Høgskulen på Vestlandet

Fakultet for ingeniør- og naturvitskap

Institutt for datateknologi, elektroteknologi og realfag

Applikasjon for brukerstatistikk fra porteføljesystem
Systemdokumentasjon

Versjon <4.0>

Dokumentet er basert på Systemdokumentasjon utarbeidet ved NTNU. Revisjon og tilpasninger til bruk ved IDER, DATA-INF utført av Carsten Gunnar Helgesen, Svein-Ivar Lillehaug og Per Christian Engdal. Dokumentet finnes også i engelsk utgave.

REVISJONSHISTORIE

Dato	Versjon	Beskrivelse	Forfatter
07/05/2023	1.0	Innledning og arkitektur	Anders
19/05/2023	2.0	Prosjektstruktur og databaseløsning	Anders
20/05/2023	3.0	Resterende kapitler	Anders
22/05/2023	4.0	Siste revisjon	Anders

INNHOLDSFORTEGNELSE

1 Innledning

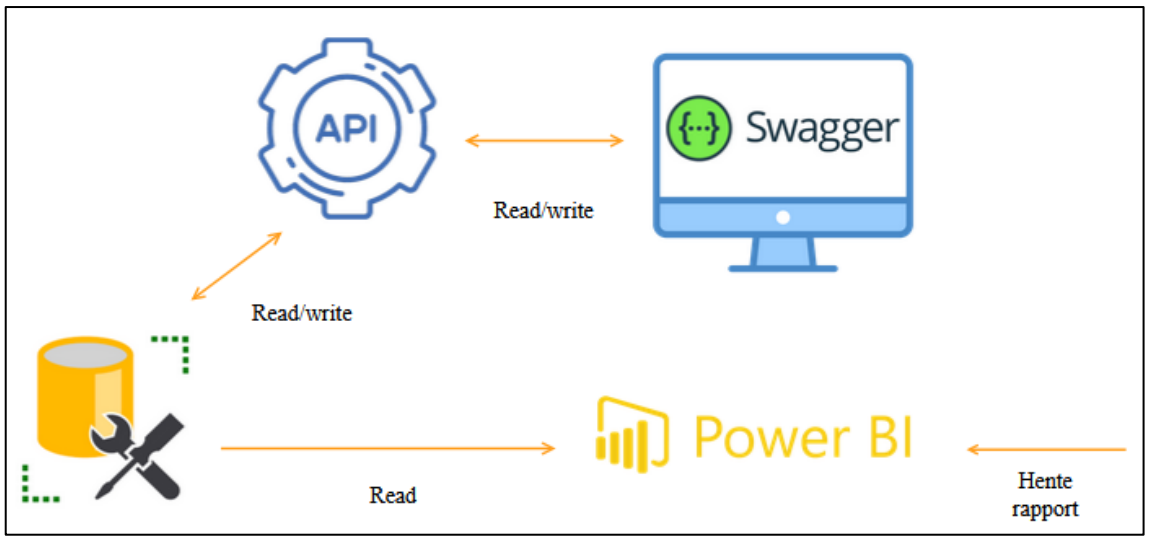
Dette dokumentet tar for seg strukturen til systemet. Først vil dokumentet beskrive arkitekturen og oppbyggingen av systemet. Videre er det oppført en detaljert forklaring på hvordan man installerer, kjører og visualiserer data. Til slutt blir dokumentasjon av kildekode og testing av systemet beskrevet.

2 Arkitektur

Dette kapitelet tar for seg arkitekturen til systemet. Først den overordnete også de ulike delene.

2.1 Overordnet arkitektur

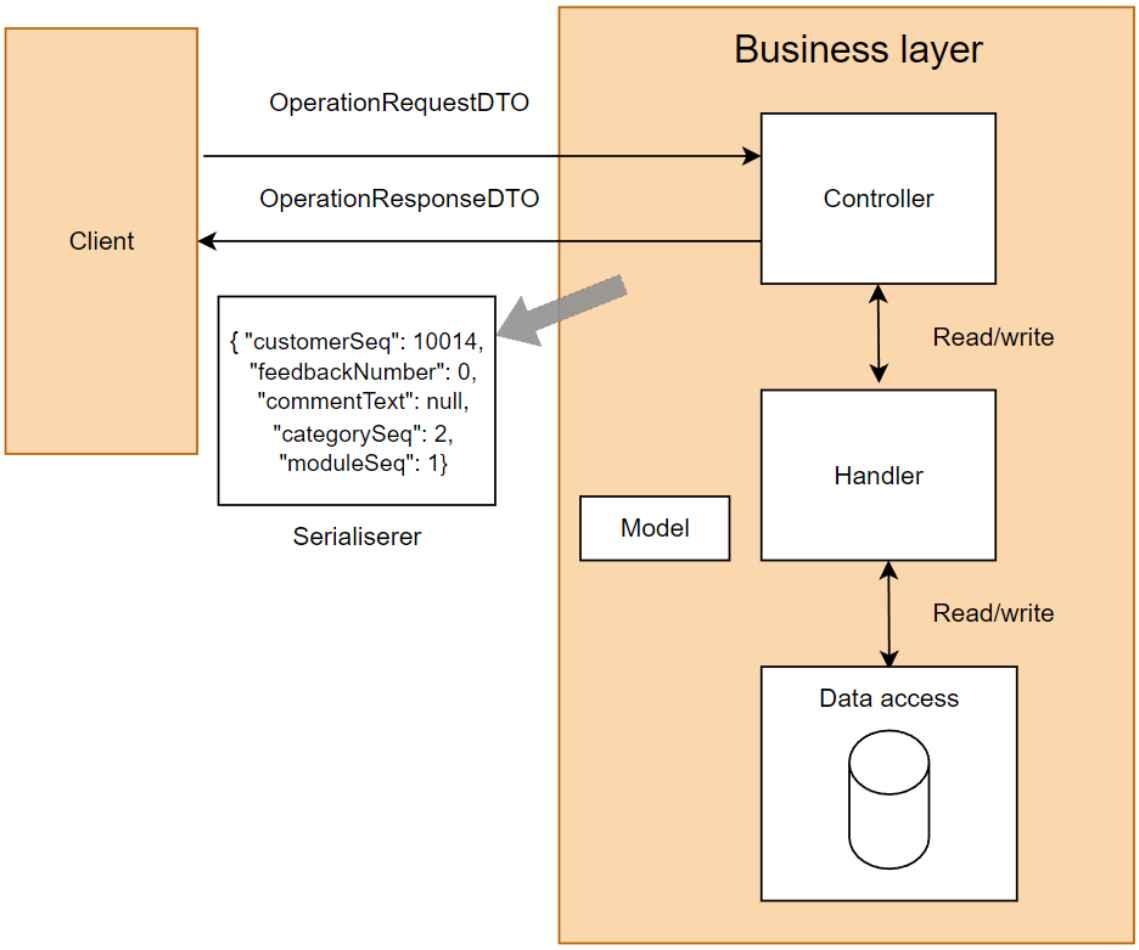
Figur 1 demonstrerer flyten i systemet. Klient sender forespørsel til programvaregrensesnittet og får en respons fra serveren. Dersom et endepunkt i programvaregrensesnittet blir truffet blir data lagret eller hentet fra databasen. Videre er en visning i Microsoft SQL databasen opprettet. Den inneholder all nødvendig informasjon for brukerstatistikk. Visningen blir importert i Power BI og presentert på et dashbord.



Figur 1:Overordnet Arkitektur

2.2 Design av API

Dette kapittelet tar for seg arkitekturen til programvaregrensesnittet (se Fig 2). API-et består av ulike komponenter som har ulike formål. Kontrollerne vil benytte seg av handler-klassene for henting og lagring av data.

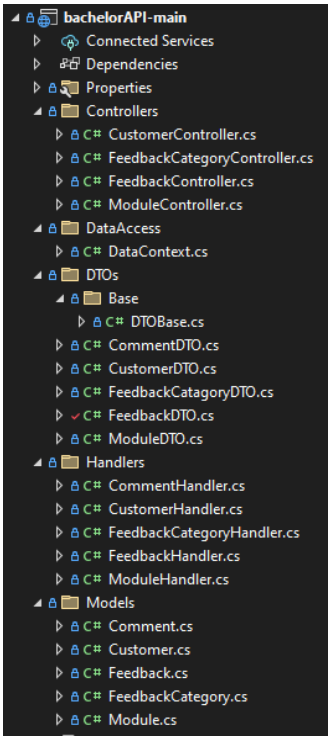


Figur 2: Design av API

3 Prosjektstruktur

Prosjektet er strukturert og delt opp i fem hoveddeler. Hensikten med dette er å gjøre API-et dynamisk. Dette gir mer fleksibilitet ved innføring av ny funksjonalitet.

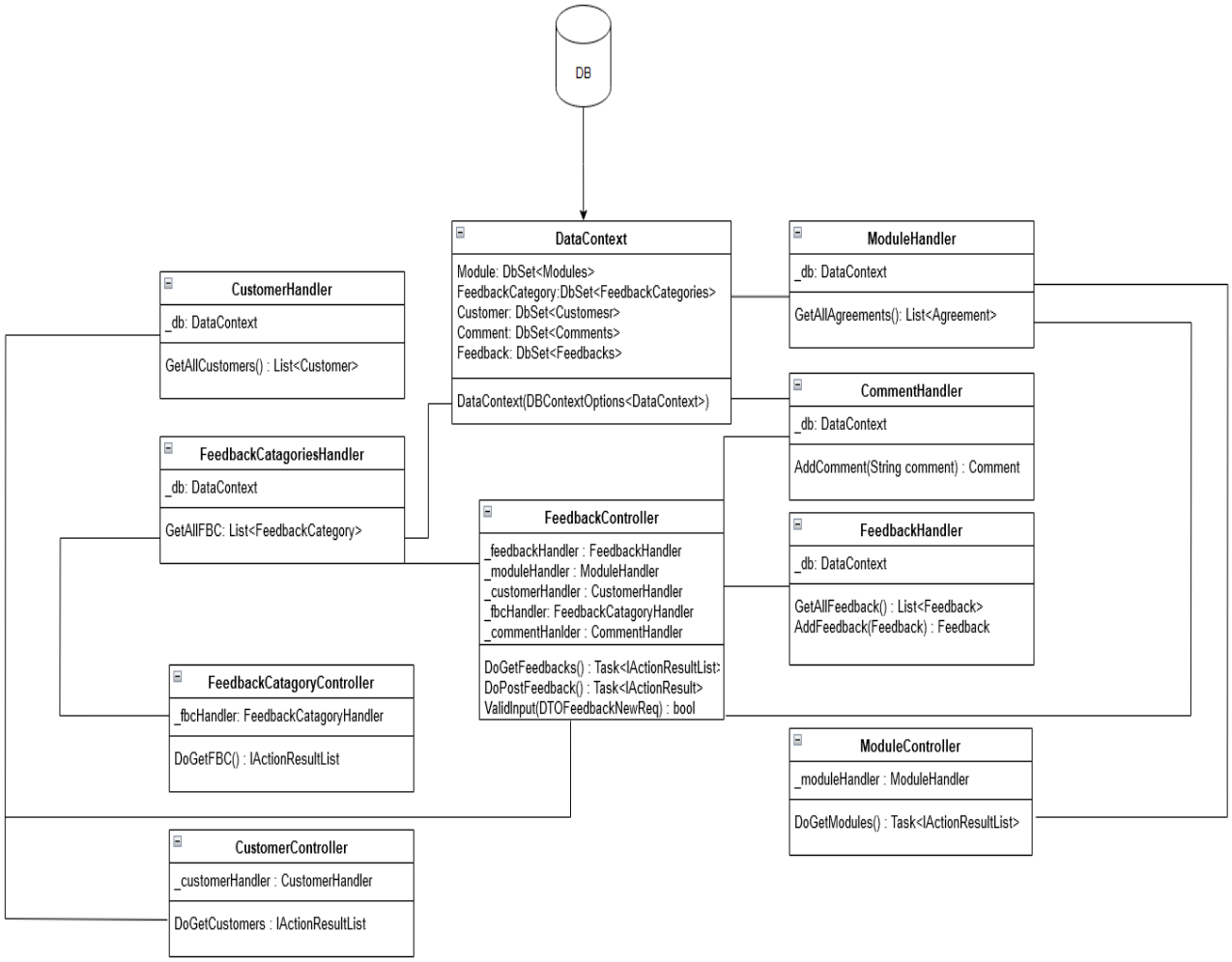
Kontrollerne håndterer forretningslogikken, mens handler-klassene utførere databaselogikken. Modeller og DTO representerer data som skal bli behandlet. DataAccess gir tilgang til Microsoft SQL databasen.



Figur 3: Prosjektstruktur

4 Klassediagram

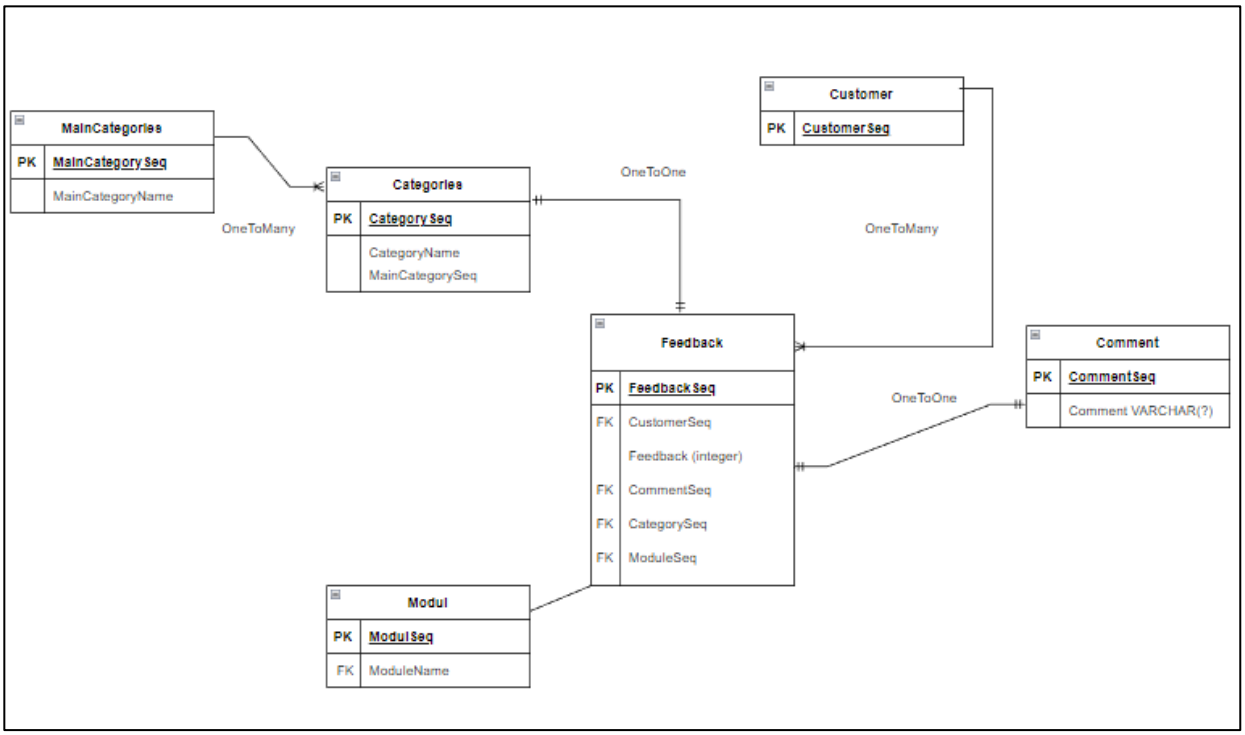
Klassediagram av programvaregrensesnittet.



Figur 4: Klassediagram for API

5 Databasemodell, visning og Power BI

Et at kravene til oppdragsgiver var å legge til entiteter i databasen slik den kunne ta imot tilbakemeldinger. Gruppen valgte å løse dette ved å legge til følgende tabeller i databasen; Categories, MainCategories, Feedback og Comment. Resterende entiteter eksisterte i tabellen fra før av. Databasen som ble tatt i bruk var en eksisterende Microsoft SQL database, eid av Stacc Escali.



Figur 5: Omstrukturering av database

Visning

En visning ble opprette for å strukturer data. Følgende spørring ble brukt.

```
CREATE VIEW [dbo].[vwFeedbackInformation]
AS
SELECT
    cu.CustomerName, mfc.MainFeedbackCategoryName, fc.FeedbackCategoryName, f.FeedbackNumber, m.ModuleName, c.CommentText, f.FeedbackSince
FROM
    dbo.Feedbacks AS f
    INNER JOIN dbo.Comments AS c                    ON c.CommentSeq = f.CommentSeq
    INNER JOIN dbo.Customers AS cu                  ON cu.CustomerSeq = f.CustomerSeq
    INNER JOIN dbo.FeedbackCategories AS fc         ON fc.FeedbackCategorySeq = f.CategorySeq
    INNER JOIN dbo.MainFeedbackCategories AS mfc    ON mfc.MainFeedbackCategorySeq = fc.MainFeedbackCategorySeq
    INNER JOIN dbo.Modules AS m                     ON m.ModuleSeq = f.ModuleSeq
```

Figur 6: Første del av visning

```
UNION

SELECT
    C.CustomerName, '' AS MainFeedbackCategoryName, '' AS FeedbackCategoryName, 0 AS FeedbackNumber,
    ''  AS ModuleName, '' AS CommentText,  CA.CalendarDate AS FeedbackSince
FROM
    Customers C
    CROSS JOIN Calendar CA
WHERE
    CA.CalendarDate <= getdate() AND CA.CalendarDate >= '1/1/2023'

UNION

SELECT
    '' AS CustomerName,  MFC.MainFeedbackCategoryName, FC.FeedbackCategoryName, 0 AS FeedbackNumber,
    ''  AS ModuleName, '' AS CommentText,  CA.CalendarDate AS FeedbackSince
FROM
    MainFeedbackCategories MFC
    INNER JOIN FeedbackCategories FC ON FC.MainFeedbackCategorySeq = MFC.MainFeedbackCategorySeq
    CROSS JOIN Calendar CA
WHERE
    CA.CalendarDate <= getdate() AND CA.CalendarDate >= '1/1/2023'


UNION

SELECT
    '' AS CustomerName,  '' AS MainFeedbackCategoryName, '' AS FeedbackCategoryName, 0 AS FeedbackNumber,
    M.ModuleName, '' AS CommentText,  CA.CalendarDate AS FeedbackSince
FROM
    Modules M
    CROSS JOIN Calendar CA
WHERE
    CA.CalendarDate <= getdate() AND CA.CalendarDate >= '1/1/2023'
```

Figur 7:Andre del av visning

Power BI

Det var nødvendig å gjøre endring på det importerte datasettet. Verktøyet Power Query ble tatt i bruk for dette.

Spørring for å endre navn på kolonnene:

```
=   Table.RenameColumns(dbo_vwFeedbackInformation,{{"CommentText","Kommentar"},
{"CustomerName","Kunde"},{"ModuleName","Modul"},{"MainFeedbackCategoryName",
"Hovedkategori"},{"FeedbackSince","Dato"},{"FeedbackCategoryName",
"Underkategori"},{"FeedbackNumber","Rangering"}})
```

Spørring for å gjøre verdiene i rangerings-kolonnen én større:

```
= Table.TransformColumns(#"Endrer navn på kolonnen", {{"Rangering", each _ + 1,
type number}})
```

Spørring for å opprette en ny kolonne; ErRespons:

```
= Table.AddColumn(#"Rangering + 1", "ErRespons", each if [Kunde] = "" or [Modul]
= "" then 0 else 1, type number)
```

6 Server-tjenester

Serversiden er et programvaregrensesnitt og blir implementert som et REST API. Dette er et API som må følge et visst sett med regler (IBM, u. å). Reglene er at klient og server skal være uavhengig av hverandre, hvert endepunkt har en unik URI og det skal være en lagdelt arkitektur. Ressursene skal kunne mellomlagres på klientsiden, og API-et skal være tilstandsløs.

FeedbackController

/api-docs/Feedback [POST]

Dette endepunktet oppretter en tilbakemelding. Klient sender en forespørsel til endepunkt med et objekt som inneholder alle nødvendige egenskaper til en tilbakemelding. Objektet blir deretter lagret i databasen.

/api-docs/Feedback [GET]

Endepunktet henter tilbakemeldinger ut fra databasen. Tilbakemeldingene blir mellomlagret på klientsiden.

CustomerController

/api-docs/Customer [GET]

Endepunktet hente kunder ut fra databasen. Objektet blir mellomlagret på klientsiden.

ModuleController

/api-docs/Module [GET]

Endepunktet hente aktive moduler ut fra databasen. Objektet blir mellomlagret på klientsiden.

FeedbackCategoryController

/api-docs/FeedbackCategory [GET]

Endepunktet hente underkategorier ut fra databasen. Objektet blir mellomlagret på klientsiden.

7 INSTALLASJON OG KJØRING

For å kjøre selve systemet må man ha .Net installert på egen maskin. Det er også nødvendig med Windows som operativsystem for å redigere rapporten.

Kjøring av API

1. Last ned prosjektet fra gruppens GitHub arkiv.
2. Naviger til mappen som heter bachelorAPI-main.
3. Kjør kommandoen dotnet run.
4. Klient kjører på localhoast 7120
5. Naviger til endepunktet /api-docs

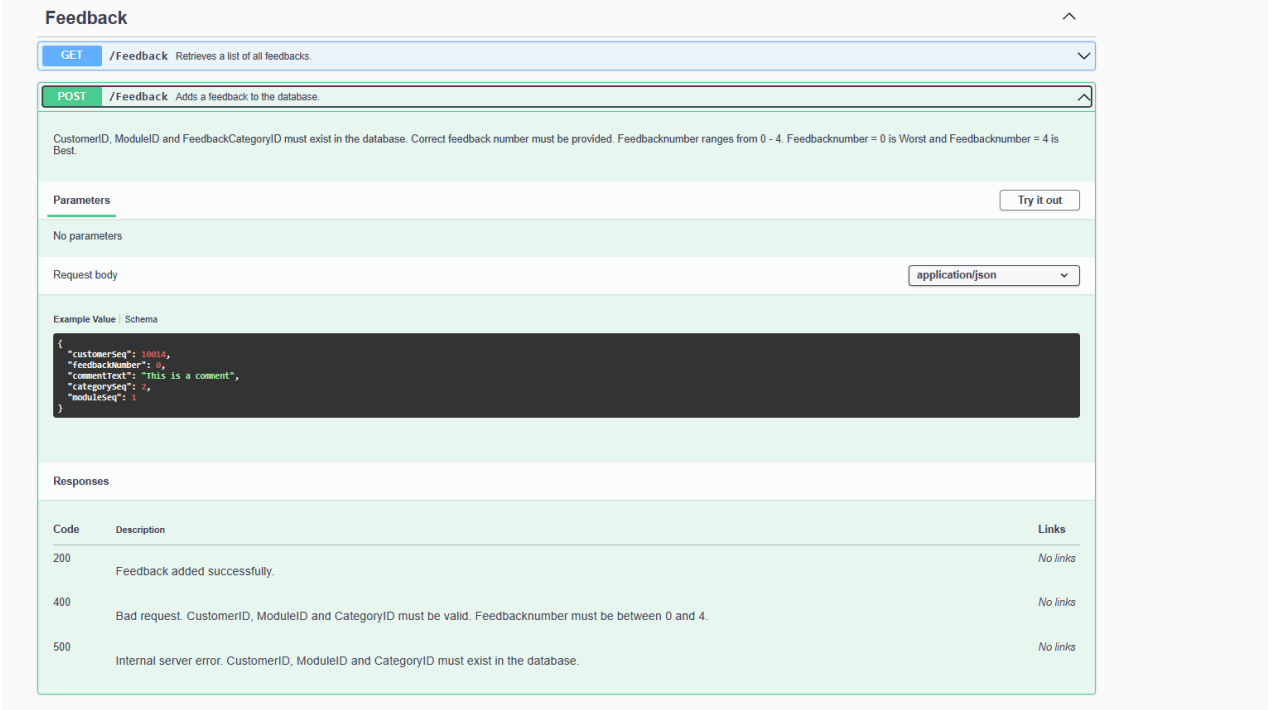
Navigering til dashbord for brukerstaistikk

1. Gå inn på Power BIs web-applikasjon
2. Logg inn med Microsoft-konto
3. Få tilgang til Power BI dashbordet.
4. Oppdatere underliggende datakilder
5. Åpne i visningsmodus
6. Alternativt kan det lastes ned for å redigere dashbordet

8 DOKUMENTASJON AV KILDEKODE

8.1 Dokumentasjon av API

Programvaregrensesnittet blir dokumentert ved å bruke Swagger. Dokumentasjonen er visualisert på en oversiktlig og god måte.



Figur 8: Dokumentasjon av API

8.2 Dokumentasjon av kildekode

I Visual Studio blir XML kommentarer brukt for dokumentering av kode. Dokumenteringen gir en beskrivelse av metoden, lister opp parameterer og returverdi.

```
/// <summary>
/// Adds a feedback to the database.
/// </summary>
/// <remarks>
///  CustomerID, ModuleID and FeedbackCategoryID must exist in the database. Correct feedback number must be provided.
///  Feedbacknumber ranges from 0 - 4.
///  Feedbacknumber = 0 is Worst and Feedbacknumber = 4 is Best.
/// </remarks>
/// <param name="feedback"></param>
/// <returns>A newly created feedback</returns>
/// <response code="200">Feedback added successfully.</response>
/// <response code="400">Bad request. CustomerID, ModuleID and CategoryID must be valid. Feedbacknumber must be between 0 and 4.</response>
/// <response code="500">Internal server error. CustomerID, ModuleID and CategoryID must exist in the database.</response>

[HttpPost]
3 references
public async Task<IActionResult> DoPostFeedback([FromBody] DTOFeedbackNewRequest feedback)
{
```

Figur 9: Dokumentasjon av kildekode

9 TESTING

9.1 Enhetstesting

Enhetstest er en isolert del av koden der man sammenligner faktiske verdier opp mot forventet verdier (SmartBear, 2022). I C# er en enhet en klasse og enhetstester består av en eller flere metoder fra klassen.

Controller

Controller-laget tester de ulike responskodene til metodene, slik at alt fungerer som tiltenkt.

Handler

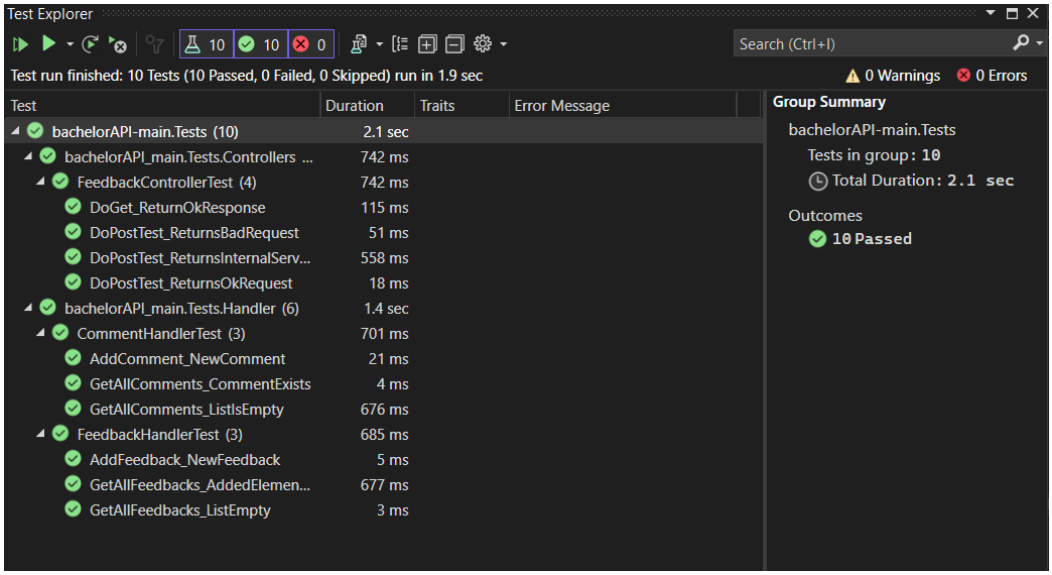
Handler klassene har ansvar i å lagre og hente data fra databasen. Testene er bygd opp med en hovedminnedatabase. Dermed er det mulig å teste om handler-klassen lagrer data.

TestUtils

Hjelpeklasse som legger test-data inn i hovedminnedatabasen.

9.2 Kjøring av tester

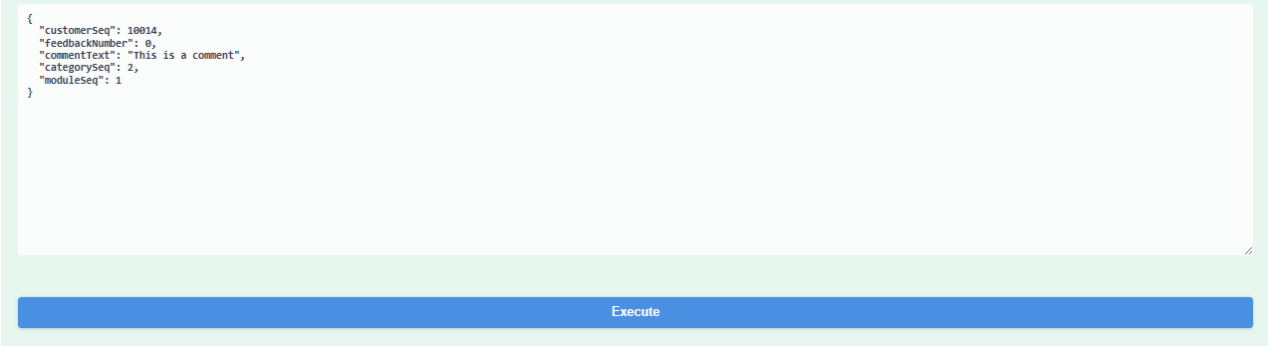
For å kjøre testene må man navigere til test-mappen. Ved å høyreklikke på mappen kan man velge «run tests». Da kjører alle testene.



Figur 10: Enhetstester

9.3 Manuell testing

For å teste ressursene til API-et ble testverktøyet Swagger tatt i bruk for å simulere en klient.



Figur 11: JSON tilbakemelding som blir sendt til API

10 REFERANSER

Tom. D, Chris. R and Stephan. H (2023*). Kestral web server in ASP.NET Core*. Tilgjengelig fra: https://learn.microsoft.com/en-us/aspnet/core/fundamentals/servers/kestrel?view=aspnetcore-7.0

SmartBear (2022*) What Is Unit Testing?* Tilgjengelig fra: https://smartbear.com/learn/automated-testing/what-is-unit-testing/ (Hentet 20. mai 2023).

IBM (u. å) *What is a REST API?* Tilgjengeli fra: https://www.ibm.com/topics/rest-apis (Hentet 21.mai.2023)