

# Timebox2- Add/Edit/Remove facilities

## Oversigt

<b>OpgaveNavn</b>	Add/Edit/Remove facilities		
<b>Implementering af krav</b>	Hel implementering af: WEB-16, WEB-17, WEB-18		
<b>Udført af</b>	Jan	<b>Dato</b>	01-10-2021
<b>Timebox</b>	2	<b>Område</b>	Website

## Contents

<b>INTRODUKTION.....</b>	<b>1</b>
<b>ANALYSE.....</b>	<b>2</b>
<b>DESIGN.....</b>	<b>2</b>
<b>IMPLEMENTERING .....</b>	<b>2</b>
<b>VERIFIKATION .....</b>	<b>3</b>
<b>TESTRESULTAT.....</b>	<b>4</b>
<b>KONKLUSION .....</b>	<b>5</b>
<b>REFERENCER .....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>

## Introduktion

Det skal være muligt at kunne indføre, redigere og slette faciliteter i databasen, for at idéen bag hjemmesiden og projektet kan fungere.

Udvikling og implementering af de før nævnte facilitets funktionaliteter, vil være beskrevet i dette dokument.

## Analyse

De 3 funktionaliteter der skal laves, for at denne opgave kan færdiggøres er:

- Indsætte nye entries ind i en database tabel.
- Overskrive/opdatere eksisterende entries i en database tabel.
- Slette eksisterende entries i en tabel.

Ud af disse 3, er den eneste "nye" funktionalitet, at kunne indsætte nye entries, hvor de 2 andre er blevet lavet i andre opgaver til hjemmesiden, som kan kopieres, for kun at ændre lidt i HTML tabellerne, URL adresserne, og hvilke database objekter der bliver lavet queries til.

For at lave en ny entry, i en database, vil der kunne laves en instans af en model, i stedet for at lave en query efter en eksisterende entry som hidtil gjort, for derefter udfylde de forskellige fields, og så bruge "model\_instance.save()".

## Design

Frontend designet vil være i stor stil som nævnt i deliverable: "TB1 – Website landing page setup.docx", i henseende til: "Facility page", "Add facility page", "Edit facility page" og "Remove facility page".

For "Edit facility page" og "Remove facility page", vil der dog være den ændring, at der vil være en filter page, som gjort ved deliverable: "TB2 – Edit-other-users.docx" i henseende til filter page.

Backenden vil i stor grad blive udfærdiget som gjort i deliverable: "TB2 – Edit-other-users.docx", dog med den forskel at "error statements" vil blive vist på input siderne, i stedet for en separat side, da dette mindsker antallet af nødvendige filer og kode, samt vil de forskellige queries blive gjort til Facilities databasemodellen, i stedet for Users databasemodellen.

## Implementering

Der vil under implementeringen kun blive vist "add facilities" og error handling, da implementeringen for de 2 andre er som i deliverable: "TB2 – Edit-other-users.docx", med kun minimale ændringer, samt er field visning og input modtagning i "add facilities", også som i førnævnte deliverable.

På Figur 1 er der vist hvordan error handling er sat op, samt hvor queries for denne opgave vil blive dirigeret hen til:

- **Linje 250:** gør at der ikke bliver vist noget tekst, i tilfælde af der ikke er nogen fejl.
- **Linje 252:** opfanger "POST requesten" fra hjemmesiden, hvis "Create facility" knappen bliver trykket på.
- **Linje 253:** henter en form for facilities modellen, som indeholder alle modellens fields.
- **Linje 254-256:** hvis formen er korrekt udfyldt, bliver den gemt som ny en ny entry.
- **Linje 257-265:** i tilfælde formen ikke er korrekt udfyldt, vil der blive vist en fejl meddelelse på siden, som indeholder hvad der blev gjort forkert. For vise

meddelelsen (og i tilfælde der er mere end 1 fejl), bliver der loopet igennem formens indbyggede `.error` klasse, hvor ved at bruge `.error` klassens `as_data()` funktion, bliver fejl beskeden returneret i følgende format: "error-dict{error-type-list(explicit-error-list())}. Efter at error statementen er loopet igennem, bliver hver error og explicit error, samlet i en en dict, der bliver passeret til HTML'en.

```
241 def facility_add(request):
242     """This funtion is for adding a new facility to the database..."""
243
244     # Initial "welcome" text
245     disp_text = {'': ''}
246
247     if request.method == 'POST':
248         form = AddFacility(request.POST)
249         if form.is_valid():
250             form.save()
251             return redirect('facility')
252         else:
253             # error text and formatting, for changing "welcome" text
254             disp_text = {"Errors": ""}
255             error_text = form.errors.as_data()
256             for key, value in error_text.items():
257                 for items in value:
258                     for item in items:
259                         disp_text["Field " + str(key) + ":"] = item
260             return render(request, "facility_add.html", {'disp_text': disp_text})
261
262     return render(request, "facility_add.html", {'disp_text': disp_text})
```

Figur 1 backend - add facility

For at vise fejlbeskeden på hjemmesiden, er der blevet gjort som på Figur 2, hvor den passerede "dict", bliver loopet igennem og fremvist i en HTML table.

```
35 <table>
36     {% for key, value in disp_text.items %}
37         <tr>
38             <td style="...">{{ key }}</td>
39             <td style="...">{{ value }}</td>
40         </tr>
41     {% endfor %}
42 </table>
```

Figur 2 frontend - error handling

## Verifikation

Der skal testes følgende punkter:

- Der skal kunne indsættes en ny facilitet.
- Faciliteters navn felt skal være unik.
- Korrekte fejlmeddelelser skal vises på siden, hvorved fejlen er hændt.
- Der skal kunne redigeres i eksisterende faciliteter.
- Der skal kunne slettes en eksisterende facilitet.

Der er antaget at følgende ting er gjort på forhånd:

- Serveren er oppe at køre.
- Der er logget ind som "office user", og dirigeret hen til "Add/Edit/Remove" siden.
- Der er lavet en admin bruger, samt der vides hvordan man kommer hen til Facilities tabellen, på admin siden.


Tabel 1: Tests til verifikation af opgave

Test	Test Steps	Pre-requisites	Pass-betingelser	Resultat
Indsæt fac.	<ol style="list-style-type: none"> <li>tryk på "Add facility".</li> <li>udfyldt felter.</li> <li>tryk på "Create facility".</li> </ol>	N/A	på admin siden, under facilities er der oprettet et korresponderende entry.	Bestået
Rediger fac.	<ol style="list-style-type: none"> <li>tryk på "Edit facility".</li> <li>vælg en facilitet, og tryk på "edit facility".</li> <li>ændre navn til "test fac 2".</li> <li>tryk "accept edits".</li> </ol>	der er en allerede eksisterende facilitet, med navnet "test fac".	på admin siden, under facilities er navnet på korresponderende entry ændret til "test fac 2".	Fejlet
Slet fac.	<ol style="list-style-type: none"> <li>tryk på "Remove facility".</li> <li>vælg en facilitet, og tryk på "remove facility".</li> <li>tryk på "remove facility"</li> </ol>	der er en allerede eksisterende facilitet	på admin siden, under facilities er det slettede facilitet fjernet.	
Fejl medd.	<ol style="list-style-type: none"> <li>tryk på "Add facility"</li> <li>udfyld felter med det samme som i test "Indsæt fac.".</li> <li>tryk på "Create facility".</li> </ol>	der er en eksisterende facilitet, med samme info som i test "Indsæt fac."	på "add facility" siden vil der blive vist en besked i toppen, med en field error, at navnet indtastet allerede eksisterer.	bestået

## Testresultat

### Test: "Indsæt fac."

På Figur 3 og Figur 4, kan resultat for denne test ses.



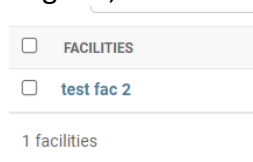
Figur 3 add fac 1



Figur 4 add fac 2

### Test: "Rediger fac."

På Figur 5, kan resultat for denne test ses.



Figur 5 edit fac

### Test: "slet fac."

På Figur 6, kan resultat for denne test ses.

---

0 facilities

---

Figur 6 remove fac

### Test: "fejl medd."

På Figur 7, kan resultat for denne test ses.



**Add facility**

**Errors**

**Field name:** Facilities with this Name already exists.

Name: test fac

Address: test addr

Owner: test owner

Key: test key

Create facility

back

Figur 7 error

### Konklusion

Implementeringen af disse funktionaliteter blev gjort uden problemer og grundet at mange af de grundlæggende backend og frontend elementer var lavet igennem andre opgaver, og derfor kun krævede små ændringer, kunne det gøres forholdsvis hurtigt.