

Timebox 4 – Automatiske test

Oversigt

OpgaveNavn	Dokumentation af test som er blevet automatiseret		
Implementering af krav	N/A		
Udført af	Marc og Jan	Dato	17-10-2021
Timebox	4	Område	Tests

Contents

INTRODUKTION	1
ANALYSE	2
DESIGN OG IMPLEMENTERING	2
LOGINTESTCASE	2
LOGSTESTCASE	2
FACILITYADDEDITREMOVETESTCASE	2
CREATEEDITDELETEUSERTESTCASE	3
EDITFACILITYACCESSTESTCASE	3
OFFICEPERSONALINFOANDACCESSTESTCASE	3
RAPITESTCASE	4
KONKLUSION	4
REFERENCER	5

Introduktion

Dette dokument beskriver de tests som er automatiseret, det vil være test på websidens del af projektet.

Analyse

Der vil blive anvendt Selenium [1], som er blevet anbefalet af vejlederen. Der er blevet fundet et Python bibliotek, django-selenium [3], som har integreret Django Frameworket med Selenium test værktøj.

Det er fundet inspiration fra guiden *"Testing in Django with Selenium"* [2]. Google Chrome er browseren som vil blive brugt til at test med.

Design og Implementering

Dette afsnit er opdelt efter hver "test case" med overskriften som stemmer overens med den pågældende "class" navn i *test.py*. Hver "class" har flere test funktioner under sig.

LoginTestCase

Her tests hjemmesidens funktionalitet til at logge ind og omdirigering af de forskellige bruger typer.

Der er 3 testfunktioner:

"test_login_field" forløbers sig således:

1. Redirect for field user login.

"test_login_office" forløbers sig således:

1. Redirect for office user login.

"test_login_admin" forløbers sig således:

1. Redirect for admin login.

LogsTestCase

Her tests hjemmesidens funktionalitet til at se log over hvem har brugt adgangen til faciliteterne.

Der er 4 testfunktioner:

"test_create_log" forløbers sig således:

1. Laver en log gennem admin siden.

"test_facility_log" forløbers sig således:

1. Tjekker facility log view funktionen.

"test_user_log" forløbers sig således:

1. Tjekker user log view funktionen.

"test_delete_log" forløbers sig således:

1. Sletter alle logs gennem admin

FacilityAddEditRemoveTestCase

Her tests hjemmesidens funktionalitet for at tilføje, redigerer og fjerne faciliteter.

Test forløbers sig således:

Der er 3 funktioner:

"test_add_facility" forløbers sig således:

1. Tjekker add facility funktionen.

"test_edit_facility" forløbers sig således:

1. Tjekker facility edit funktionen.

"test_remove_facility" forløbers sig således:

1. Tjekker facility remove funktionen.

CreateEditDeleteUserTestCase

Her tests hjemmesidens funktionalitet til at tilføje, redigerer og fjerne bruger.

Test forløbers sig således:

Der er 4 funktioner:

"test_create_user_lp" forløbers sig således:

1. Tjekker create user landing page.

"test_create_user" forløbers sig således:

1. Tjekker "create user" funktionen.

"test_edit_other_user" forløbers sig således:

1. Tjekker edit users funktionen for office users.

"test_delete_other_user" forløbers sig således:

1. Tjekker delete users funktionen for office users.

EditFacilityAccessTestCase

Her tests hjemmesidens funktionalitet for at tilføje og fjerne adgang til faciliteter.

Test forløbers sig således:

Der er 3 funktioner:

"test_edit_facility_lp" forløbers sig således:

1. Tjekker redirects for:
 - a. Facility access landing page
 - b. Give access landing page
 - c. Remove access landing page

"test_give_access" forløbers sig således:

1. Tjekker give access (til et facilitet) funktionen.

"test_remove_access" forløbers sig således:

1. Tjekker remove access (til et facilitet) funktionen.

OfficePersonalInfoAndAccessTestCase

Her tests hjemmesidens funktionalitet til at se ens personlige information og ens facilitets adgang.

Test forløbers sig således:

Der er 3 funktioner:

"test_personal_info_lp" forløbers sig således:

1. Tjekker redirect for info landing page.

"test_facility_access_lp" forløbers sig således:

1. Tjekker redirect for facility access landing page.

"test_edit_personal_info" forløbers sig således:

1. Tjekker funktionaliteten for personal info ved at redigere i ens egen informationer.

RAPITestCase

Her tests hjemmesidens funktionalitet til at lave endpoints for REST Api.

Test forløbers sig således:

Der er 3 funktioner:

"test_curl_create" forløbers sig således:

1. Laver en ny facilitet gennem admin.

"test_RAPI_endpoints" forløbers sig således:

1. Tjekker de forskellige restAPI endpoints.

"test_curl_delete" forløbers sig således:

1. Sletter alle faciliteter gennem admin.

Konklusion

Der er nu implementeret automatiske test for projektets hjemmeside del, som tester systemets "core" elementer

Referencer

[1]	Selenium https://www.selenium.dev/ (sidst besøgt 17-10-2021)
[2]	Selenium med Django guide https://www.ordinarycoders.com/blog/article/testing-django-selenium (sidst besøgt 17-10-2021)
[3]	Python lib: django-selenium https://pypi.org/project/django-selenium/ (sidst besøgt 17-10-2021)