

# Timebox 3- Facility access

## Oversigt

<b>OpgaveNavn</b>	Implementering af "facility access give" og "facility access remove"		
<b>Implementering af krav</b>	Hel implementering af kravene: <ul style="list-style-type: none"><li>- Web-5 "Det skal være muligt at fjerne adgang til faciliteter "</li><li>- Web-6 "Det skal være muligt at give bruger adgang til faciliteter. "</li><li>- Web-7 "Det skal være muligt at definere en variable tidsperiode for adgangen til faciliteter per bruger. "</li></ul>		
<b>Udført af</b>	Marc	<b>Dato</b>	05-10-2021
<b>Timebox</b>	3	<b>Område</b>	Website

## Contents

INTRODUKTION.....	1
ANALYSE OG DESIGN.....	2
IMPLEMENTERING .....	2
VERIFIKATION .....	7
TESTRESULTAT.....	8
KONKLUSION .....	10

## Introduktion

Denne opgave omhandler implementering af at give og fjerne adgang til faciliteter for andre brugere af systemet.

## Analyse og Design

Største delen af analysen er i "Launch Phase" rapportens afsnit 4, 5, 6 og 7.

Det er tre hovedelementer i de to opgaver "facility access give" og "facility access remove":

- Det skal være muligt at give en bruger adgang til en facilitet.
- Det skal være muligt at give en tidsbestemt adgang.
- Det skal være muligt at fjerne adgang til en facilitet.

Opgaverne skal udvikles i appen "office\_user", da det er brugertypen "office user" som skal anvende funktionaliteten.

Opgaven "facility access give" vil blive implementeret i funktionerne "facility\_access\_give" og "facility\_access\_give\_filter", opgaven "facility access remove" vil blive implementeret i funktionen "facility access remove".

## Implementering

Funktionen "facility\_access\_give\_filter" er implementeret som på Figur 1, først filtreres de unikke facilitets lokationer, derefter hentes alle facilitets navne. Når en bruger har lavet en "POST request", har brugeren valgt en lokation, som facilitetsnavnene filtreres efter.

```
230 # Query all unique companies in the database
231 location = Facilities.objects.order_by().values_list('location', flat=True).distinct()
232 # No company chosen yet, so query all users in the database
233 names = Facilities.objects.order_by()
234
235 if request.method == 'POST':
236     data = request.POST['location_list']
237     if data != "None":
238         # Query database for users, filtered by chosen location
239         names = Facilities.objects.order_by().filter(location=data)
240         user_header_text = "Facilities for: " + str(data)
241         # Render page with filtered users, instead of all users
242         render_dict = {
243             'location': location,
244             'names': names,
245             'param': data,
246             'header_text': user_header_text}
247         return render(request, "facility_access_give_filter.html", render_dict)
248
249 # Text for a header
250 user_header_text = "All facilities:"
251 # Default render with all companies, and all users
252 render_dict = {
253     'location': location,
254     'names': names,
255     'header_text': user_header_text}
256 return render(request, "facility_access_give_filter.html", render_dict)
```

Figur 1 Funktionen "facility\_access\_give\_filter"

Den HTML der bliver renderet er sat op som tidligere opgaver, den største ændring her er, som man kan se på Figur 2, "form action" som tager brugeren hen til "facility access give" siden.

```
46      <!-- BOX-DROPDOWN: names -->
47      <h1>{{ header_text }}</h1>
48      <form action={% url 'facility_access_give' %}>
49      {% csrf_token %}
50      <select name="names_list">
51      {% for val in names %}
52          <option value="None" selected hidden>Choose facility</option>
53          <option value='{{ val.name }}'> {{ val.name }} </option>
54      {% endfor %}
55      </select>
56      <input type="submit" value="Edit facility" style="...">
57  </form>
58  <button onClick="document.location.href = '../' style="...">Back</button>
```

Figur 2 facility\_access\_give\_filter.html

Ved "Facility\_access\_give" funktionen, hentes det facilitetsnavn som var filtreret i "facility\_access\_give\_filter" og anvendes til at finde facilitetsobjektet, se Figur 3.

```
276      # Get facility chosen from "facility_access_give_filter"
277      data = request.GET['names_list']
278      # Query all Users in the database
279      all_user = Users.objects.order_by('name')
280      # If there's no user chosen, redirect back to "facility access give filter"
281      if data == 'None':
282          return redirect("facility_access_give_filter")
283      # Make query based on chosen user from "facility_access_give_filter"
284      facility_choice = Facilities.objects.get(name=data)
```

Figur 3 Funktionen "facility\_access\_give" hent info fra "facility\_access\_give\_filter"

På siden "facility\_access\_give" kan der vælges en bruger og en start-og sluttid, se Figur 4.

```
56      <select name="user">
57      {% for val in users %}
58          <option value="None" selected hidden>Choose user</option>
59          <option value='{{ val.email }}'> {{ val.name }} ({{ val.email }}) </option>
60      {% endfor %}
61      </select>
62      <tr>
63          <td><label for="timer_start">Access from:</label></td>
64          <td><input type="date" id="date" name="timer_start" ></td>
65      </tr>
66      <tr>
67          <td><label for="timer">Access until:</label></td>
68          <td><input type="date" id="date" name="timer" ></td>
69      </tr>
```

Figur 4 "facility\_access\_give.html"

Når det er valgt, hentes informationerne og tjekkes efter om sluttiden er mindre end starttiden, hvis den er, kommer der en fejlbesked, se Figur 5.

```
286     if request.method == 'POST':
287         access = JoinTable()
288         # get User object from unique Email
289         access.user = Users.objects.get(email=request.POST['user'])
290         # get facility info
291         access.facility = Facilities.objects.get(name=facility_choice.name)
292         # get date info
293         access.timer_start = request.POST['timer_start']
294         access.timer = request.POST['timer']
295         # check if date is ok
296         if access.timer < access.timer_start:
297             print("Im here in error")
298             return redirect('access_time_error')
```

Figur 5 Funktionen "facility\_access\_give" ved POST

Hvis sluttiden ikke er mindre end starts tiden, tjekkes derefter om der allerede er givet adgang til den pågældende facilitet, for den specifikke bruger, hvis der er, opdateres tiden med det nye tidsstempel, ellers gemmes den ny adgang, se Figur 6.

```
296     if access.timer < access.timer_start:
297         print("Im here in error")
298         return redirect('access_time_error')
299     else:
300         # Try rewrite timer if the access to same user and facility is already given
301         try:
302             access_exist = JoinTable.objects.get(user=access.user, facility=access.facility)
303             # Get object
304             access_exist_obj = JoinTable.objects.get(pk=access_exist.pk)
305             print(access_exist.pk)
306             access_exist_obj.timer = request.POST['timer']
307             access_exist_obj.timer_start = request.POST['timer_start']
308             access_exist_obj.save()
309             return redirect('facility_access')
310         except JoinTable.DoesNotExist:
311             # Save
312             access.save()
313             return redirect('facility_access')
```

Figur 6 Funktionen "facility\_access\_give" tjeks

Opgaven for at fjerne adgang til et facilitet fra en bruger, sker i funktionen "facility\_access\_remove", hvor der først hentes information fra allerede givet adgang både fra brugere og faciliteter, se Figur 7.

```
336     # Used for html
337     facility_is_chosen = False
338     user_is_chosen = False
339     access_remove = False
340     # No facility location chosen yet, so query all facilities in the database
341     facilities = JoinTable.objects.values_list('facility__name', flat=True).distinct()
342     users = JoinTable.objects.values_list('user__name', 'user__email').distinct()
```

Figur 7 Funktionen "facility\_access\_remove" filter

HTML'en forgår på same måde som før, så det vises ikke her, men kan ses i bilaget med koden.

Når der valgt en facilitet der skal fjernes adgang fra, filtreres der efter hvilke bruger der har adgang, som vises i HTML'en, ses Figur 8.

```
344     # Facility is chosen:
345     if request.method == 'POST':
346         if 'sel_facility' in request.POST:
347             data = request.POST['facility']
348             if data != "None":
349                 facility_is_chosen = True
350                 # Set header text
351                 user_header_text = "Facility chosen: " + str(data)
352                 # Query all unique users in the database that has access to chosen facility
353                 users = JoinTable.objects.filter(facility__name=data).values_list('user__name',
354                                                                                     'user__email').distinct()
355                 # Render page with filtered facilities, instead of all facilities
356                 render_dict = {
357                     'users': users,
358                     'facilities': data,
359                     'fc': facility_is_chosen,
360                     'uc': user_is_chosen,
361                     'ac': access_remove,
362                     'param': data,
363                     'header_text': user_header_text}
364                 return render(request, "facility_access_remove.html", render_dict)
```

Figur 8 Funktionen "facility\_access\_remove" ved udvælgelse af facilitet

Når en bruger som har adgang, til den valgte facilitet, er valgt, tjekkes der om brugeren har adgang til facilitet, hvis brugeren har adgang, slettes adgangen, hvis ikke vises en fejlmeddeles, se Figur 9.

```
368 elif 'sel_user' in request.POST:
369     data = request.POST['user']
370     if data != "None":
371         user_is_chosen = True
372         list_from_user = data.split("&", 1)
373         try:
374             remove_chosen = JoinTable.objects.get(user__email=list_from_user[0],
375                                                    facility__name=list_from_user[1])
376             # Delete the chosen access
377             remove_chosen.delete()
378             facility_is_chosen = False
379             user_is_chosen = False
380             access_remove = True
381             # Set header text
382             user_header_text = "Access successfully removed"
383             render_dict = {
384                 'fc': facility_is_chosen,
385                 'uc': user_is_chosen,
386                 'ac': access_remove,
387                 'header_text': user_header_text}
388             return render(request, "facility_access_remove.html", render_dict)
389         except JoinTable.DoesNotExist:
390             # Something failed
391             user_header_text = "Something failed try again"
392             render_dict = {
393                 'fc': facility_is_chosen,
394                 'uc': user_is_chosen,
395                 'ac': access_remove,
396                 'header_text': user_header_text}
397             return render(request, "facility_access_remove.html", render_dict)
```

Figur 9 Funktionen "facility\_access\_remove" ved fjernelsen af adgang

## Verifikation

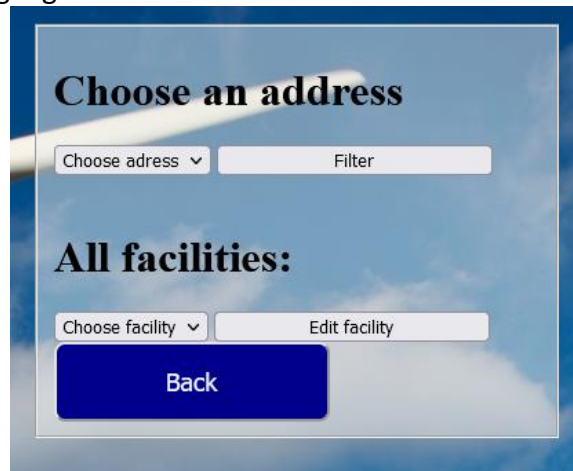
For at teste om implementering virker er testen udført som beskrevet i Tabel 1.

Tabel 1 Tests til verifikation af opgave

Test	Test Steps	Pre-requisites	Pass-betingelser	Resultat
Give adgang til andre bruger	<ol style="list-style-type: none"> <li>1. Login med "office user"</li> <li>2. klik på "Edit facility access"</li> <li>3. Klik på "Give access"</li> <li>4. Vælg enten at filter gennem adresse eller vælg facilitet "test fac1" og klik "Edit facility"</li> <li>5. Vælg brugeren test og en starts- og slutdato og klik på "Give access"</li> <li>6. Gentag 3-4 med facilitet "test fac2" og brugeren "test2"</li> <li>7. Gentag 3-4 med facilitet "test fac3" og brugeren "test3"</li> <li>8. Gentag 3-4 med facilitet "test fac4" og brugeren "test4"</li> <li>9. Gentag 3-4 med facilitet "test fac2" og brugeren "test"</li> <li>10. Gentag 3-4 med facilitet "test fac3" og brugeren "test"</li> <li>11. Gentag 3-4 med facilitet "test fac4" og brugeren "test"</li> <li>12. For at verificere kan man enten logge ind på alle brugerne eller logge ind som admin og se tabellerne</li> </ol>	<ol style="list-style-type: none"> <li>1. Severen kører</li> <li>2. Der er oprettet følgende bruger: Admin, office, test, test2, test3 og test4</li> <li>3. Man har adgang til brugerne</li> <li>4. Der oprettet følgende faciliteter: test fac1, test fac2, test fac3 og test fac4</li> </ol>	<ul style="list-style-type: none"> <li>• Brugeren "test" har adgang til faciliteterne test fac1, test fac2, test fac3 og test fac4 i den givet tidsperiode.</li> <li>• Brugeren "test2" har adgang til facilitet test fac2 i den givet tidsperiode.</li> <li>• Brugeren "test3" har adgang til facilitet test fac3 i den givet tidsperiode.</li> <li>• Brugeren "test4" har adgang til facilitet test fac4 i den givet tidsperiode.</li> </ul>	Bestået
Fjerne adgang til facilitet for en bruger	<ol style="list-style-type: none"> <li>1. Login med "office user"</li> <li>2. klik på "Edit facility access"</li> <li>3. Klik på "remove access"</li> <li>4. Vælg facilitet "test fac2" og klik på "Choose facility"</li> <li>5. Vælg brugeren "test" og klik på "Remove access"</li> <li>6. Klik på "back"</li> <li>7. Gentag punkt 3-6 for faciliteterne "test fac3" og "test fac4"</li> <li>8. Gentag punkt 3-6 for brugeren "test4" og facilitet "test fac4"</li> <li>9. For at verificere kan man enten logge ind på alle brugerne eller logge ind som admin og se tabellerne</li> </ol>	<ol style="list-style-type: none"> <li>1. Man har lige kørt testen "Give adgang til andre bruger" og testen er bestået</li> </ol>	<ol style="list-style-type: none"> <li>2. Brugeren "test" har adgang til faciliteterne test fac1 i den givet tidsperiode.</li> <li>3. Brugeren "test2" har adgang til facilitet test fac2 i den givet tidsperiode.</li> <li>4. Brugeren "test3" har adgang til facilitet test fac3 i den givet tidsperiode.</li> <li>5. Brugeren "test4" ingen adgang</li> </ol>	Bestået

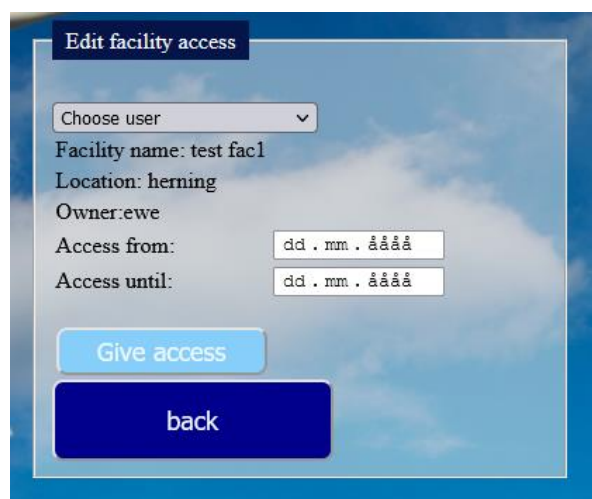
## Testresultat

På Figur 10, kan man se, at der kan vælges en adresse, man vil filtrere på og derefter det facilitet man vil give adgang til.



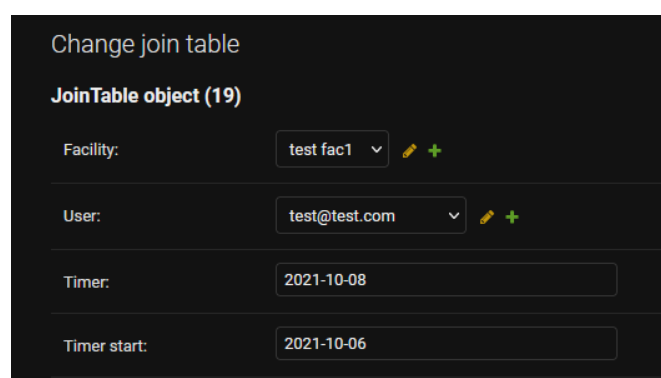
Figur 10 "faciliet\_access\_filter" vælg facilitet

Efter en facilitet er valgt, kan man vælge en bruger og to datoer, en for starts datoen og en for slut datoen, se Figur 11.



Figur 11 "faciliet access give" give adgang

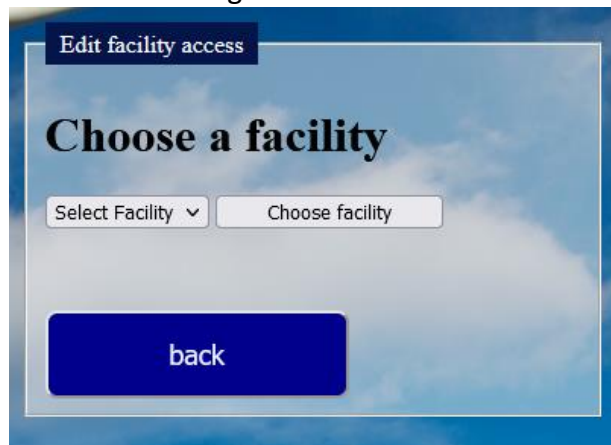
Fra Admin siden kan man se, at en bruger har fået adgang til et facilitet, se Figur 12.



Figur 12 Bruger "test" har adgang til "test fac1"



På Figur 13, kan man se at man kan vælge et facilitet.

The screenshot shows a web interface titled "Edit facility access" in a dark blue header. Below the header, the main heading is "Choose a facility". There are two input fields: "Select Facility" with a dropdown arrow and "Choose facility". At the bottom, there is a large blue button labeled "back". The background of the form is a light blue sky with white clouds.

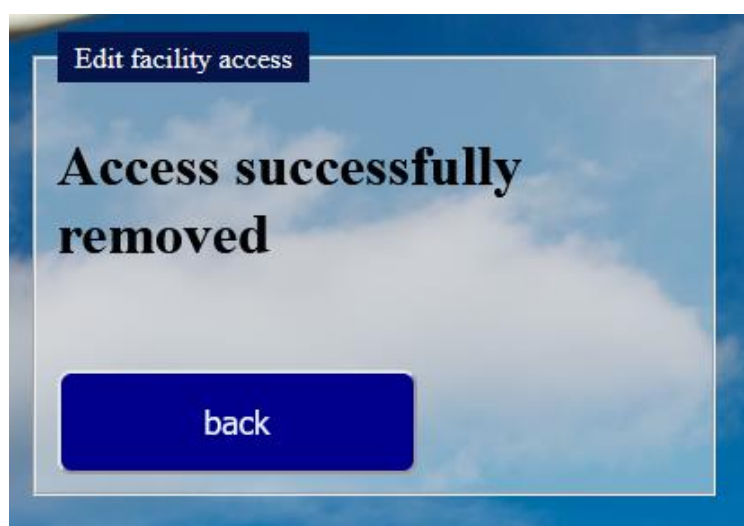
Figur 13 "Remove access" vælg en facilitet

Brugeren "test" skal have fjernet adgangen til "test fac2", se Figur 14.

The screenshot shows the same "Edit facility access" interface, but now it displays "Facility chosen: test fac2". Below this, there is a dropdown menu showing "test (test@test.com)". A red button labeled "Remove access" is prominently displayed. At the bottom, there is a large blue button labeled "back". The background remains a light blue sky with white clouds.

Figur 14 "Remove access" brugeren "test"

Når en adgang er fjernet korrekt, får man en meddelelse som på Figur 15.

The screenshot shows the "Edit facility access" interface with the message "Access successfully removed" in large, bold, black text. At the bottom, there is a large blue button labeled "back". The background is a light blue sky with white clouds.

Figur 15 "Remove access" Adgang fjernet korrekt

## Konklusion

Det er nu muligt at brugertypen "office user" kan fjerne og give adgang for faciliteter til andre bruger.