

0.0.1 Komponent 3: Administrations hjemmeside

Denne komponent har til formål at håndtere alle de administrative opgaver i systemet. Dette består af 4 delkomponenter:

- Den første delkomponent gør det muligt at tilføje en bus til systemet, fjerne den, eller rediger i en bus der allerede findes i systemet.
- Derefter skal det være muligt at tilføje eller fjerne en bus fra en rute der findes i systemet.
- Den tredje delkomponent gør det muligt at kunne oprette en hel ny busrute i systemet, ændrer i en allerede eksisterende busrute, eller slette en fra systemet.
- Den sidste delkomponent består af muligheden for at kunne tilføje, ændre samt fjerne busstoppesteder fra systemet.

Alle disse delkomponenter udgøre tilsammen en vigtig del af systemet, da uden nogle af dem vil det ikke være muligt at kunne få vist nogle af overstående ting på mobil applikationen.

Design:

Hjemmesiden er blevet implementeret ved brug af Microsoft ASP.NET MVC 4 frameworket. Dette gør det nemt og hurtigt at implementere en sofistikeret og moderne hjemmeside, der følger gode design principper. MVC står for Model-View-Controller og følger de samme principper som MVVM angående 'separation of concerns'.

For at kunne indtegne busruter og stoppesteder skal der bruges et kort, til dette er der blevet brugt Google maps samt Google Directions API.

Hjemmesiden består af 4 view, først og fremmest et view til startsiden der linker til de 3 andre views, der består af et der håndterer alt vedrørende busser, et til stoppesteder samt et til busruter.

Det første view der håndterer alt om busserne består af 2 dele. Første del gør det muligt at tilføje en ny bus til systemet, fjerne en bus fra systemet og rediger ID'et for en bus. Dette er blevet implementeret ved at når view'et bliver loaded, bliver en JavaScript function kaldet, der kalder funktionen GetAllBusses() i controlleren, der henter alle busser der er i

MySQL databasen. Til at lave dette kald fra JavaScript til controlleren, bliver der brugt ajax. Ajax gør det muligt at udvæksle data med controlleren og updatere view'et uden at skulle reloade hele websiden.

Kodeudsnit 1: Ajax kald til controller funktionen 'GetAllBusses'

```
1      $.ajax({
2          type: "POST",
3          url: "Bus/GetAllBusses",
4          dataType: "json",
5          success: function (result) {
6              var select = document.getElementById("busses");
7              select.options.length = 0;
8              for (var i = 0; i < result.length; i++) {
9                  select.options.add(new Option(result[i]));
10                 ListOfAllBusses.push(result[i]);
11             }
12         }
13     });
```

Dette eksempel på et ajax kald, kalder GetAllBusses(), dette er en funktion der ligger i controlleren, som henter en liste af alle bussernes ID'er fra MySQL databasen. Når controlleren er færdig returnere den et json object, og callback funktionen der er defineret i success parameteren af ajax bliver kaldt. Result parameteren på callback funktionen er returnværdien fra controller funktionen, der i dette tilfælde er et json object, der indeholder en liste af alle bussernes ID'er, hentet fra MySQL databasen. Callback funktionen løber igennem listen af ID'er og tilføjer dem til et HTML select element. Dette gør det muligt for administratoren at se hvilke busser der er gemt i databasen.