LINKÖPINGS UNIVERSITET

# TDP003 Projekt: Egna datormiljön

## Systemdokumentation

Författare

Kasper Nilsson, kasni325@student.liu.se
Andrei Plotoaga, andpl509@student.liu.se

INSTITUTIONEN FÖR DATAVETENSKAP

Höstterminen 2020
Version 1.0

2020-10-15

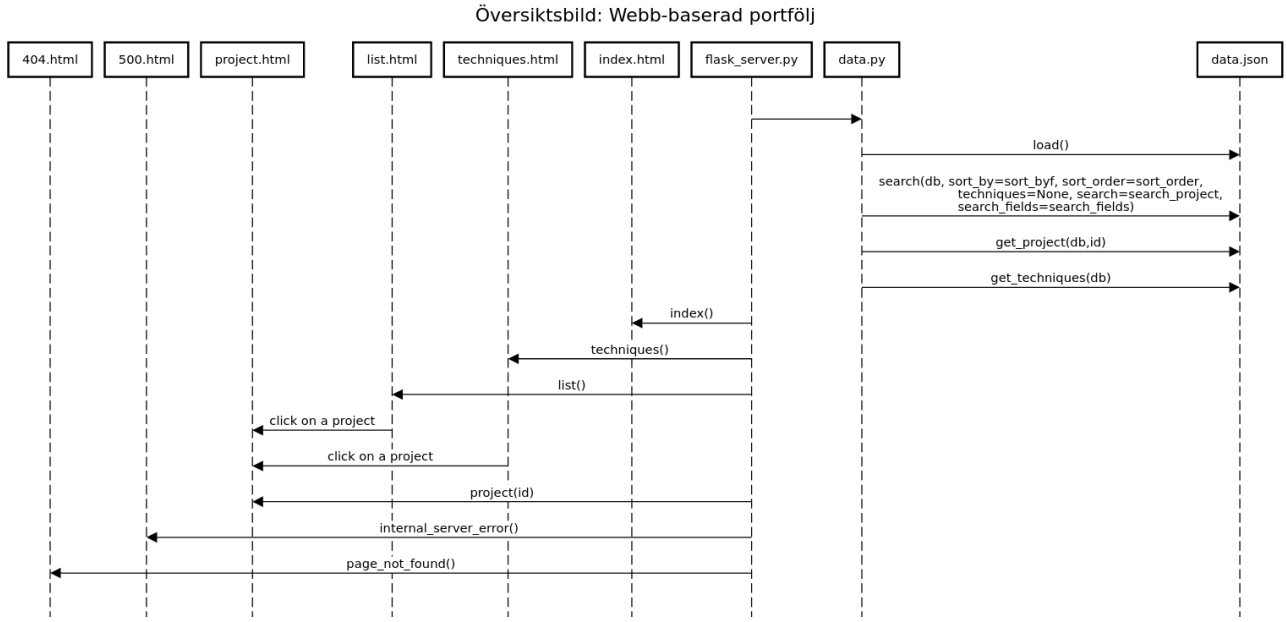
# 1 Revisionshistorik

| Ver. | Revisionsbeskrivning | Datum |
|---|---|---|
| 1.0 | Första utkast | 15/10-20 |

# 2 Översikt av systemet

Figuren nedan visar en översiktsbild av systemet. Det finns ett datalager som består av samtliga projekt sparade i en JSON-fil kallad “data.json”. Sedan finns det ett presentationslager som använder sig av python-modulen “flask” som webbramverk sparad i filen “flask_server.py”. De HTML-mallar som flask_server.py använder sig av är: index.html, techniques.html, list.html, project.html, 404.html, 500.html, layout.html och error_layout.html. Varav layout.html och error_layout.html används enbart vid ärvning av basal HTML-struktur. Presentationslagret använder sig av API-funktioner som finns sparade i filen “data.py”. För mer information om samtliga API-funktioner se här:

*API dokumentation*




Figur 1

## 2.1 Kravspecifikation och motivering

Samtliga funktioner i flask_server.py är i enlighet med kursens kravspecifikation, för mer information om kravspecifikationen se här:

*Kravspecifikation*

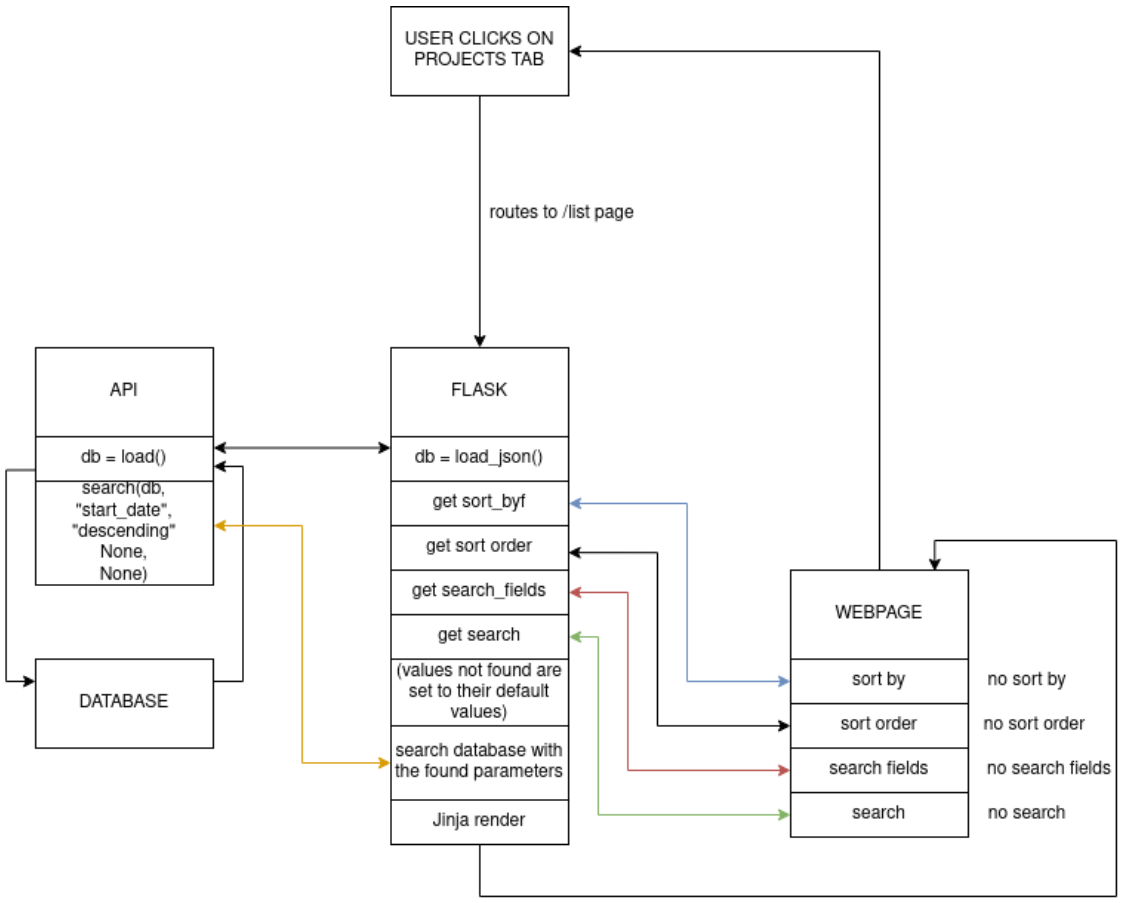
Tabell 1 nedan motiverar funktionerna utifrån kravspecifikationen:

Tabell 1

| Krav_ID | Funktion | Motivering |
|---|---|---|
| 1.1 | index() | Det finns en profilbild och index.html visas. |
| 1.2, 1.5, 2.4 | list() | list.html visas som listar projekt med bild och info. Kan söka bland dem och sortera. |
| 1.3, 1.5 | project(id) | project.html visas och presenterar stor beskrivning, stor bild och annan info om projektet. |
| 1.4, 1.5 | techniques() | listar olika projekt med liten bild och info, beror på användarens val av tekniker. |
| 1.6, 1.7 | page_not_found(e), internal_server_error(e) | visar 404.html respektive 505.html som har felmeddelanden. |
| 2.1, 2.2, 2.3, 2.7, 2.8 | data.json | Varje projekt i filen kan hantera dessa fält. Varje projekt har ett unikt ID. UTF-8 teckenkodning gäller. |
| 2.9 | load_json() | load_json() kallar på API-funktionen load() och hämtar alla projekt när det behövs t.ex. vid url:en för /list och /techniques, vilket innebär att ingen omstart av servern behövs. |

## 2.2 Användarscenario

Figur 2 nedan visar händelseförloppet i kronologisk ordning för ett användarexempel, exemplet handlar om när användaren trycker på länken till url:en /list. Detta händelseförlopp inträffar även när användaren söker utan att ändra något fält (sort by, search fields, sort order, search).




Figur 2

# 3 Dokumentation

Nedan finns dokumentation för flask_server.py:

*Dokumentation*

# 4 Felhantering

Eftersom det är en elementär hemsida så bestämdes att det enbart ska vara fokus på felkoderna “404” och “500”, dessa fångas upp i funktionerna “page_not_found(e)” (404) och “internal_server_error(e)” (500) via flask:s inbyggda funktion “error handler”. En “error handler” används för att fånga upp felkoder och returnerar en egengjord error-hemsida för den specifika felkoden. Till exempel om användaren skulle inmata någon url som inte är route:ad kommer felkoden “404” uppstå och då kallas funktionen “page_not_found(e)” som returnerar “404.html” och visar upp för användaren att sidan inte existerar, liknande gäller för felkoden “500” och dess tillhörande funktion “internal_server_error(e)”. När vi kallar på dessa funktioner kommer datum, tid, felkoden och information om felet att loggas i en textfil kallad “error_log” som finns sparad under katalogen “docs”.

Vid utvecklingen av systemet har test utförts på API-funktionerna med hjälp av gemensamma tester tillgängliga via ett repo på gitlab. Dessa tester fungerar enligt python:s “unittest” ramverk som testar API-funktionerna med en hårdkodad databas och olika specificerade testfall som har ett förväntat utfall. Om någon av testerna skulle misslyckas notifieras utvecklaren om anledningen till detta i terminalen. För mer information om “unittest”, klicka här: *Unittest documentation*