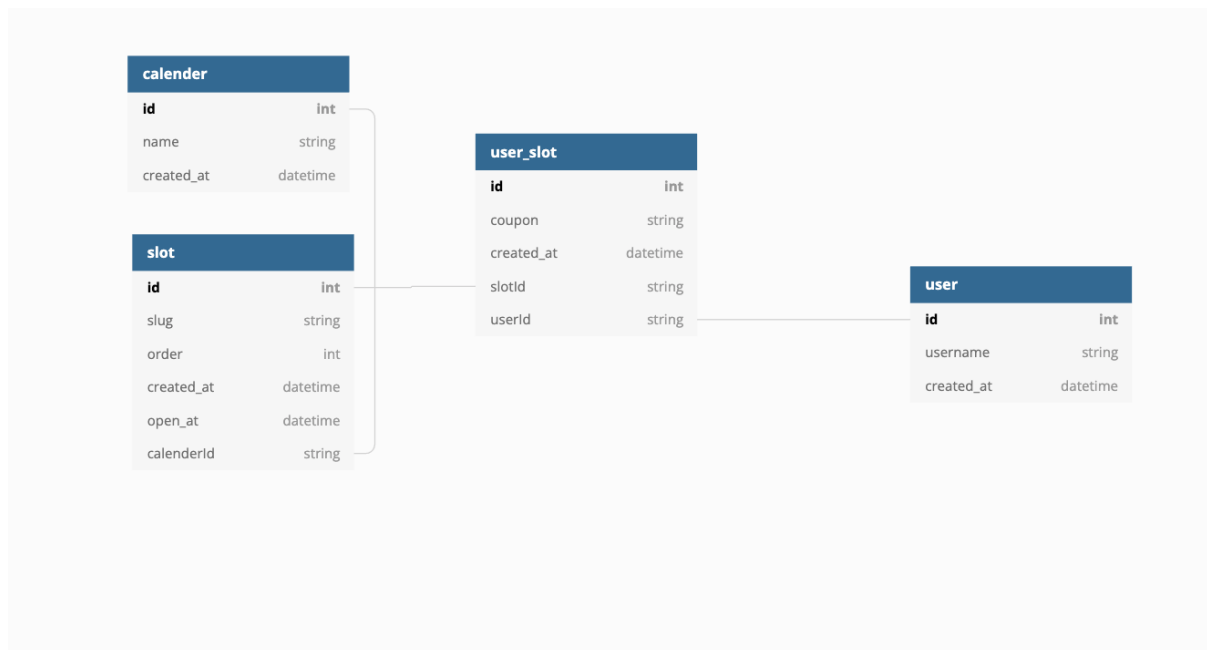
### Din oppgave er å:

- Gjøre nødvendig refaktorering på eksisterende API slik at det blir mer robust.
  - Ta utgangspunkt i strukturen gått igjennom i timen / generalisering av kode
  - Ha med feilhåndtering og feilkoder
- Sette opp nødvendige relasjoner slik at at relasjonene i tabellen under ivaretas.



Her er tabellen brukt til å lage julekalenderen

- En **Calender** kan ha mange **Slots**, men en **Slot** kan kun høre til en **Calender**
- En **Slot** kan ha mange **UserSlots**, men en **UserSlot** kan kun høre til en **Slot**
- En **User** kan ha mange **UserSlots**, men en **UserSlot** kan kun høre til en **User**
- **UserSlot** er den tabellen som brukes for å ha kontroll på en **User** sin deltakelse



### Deloppgave 3

API er ikke ferdig og mangler blant annet all **businesslogikk**. Din oppgave er å implementere logikken som ivaretar punktene nedenfor. Du bør gjenbruke strukturen fra tidligere for å sikre god kvalitet på koden.

I denne oppgaven er det kun fokus på API. API kan testes via f.eks **Postman**. I neste oppgave kobles alt sammen.

- Skal kun være mulig å åpne luker som er tilgjengelig
- Skal kun få lov til å delta en gang
- Listervisningen skal tydelig vise hvilke luker en bruker har åpnet fra før
- **coupon** skal være en random string bestående av 4 bokstaver og 4 tall. Dette er en **unik** kode som skjuler seg bak dagens luke. Hvis en annen bruker åpner

samme luke, skal de få en annen kode.

- Skal som `admin` ha mulighet til å trekke en deltaker fra en gitt `luke` som får en ekstra julebonus

Du skal i tillegg teste en `controller` med `service` og `repository` fra kode du har laget.

## Deloppgave 4

Koble sammen `API` og `frontend` slik at akseptansekriteriene oppfylles.

## Oppgave 3

I denne oppgaven skal du bygge videre på `master mind` utdelt som kode. Koden inneholder en 5 `TODOS` som du må løse. Det meste rundt frontenden er klart, men det er ingen kobling mot API, samt at noe logikk må flyttes til API. Koden til denne oppgaven ligger i mappen `master_mind`.

- Et spill vil si X antall gjennomføringer
- En gjennomføring vil si et forsøk på å gjette rett kombinasjon

**Dette er akseptansekriteriene kunden har satt:**

- Det skal automatisk lages en randomisert kombinasjon av 4 farger når et nytt spill starter. En farge skal eksistere kun en gang.
  - `TODOS` ligger i `/contexts/game-context.js => linje 135`
- Spilleren sitt brukernavn (som blir automagisk generert når applikasjonen starter) skal lages når spillet lagres. Se utdelt funksjon i `dummy.js` for å hente ut informasjon om spilleren.
- For hver gjennomføring skal det via API gis et hint tilbake. Hintet baserer seg på reglene i spillet
  - `TODOS` ligger i `/components/game/Rows.js => linje 20 / 37`
- For hvert spill skal det lagres antall forsøk (kan også gjøres for hver gjennomføring)

- `TODO` ligger i `/components/game/Rows.js => linje 45 / 49`
- For hvert spill skal det lagres om oppgaven ble løst eller ikke
  - `TODO` ligger i `/components/game/Rows.js => linje 45 / 49`
- Det skal gis hensiktsmessige feilmeldinger og statuskoder

Denne tabellen kan være til hjelp

```
Table Game {
  id Int [pk, auto_increment]
  combination String
  user String [not null]
  numberOfTries Int [not null] default(0)
  foundCombination Boolean default(false)
}
```

## Oppgave 4

Denne oppgaven har noen komponenter og tilhørende tester som ikke er ferdig. Din oppgave er å fullføre testene for hver av komponentene. Koden til denne oppgaven ligger i mappen `testing`.

### Stepper

I denne oppgaven skal du teste funksjonaliteten til komponenten nedenfor.

```
import { useState } from 'react'

const steps = [{ name: 'Hjem' }, { name: 'Game' }, { name: 'End' }]

const Stepper = () => {
  const [step, setStep] = useState(0)

  return (
    <>
      {step + 1 < steps.length ? (
        <button type="button" onClick={() => setStep(step + 1)}>
          {steps[step + 1].name}
        </button>
      ) : null}
    </>
  )
}
```

Her er test-casene du skal ferdigstille

```
describe('Stepper component', () => {

  it('should render button', () => {
  })

  it('should have correct text content on button', () => {
  })

  it('should update step-count and button content on click', async () => {
  })

  it('should remove button when step count is higher than amount of steps', async () => {
  })
})
```

## ColorPicker

I denne oppgaven skal du teste funksjonaliteten til komponenten nedenfor

```
const ColorPicker = ({ colors, selectedColor, handleSelectedColor }) => {
  return (
    <ul className="colorPicker">
      {colors.map((color) => (
        <li key={color} data-testid="color">
          <button
            style={{
              backgroundColor: color,
              opacity: selectedColor ? (selectedColor === color ? 1 : 0.2) : 1,
            }}
            disabled={selectedColor && selectedColor !== color}
            onClick={() => handleSelectedColor(color)}
          ></button>
        </li>
      ))}
    </ul>
  )
}

export default ColorPicker
```

Her er test-casene du skal ferdigstille

```
describe('ColorPicker', () => {
  it('should render a list of all colors passed to it', () => {
  })

  it('should have disabled button if color does not match', () => {
  })

  it('should have one active button if color match', () => {
  })

  it('should have called onClick on button', async () => {
  })

  it('should not have called onClick on disabled button', async () => {
  })

  it('should update selectedColor and active buttons on click', async () => {
  })
})
```

## useCounter

Denne testen feiler når vi prøver å teste vårt `custom` hook. Gjør nødvendige endringer i koden nedenfor slik at testen virker

Her er `useCounter` hooket

```
import { useState } from 'react'

export function useCounter() {
  const [counter, setCounter] = useState(1)

  const add = () => {
    setCounter(counter + 1)
  }

  const subtract = () => {
    setCounter(counter - 1)
  }

  return { add, subtract }
}
```

Her er selve testen som feiler

```
import { act, renderHook } from '@testing-library/react-hooks'

import { useCounter } from '@hooks/useCounter'

describe('Testing useCounter hook', () => {

  it('should add +1 to counter', () => {
    const { result } = renderHook(useCounter)

    act(() => result.current.add())
    expect(result.current.counter).toBe(1)
  })

  it('should subtract -1 to counter', () => {
    const { result } = renderHook(useCounter)

    act(() => result.current.subtract())
    expect(result.current.counter).toBe(-1)
  })
})
```