howest
hogeschool

Project Advanced Programming & Maths

Daan De Wile

Inhoud

Groepsleden

1) Daan De Wilde 2MCT4-AIE

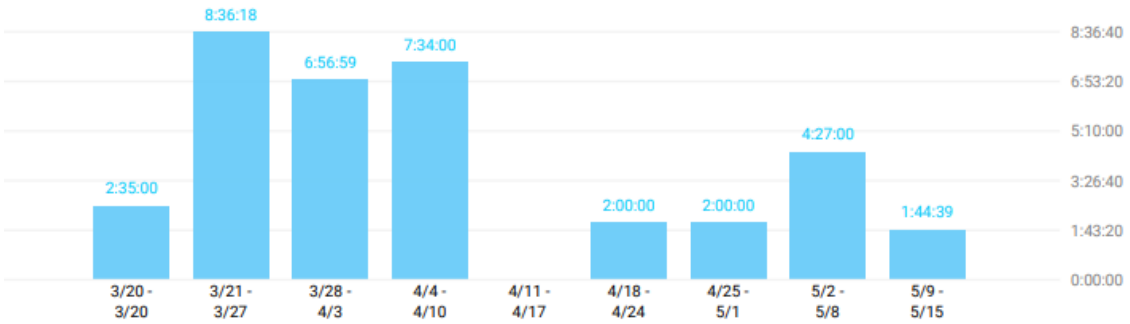
Tijdsindeling (Toggle track report)

Summary Report

toggl track

03/20/2022 – 05/11/2022

TOTAL HOURS: 35:53:56



CLIENT	DURATION
MCT-S4: Project Adv P&M	35:53:56

TIME ENTRY	DURATION
Server	9:34:00
Server gui	6:27:00
Server side logs	4:56:00
Server commands	4:41:59
client window	3:24:15
Zoekopdrachten bepalen en apart testen	2:35:00
Opzetten verslag	1:04:23
Server window	1:02:04
Other time entries	2:09:15

Bespreking zoekopdrachten & dataset

Dataset

Als dataset heb ik gekozen voor tripadvisor_european_restaurants.csv dat bestaat uit ene lijst met alle Europese restaurants. De data bestaat uit hun naam, het land, regio, provincie, stad, adres, coördinaten, prijs level, ratings en andere interessante info.

Zoekopdrachten

1) Bij de eerste zoekopdracht geef je een land en regio mee en krijg je een random restaurant binnen dat land en die regio terug.

```
country = "belgium"
region = "flanders"
df = dataset[dataset['country'].str.contains(country) & dataset['region'].str.contains(region)]
df.iloc[[random.randrange(len(df.index))]]
```

2) Bij de tweede zoekopdracht kan je een speciefiek restaurant op naam zoeken.

```
name = "le saint jouvent"
df = dataset[dataset['restaurant_name'].str.contains(name)]
df
```

3) Bij de derde zoekopdracht kan je een radius meegeven (en jouw eigen coördinaten) en krijg je een random restaurant terug dat binnen die radius van jouw locatie ligt.

```
lat = 51.187528
long = 4.453970
radius = 0.1
restaurants = []
index = 0

for i, r in df.iterrows():
    a = lat - float(r['latitude'])
    b = long - float(r['longitude'])
    c = sqrt(a * a  +  b * b)
    if c<radius:
        restaurants.append(index)
    index = index + 1

df.iloc[[restaurants[random.randrange(len(restaurants))]]]
```

4) Bij de vierde zoekopdracht kan je statistische data opvragen hier krijg je de gemiddelde rating en totaal aantal restaurants per land terug.

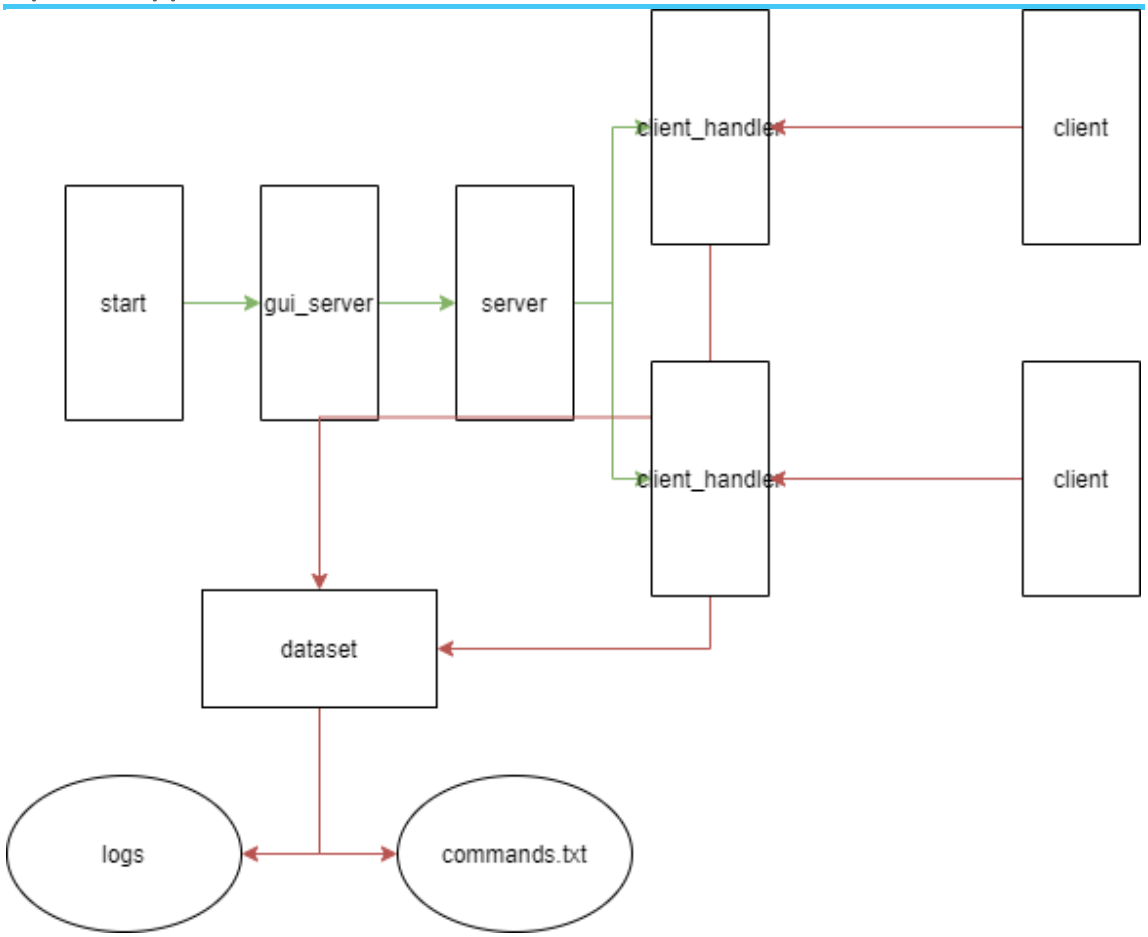
```
countrys = dataset['country'].unique().tolist()
stats = []
for c in countrys:
    country = []
    country.append(c)
    df = dataset[dataset['country'] == c]
    total_restaurants = df['country'].count()
    country.append(total_restaurants)
    avg_rating = df['avg_rating'].mean()
    country.append(round(avg_rating, 2))
    stats.append(country)

print(stats)
```

5) Bij de vijfde zoekopdracht krijg je een grafiek weer over hoeveel restaurents er zijn per provincie van een land dat jijzelf meegeeft.

```
country = "belgium"
df = dataset[dataset['country'].str.contains(country)]
plt.figure(figsize=(10,5))
plt.grid()
plt.xticks(rotation=90)
sns.countplot(data=df, x=df['province']).set(title='Countplot per province')
```

Opbouw applicatie



Bij de start van de server:
- De server wordt gestart door het uitvoeren van start.py.
- Deze zal dataset.py aanspreken en de dataset inlezen.
- Dataset.py zal commands.txt inlezen om bij te houden hoe vaak een request al is gebeurt.
- Deze zal gui_server.py aanspreken en een gui aanmaken.
- Eens daar op de knop ‘start srever’ gedrukt wordt zal deze server.py aanspreken.
- Dit start een server thread op en zal wachten op inkomende clients .
- Eens er een client verbinding maakt zal server.py een clienthandler aanmaken in een nieuwe thread en daar de client ontvangen.

Bij een request van een client:
- De client zal een request sturen naar clienthandler.py.
- Deze zal die opvangen en dataset.py aanspreken en de juiste variabelen meegeven.
- Dataset.py zal de juiste data zoeken en returnen.
- Dataset.py zal de request ook loggen naar een log file per user.
- Dataset.py zal ook bijhouden hoe vaak een request is gebeurt en wegschrijven naar commands.txt.
- Clienthandler.py zal dan de data doorsturen naar client.py.
- Clienthandler.py zal een message in de info queue zetten om weer te geven in gui_server.

Andere relaties:
- Gui_server.py kan aan de logs file om de logs per user weer te geven.
- Gui_server kan aan clienthandler.py vragen welke clients er op dit moment online zijn.
- Gui_server kan aan dataset.py vragen hoe vaak alle requests zijn aangesproken.

Projectopgave VS afgeleverd project

Client side

5 zoekopdrachten.

- 3 met 1 of meer parameters.
- 1 met grafiek.
- 1 met statistische data. ☑

Client moet zich registreren met naam, nickname en email. ☑

Resultaten kunnen weergeven. ☑

Bericht van server kunnen weergeven. ✗

Server side

Belangrijke activiteiten via logvenster. ☑

Overzicht van ingelogde clients. ☑

Gegevens van clients weergeven. ☑

Per client een overzicht van opgevraagde zoekopdrachten. ☑

Bericht aan aangemelde clients kunnen sturen. ✗

Populariteit van zoekopdrachten bekijken. ☑

Server kunnen afsluiten. ☑

Interessante problemen

- Tijdens het maken van de opdracht had ik regelmatig problemen met VSCode die mijn files niet vind.

Aanrader voor volgend jaar:

Library: CustomTkinter

Deze library werkt bovenop Tkinter en de werking is praktisch hetzelfde alleen komt er overal een 'C' voor te staan (Vb: Button → CButton). De bedoeling van deze library is om een volledig nieuwe look te geven aan Tkinter en de gui er weer modern en mooi uit te laten zien.

Helaas is deze nog niet zo lang uit en heb ik deze dus niet in mijn project kunnen implementeren, maar dit wil ook zeggen dat er eventueel nog kleine dingen in zitten die niet werken aangezien er nog aan gewerkt wordt.

howest
hogeschool

howest.be