

Komplett systemløsning og design av en webapplikasjon for orkestersceneoppsett ved bruk av React-Node.js og Firebase integrasjon

Systemdokumentasjon

Versjon 3.0

Dokumentet er basert på Systemdokumentasjon utarbeidet ved NTNU. Revisjon og tilpasninger til bruk ved IDER, DATA-INF utført av Carsten Gunnar Helgesen, Svein-Ivar Lillehaug og Per Christian Engdal. Dokumentet finnes også i engelsk utgave.

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
24.04.2023	1.0	Førsteutkast	Maksim Ohvrill
07.05.2023	2.0	Oppdatering av kapittel 3 og 4	Maksim Ohvrill
21.05.2023	3.0	Øvrige finjusteringer før innlevering	Andreas Foss Wergeland

Innholdsfortegnelse

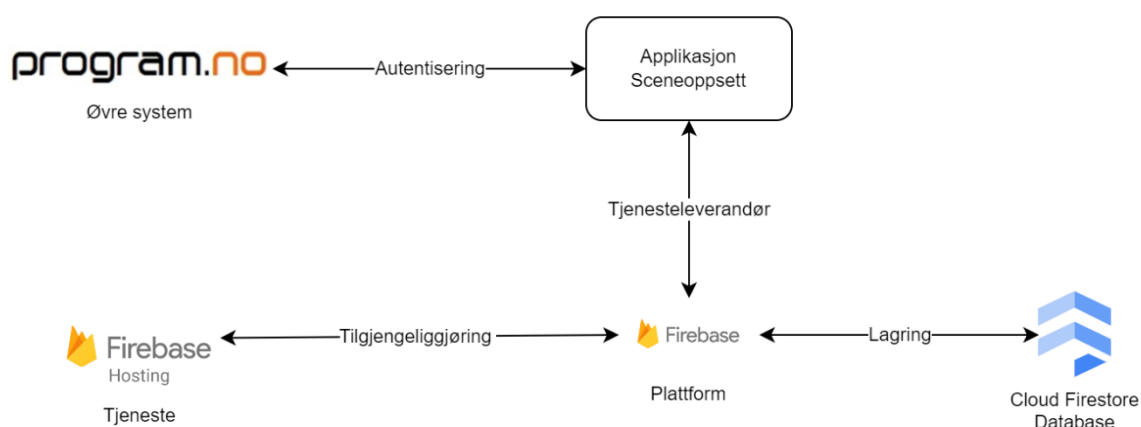
1	Innledning	1
2	Arkitektur	1
3	Prosjektstruktur	2
4	Servertjenester.....	3
5	Sikkerhet	3
6	Installasjon og kjøring.....	3
7	Dokumentasjon av kildekode.....	4
8	Kontinuerlig integrasjon og testing.....	6
9	Referanser	8

1 Innledning

Dette dokumentet har som formål å dokumentere systemet til applikasjonen Sceneoppsett ved å benytte utdrag fra README-dokumentet, som er tilgjengelig via GitHub-lagringsplassen (Microsoft, 2023a) for prosjektet. Dette skal videre legges som vedlegg i prosjektets hovedrapport. Dokumentet skal også gi en detaljert forklaring på prosjektets arkitektur, struktur, database og øvrige tjenester, samt dokumentere prosessen for hvordan applikasjonen kan kjøres for videre utvikling og hvor det gjøres tilgjengelig.

2 Arkitektur

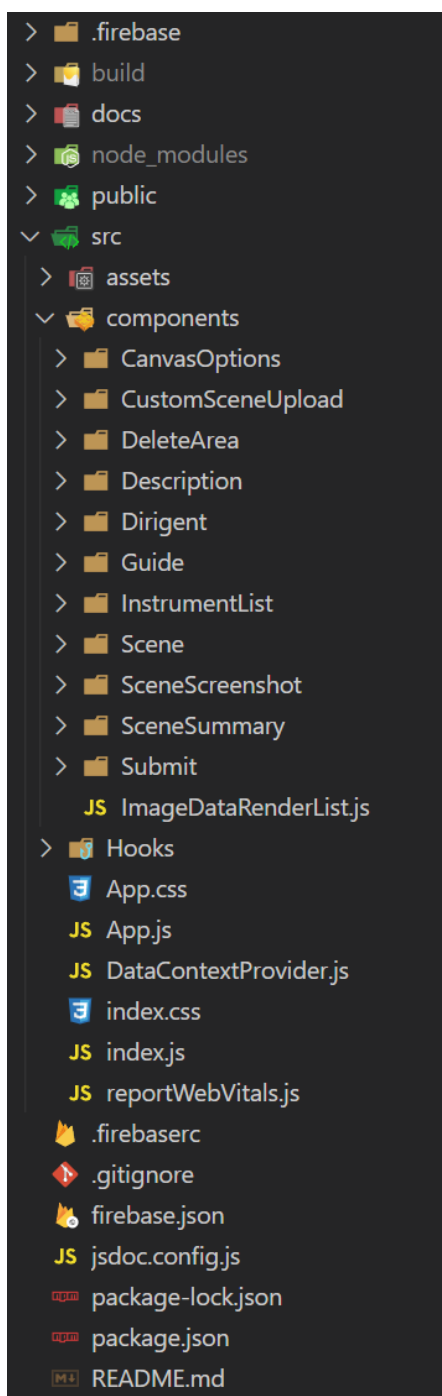
Den følgende figuren gir en initiell skisse over applikasjonens arkitektur for Sceneoppsett. I utgangspunktet var planen å integrere databaseløsningen Cloud Firestore (Google, 2023a), som er en del av Firebase-plattformen, for å lagre sceneoppsett generert ved hjelp av applikasjonen. På grunn av prosjektets tidsbegrensninger ble dette imidlertid ikke gjennomført. Selv om autentiseringsprosessen ikke er en del av applikasjonens domene, er den fortsatt illustrert i det følgende for å fremme forståelsen av hvordan applikasjonen skal integreres i oppdragsgiverens systemer. Videre er systemet gjort tilgjengelig via en tjeneste levert av Firebase-plattformen, kjent som Firebase Hosting (Google, 2023b). Denne tjenesten gjør det mulig å tilgjengeliggjøre et ferdigbygd React-prosjekt via deres servere ved å opprette en nettadresselink til applikasjonen slik at den kan åpnes via en nettleser.



Figur 2-1 Arkitekturskisse for applikasjon Sceneoppsett

Ettersom databaseløsningen ikke ble implementert, vil det være opp til oppdragsgivers utviklere å gjennomføre dette. Grunnlaget for implementasjonen er lagt ved at applikasjonen kan generere en JSON-streng av sceneoppsettet som skal lagres i databasen, samt konvertere det tilbake til et sceneoppsett ved innhenting fra databasen.

3 Prosjektstruktur



Figur 2-1 Utklipp av prosjektstruktur fra VSCode

Dette dokumentet gir en beskrivelse av prosjektets struktur, som er illustrert i følgende figur, et utklipp fra Integrated Development Environment (IDE) - Visual Studio Code (VSCode). VSCode ble benyttet som utviklingsmiljø for applikasjonen Sceneoppsett. Her er det mulig å observere mapper dedikert til dokumentasjon, som ble brukt til å generere nettsider ved hjelp av JSDoc (Mathews, 2017). JSDoc er et verktøy som genererer nettsidedokumentasjon fra kommentarer i koden.

I tillegg gir figuren en oversikt over applikasjonens kildemappe, som inneholder de forskjellige komponentene den består av. Applikasjonen inkluderer også flere Javascript-filer for å laste inn nevnte komponenter. Det finnes også filer dedikert til bygging av applikasjonen og dens tilgjengeliggjøring via Cloud Firestore-tjenesten, som er tilgjengelig via Firestore-plattformen.

Prosjektstrukturen inneholder også en fil for README-dokumentasjon. Denne filen brukes av GitHub-lagringsstedet for å generere prosjektets dokumentasjonsside.

4 Servertjenester

Servertjenester prosjektet benytter anskaffes via tredjepartstjenester slik som Firebase og GitHub plattformer, for tilgjengeliggjøring av testapplikasjonen og den endelige applikasjonen samt lagring og deling av kildekode.

5 Sikkerhet

Sikkerhetsaspektet ved utviklinga av denne applikasjonen består av at all kode ved utvikling kjøres på lokale servere som vil ikke være spesielt utsatt for ytre trusler. Ved tilgjengeliggjøring av applikasjon for testing gjennom Firebase plattformen, er tilgang til en konto for plattformen gitt der bare utviklerne for prosjektet har tilgang til. Kildekoden for prosjektet ligger på en privat GitHub "repository" og er ikke tilgjengelig offentlig. Når testversjonen og den ferdige versjonen av applikasjonen gjøres tilgjengelig er det private linker som kan ikke søkes opp gjennom søkemotor som opprettes, all videre sikkerhet av servere som applikasjonen blir tilgjengelig fra administreres av Firebase plattformen driftet av Google, som anses som trygg leverandør av sine tjenester, noe som minimerer eventuelle sikkerhetsbekymringer. Dokumentasjonen for prosjektet gjøres tilgjengelig via GitHub pages (Microsoft, 2023b), som er også lagres på privat "repository" og er driftet av GitHub, en leverandør anerkjent for pålitelighet og sikkerhet i sine tjenester.

6 Installasjon og kjøring

For å komme i gang med applikasjon Sceneoppsett, klon "repository" og kjør "npm install" for å installere nødvendige avhengigheter. Deretter kjør "npm start" for å starte utviklingsserveren.

Getting Started

To get started with SceneOppsett, clone the repository and run `npm install` to install the necessary dependencies. Then, run `npm start` to start the development server.

Figur 6-1 Utklipp fra GitHub repository for installasjon og kjøring

7 Dokumentasjon av kildekode

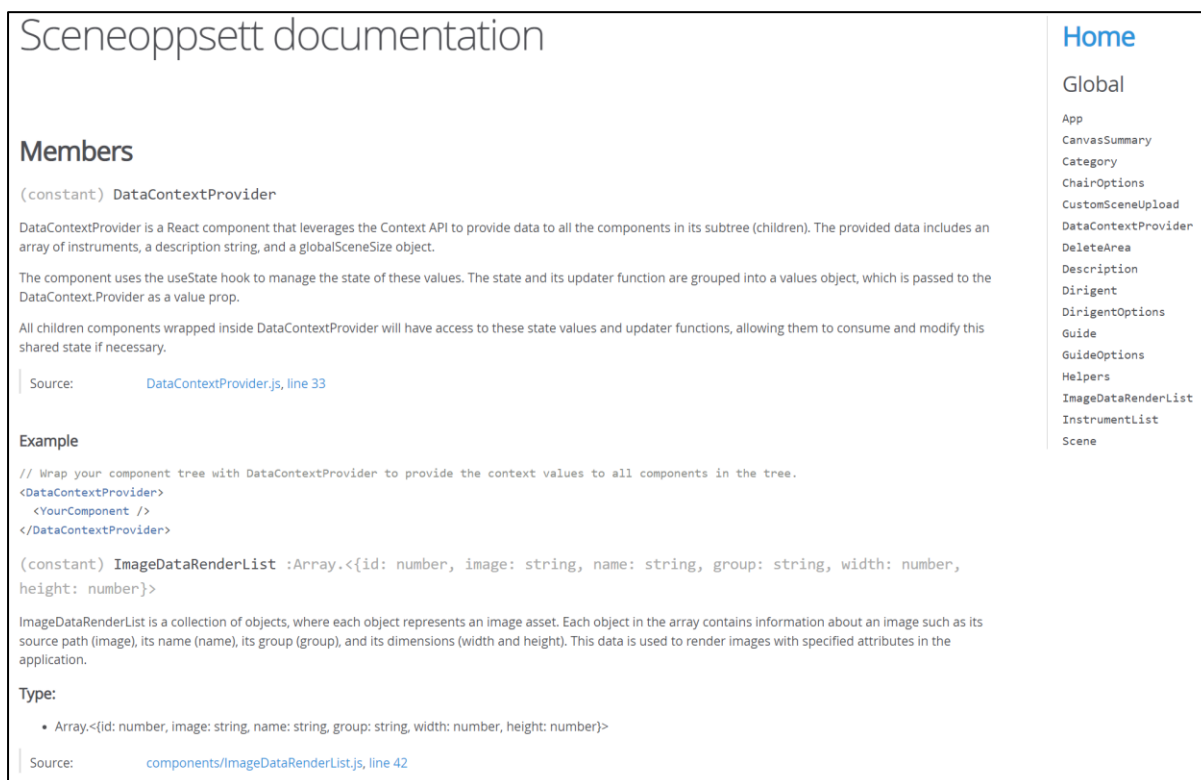
Dokumentasjonen av kildekode er utført ved bruk av JSDoc, som nevnt tidligere. Lenken til denne dokumentasjonen er tilgjengelig på GitHub-lagringsstedet og er inkludert som vedlegg i hovedrapporten.

Documentation

- Documentation is provided here: [JSDoc](#).

Figur 7-1 Utklipp fra GitHub repository for lenke til dokumentasjon

Ved å følge lenken som vises i figuren ovenfor fra GitHub-lagringsstedet, kan leseren få tilgang til dokumentasjonen for kildekodens komponenter og funksjoner. All dokumentasjon er skrevet på engelsk, et bevisst valg for å gjøre dokumentasjonen mer tilgjengelig for flertallet av utviklere som potensielt vil arbeide med applikasjonen.



Sceneoppsett documentation

Members

(constant) `DataContextProvider`

`DataContextProvider` is a React component that leverages the Context API to provide data to all the components in its subtree (children). The provided data includes an array of instruments, a description string, and a `globalSceneSize` object.

The component uses the `useState` hook to manage the state of these values. The state and its updater function are grouped into a values object, which is passed to the `DataContext.Provider` as a value prop.

All children components wrapped inside `DataContextProvider` will have access to these state values and updater functions, allowing them to consume and modify this shared state if necessary.

Source: [DataContextProvider.js, line 33](#)

Example

```
// Wrap your component tree with DataContextProvider to provide the context values to all components in the tree.
<DataContextProvider>
  <YourComponent />
</DataContextProvider>
```

(constant) `ImageDataRenderList` :Array.<{id: number, image: string, name: string, group: string, width: number, height: number}>

`ImageDataRenderList` is a collection of objects, where each object represents an image asset. Each object in the array contains information about an image such as its source path (image), its name (name), its group (group), and its dimensions (width and height). This data is used to render images with specified attributes in the application.

Type:

- Array.<{id: number, image: string, name: string, group: string, width: number, height: number}>

Source: [components/ImageDataRenderList.js, line 42](#)

Home

- Global
- App
- CanvasSummary
- Category
- ChairOptions
- CustomSceneUpload
- DataContextProvider
- DeleteArea
- Description
- Dirigent
- DirigentOptions
- Guide
- GuideOptions
- Helpers
- ImageDataRenderList
- InstrumentList
- Scene

Figur 7-2 Utklipp fra JSDoc dokumentasjonsside for applikasjonen

Figuren ovenfor gir en detaljert forklaring av den gitte JavaScript-klassen: dens formål, den funksjonaliteten den tilbyr, og hvordan den kan brukes i koden.

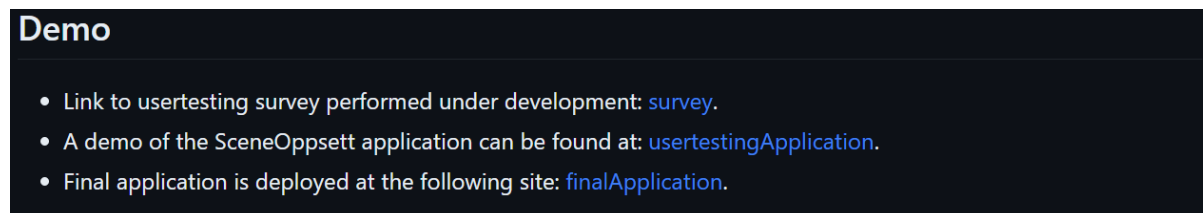
```
JS CustomSceneUpload.js X
src > components > CustomSceneUpload > JS CustomSceneUpload.js > CustomSceneUpload > handleChange
You, 2 weeks ago | 2 authors (You and others)
1 import React, { useState } from "react";
2 import "./CustomSceneUpload.css";
3 import InsertLinkIcon from "@mui/icons-material/InsertLink";
4
5 /**
6  * CustomSceneUpload is a functional React component that allows users to upload
7  * custom images for their own scenes. The component accepts files in png, jpg, or pdf format.
8  *
9  * @param {object} props - The properties passed to this component.
10 * @param {React.RefObject} props.sceneRef - A React ref object that refers to the scene element.
11 *
12 * @returns {JSX.Element} A form element with the class 'formContainer',
13 * that contains inputs for file selection, and buttons for deleting and uploading the selected file.
14 *
15 * @example
16 * // To include the CustomSceneUpload component in a parent component:
17 * <CustomSceneUpload sceneRef={sceneRef} />
18 */
19 function CustomSceneUpload(props) {
20   const [file, setFile] = useState();
21
22   function handleChange(event) {
23     setFile(event.target.files[0]);
24   }
25
26   function handleDelete() {
27     const sceneDiv = props.sceneRef.current;
28     if (sceneDiv.lastChild.className === "uploadedImage") {
29       sceneDiv.removeChild(sceneDiv.lastChild);
30       sceneDiv.children[0].style.display = "block";
31       sceneDiv.children[1].style.display = "block";
32     }
33     if (!file) {
34       setFile(null);
35     }
36   }
37 }
```

Figur 7-3 Utklipp fra kildekode med dokumentasjon i kode

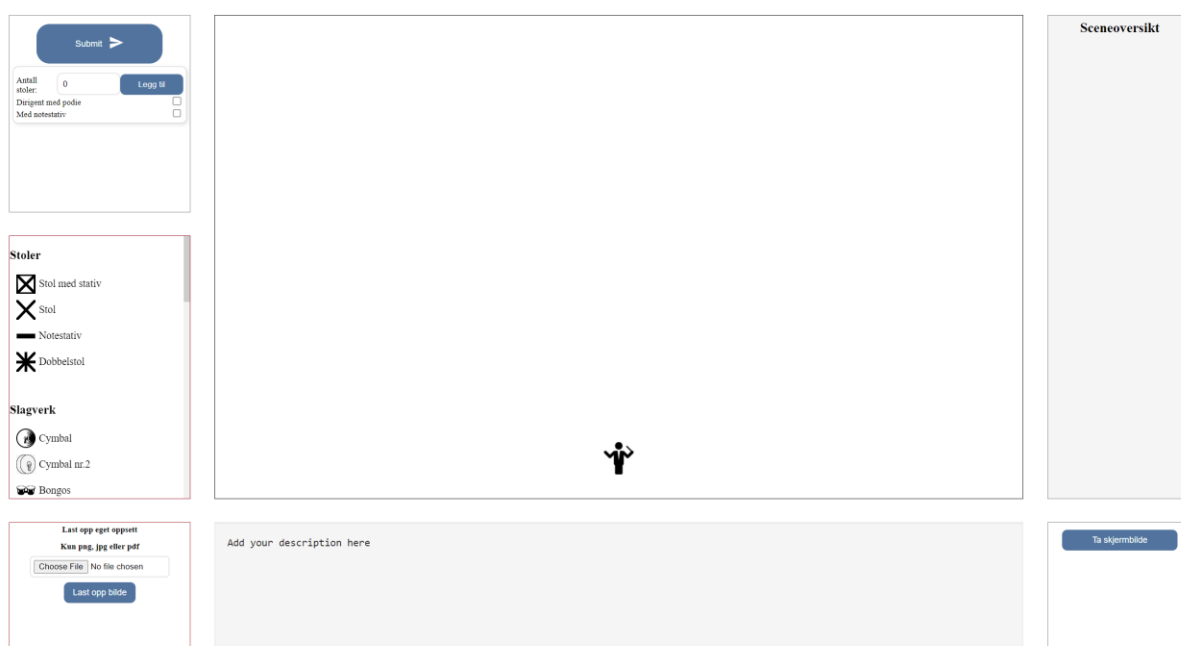
Kildekodeutklippet viser hvordan koden ble dokumentert, som deretter dannet grunnlaget for genereringen av JSDoc-nettsiden. Denne nettsiden inneholder fri tilgang til dokumentasjonen, noe som resulterer i en ryddig og oversiktlig dokumentasjon av applikasjonen Sceneoppsett.

8 Kontinuerlig integrasjon og testing

Gjennom bruk av React, var applikasjonen kontinuerlig i en kjørbar tilstand, noe som tillot konstant testing av implementerte funksjoner. I følgende illustrasjon kan det observeres lenker for applikasjonen gjort tilgjengelig gjennom README dokumentet til prosjektet.

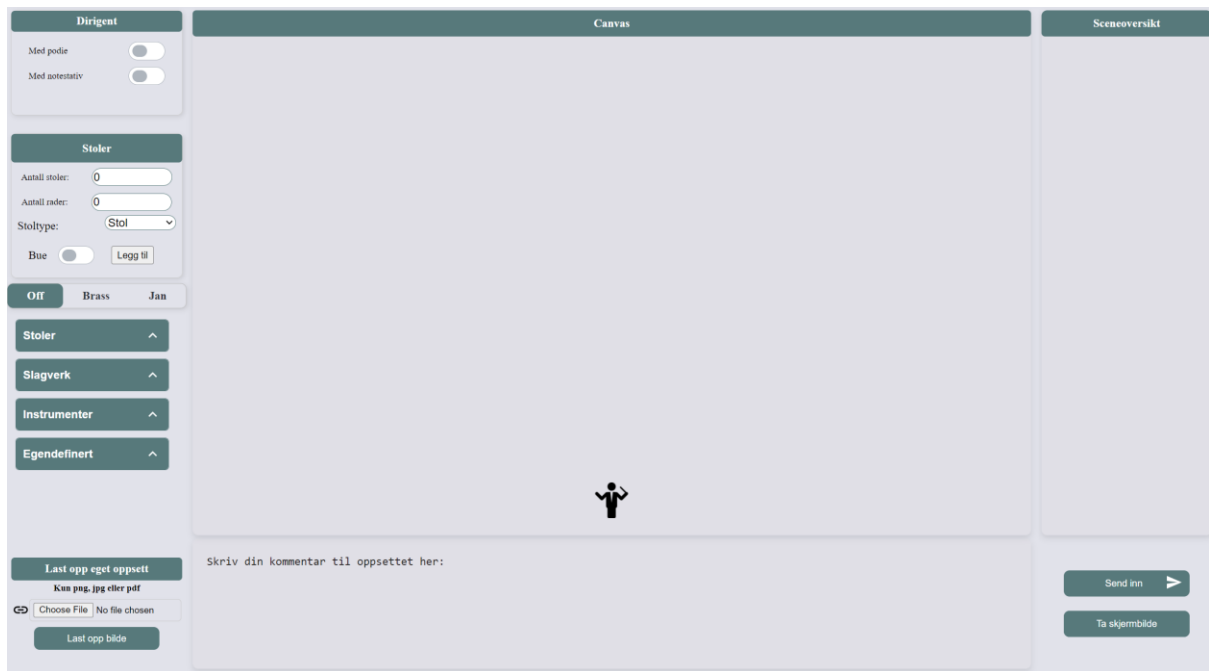


Figur 8-1 Utklipp fra GitHub repository for lenker til applikasjonen



Figur 8-2 Illustrasjon av applikasjon under testfase

Utklipp ovenfor er applikasjon under testfase som ble blant annet brukt til brukerundersøkelse -og testing som er den første lenken i README dokumentet til prosjektet.



Figur 8-3 Illustrasjon av ferdigstilt applikasjon

Dette utklippet illustrerer ferdigstilt applikasjon som ble vedlagt under README dokumentet til prosjektet som gjøres tilgjengelig via GitHub lagringsstedet og er vedlagt ved hovedrapporten til prosjektet.

9 Referanser

Google (2023a) *Cloud Firestore*. Tilgjengelig fra: <https://firebase.google.com/docs/firestore> (Hentet: 2023-03-20).

Google (2023b) *Firebase Hosting*. Tilgjengelig fra: <https://firebase.google.com/docs/hosting> (Hentet: 2023-04-16).

Mathews, M. (2017) *Use JSDoc: Index*. Tilgjengelig fra: <https://jsdoc.app/> (Hentet: 2023-05-21 2023).

Microsoft (2023a) *GitHub*. Tilgjengelig fra: <https://github.com> (Hentet: 2023-05-21).

Microsoft (2023b) *GitHub Pages*. Tilgjengelig fra: <https://pages.github.com/> (Hentet: 2023-05-16).