Høgskulen på Vestlandet

Fakultet for ingeniør- og naturvitskap

Institutt for datateknologi, elektroteknologi og realfag

Design og utvikling av en testplattform for utviklerkandidater

Systemdokumentasjon

Versjon <1.2>

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

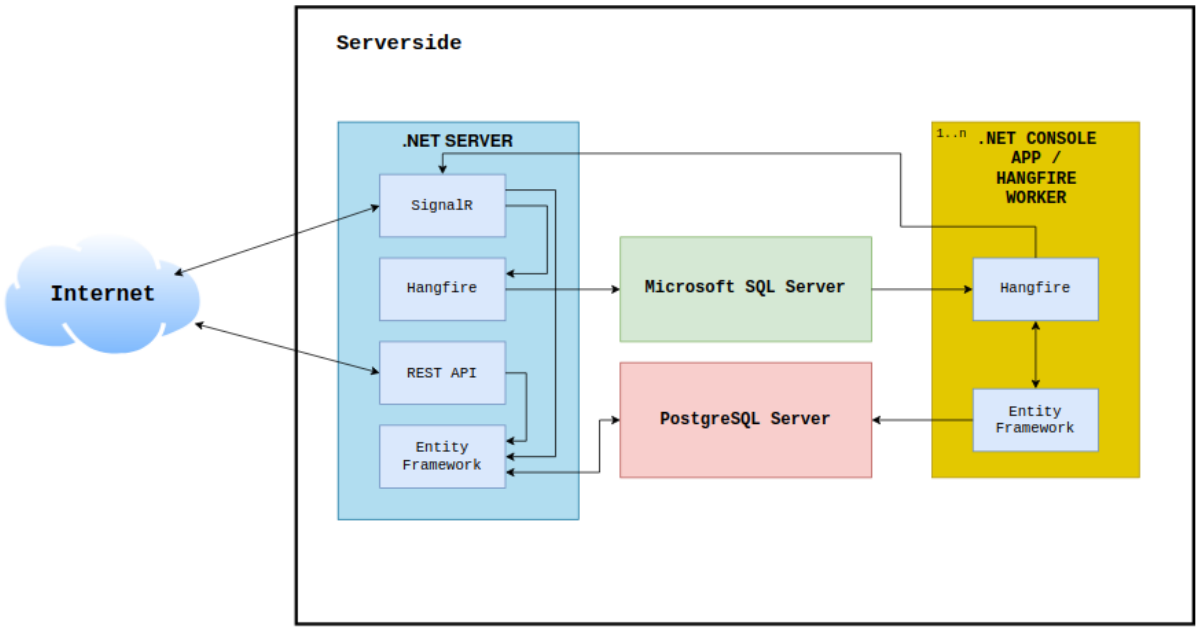
Dato	Versjon	Beskrivelse	Forfatter
<26/02/2023>	<1.0>	Første iterasjon	Anders-Marius Gjerdalen, Anine Hammersborg, Birk Johannessen
<07/05/2023>	<1.1>	Andre iterasjon	Anders-Marius Gjerdalen, Anine Hammersborg, Birk Johannessen
<16/05/2023>	<1.2>	Tredje iterasjon	Anders-Marius Gjerdalen, Anine Hammersborg, Birk Johannessen

INNHOLDSFORTEGNELSE

1 INNLEDNING

Formålet med systemdokumentasjonen er å gi en omfattende forståelse av design og arkitektur for webapplikasjonen som utvikles i dette bachelorprosjektet. Dokumentasjonen vil inneholde beskrivelser av sentrale elementer, som arkitektur, struktur, sikkerhet, samt oppsett og kjøring. Dette vil gi et godt grunnlag for å forstå hvordan systemet fungerer og hvordan det kan videreutvikles på en effektiv måte.

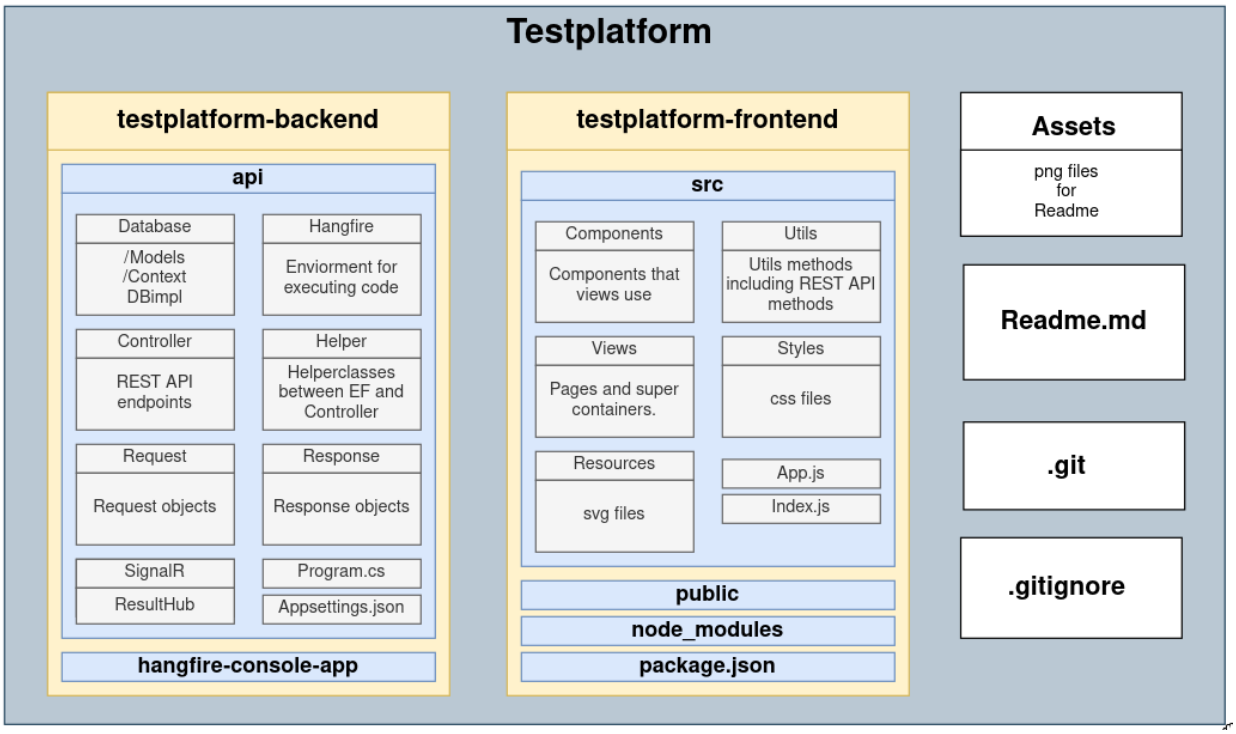
2 Arkitektur



Figur 2-1: Serversiden

Figur 1 viser en systemskisse over serversiden med tilhørende servere og rammeverk. Relasjonen mellom de ulike komponentene på serversiden er modellert. Skissen er laget for å enklere se sammenhengen mellom komponentene på serversiden. Dette blir nøye forklart i hovedrapporten.

3 PROSJEKTSTRUKTUR



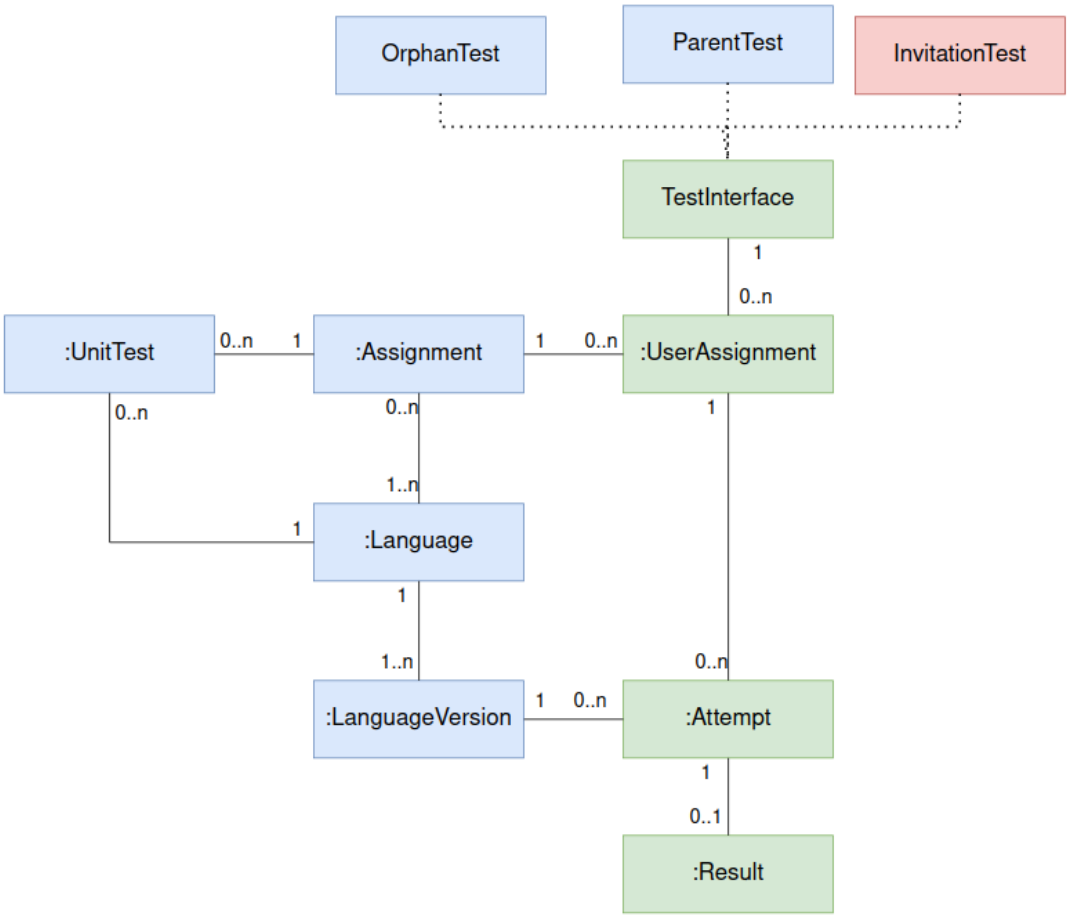
Figur 3-1: Mappestruktur i prosjektet

Figur 2 presenterer mappestrukturen til webapplikasjonen. Prosjektgruppen kom tidlig til enighet om å benytte presise og konkrete navn på mapper og filer. Samtlige filnavn er på engelsk, noe som er standardisert i programvareutvikling. Men det er forskjellig casing til forskjellige standarder. All mappestruktur som ikke går under react og .net er kebab casing (lower) med unntak av autogenererte filer som Readme. Javascipt og C# filer er Pascal case og mappes

Å benytte engelsk språk i kodebasen har blitt en salgs «standard» i programvareutvikling, og det har flere fordeler. For det første er det et universelt språk som er forstått og brukt globalt av utviklere og brukere over hele verden. Dette kan være spesielt gunstig i teamprosjekter, hvor deltakere kan komme fra forskjellige deler av verden og ha forskjellige morsmål. Videre er det enklere å søke etter dokumentasjon, feilsøke eller samarbeide med andre utviklere når kodebasen har et enhetlig språk. Siden engelsk er det vanligste språket for teknisk dokumentasjon, er det også enklere å finne hjelp og støtte på engelsk.

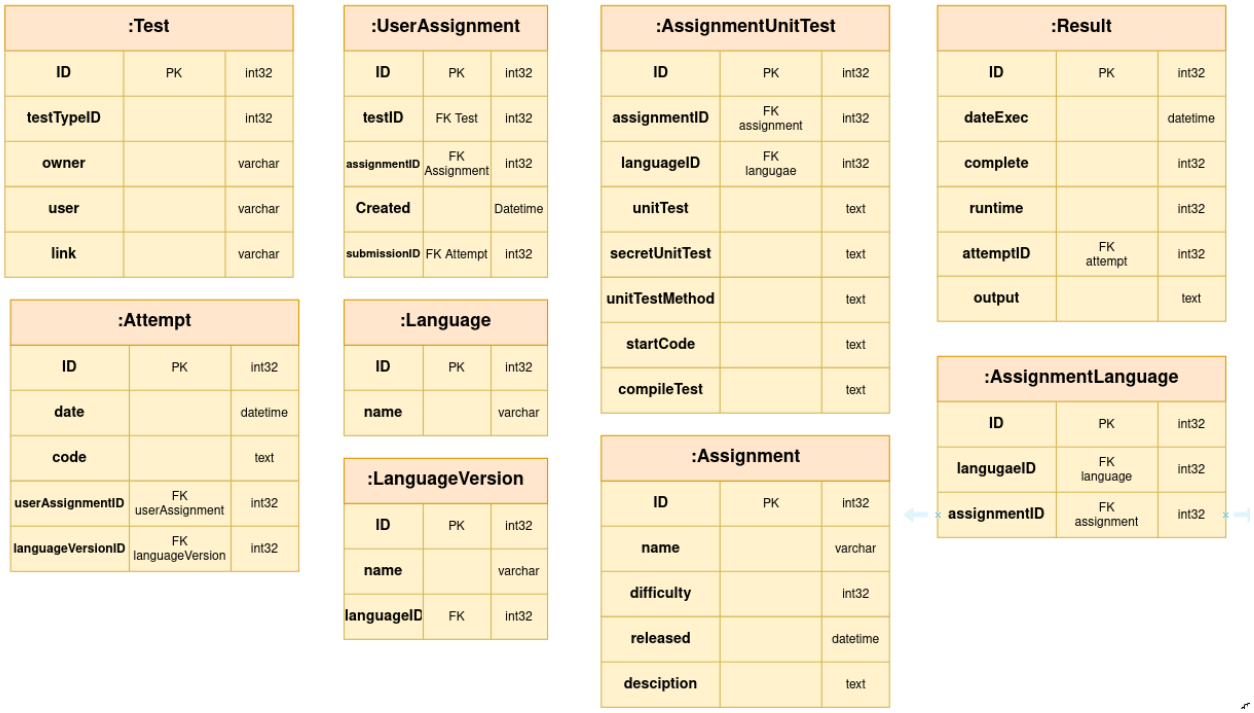
Dette vil også være til fordel for WA når de skal overta koden, videreutvikle produktet og integrere det på deres plattform. Dermed kan de enkelt forstå mappestrukturen til prosjektet.

4 KLASSEDIAGRAM



Figur 4-1: Klassediagram med initielle kravene

Figur 3 viser det første prosjektprosjektgruppen skisserte av et klassediagram basert på de initielle kravene til oppdragsgiver. Dette blir brukt i utformingen av testplattformen.

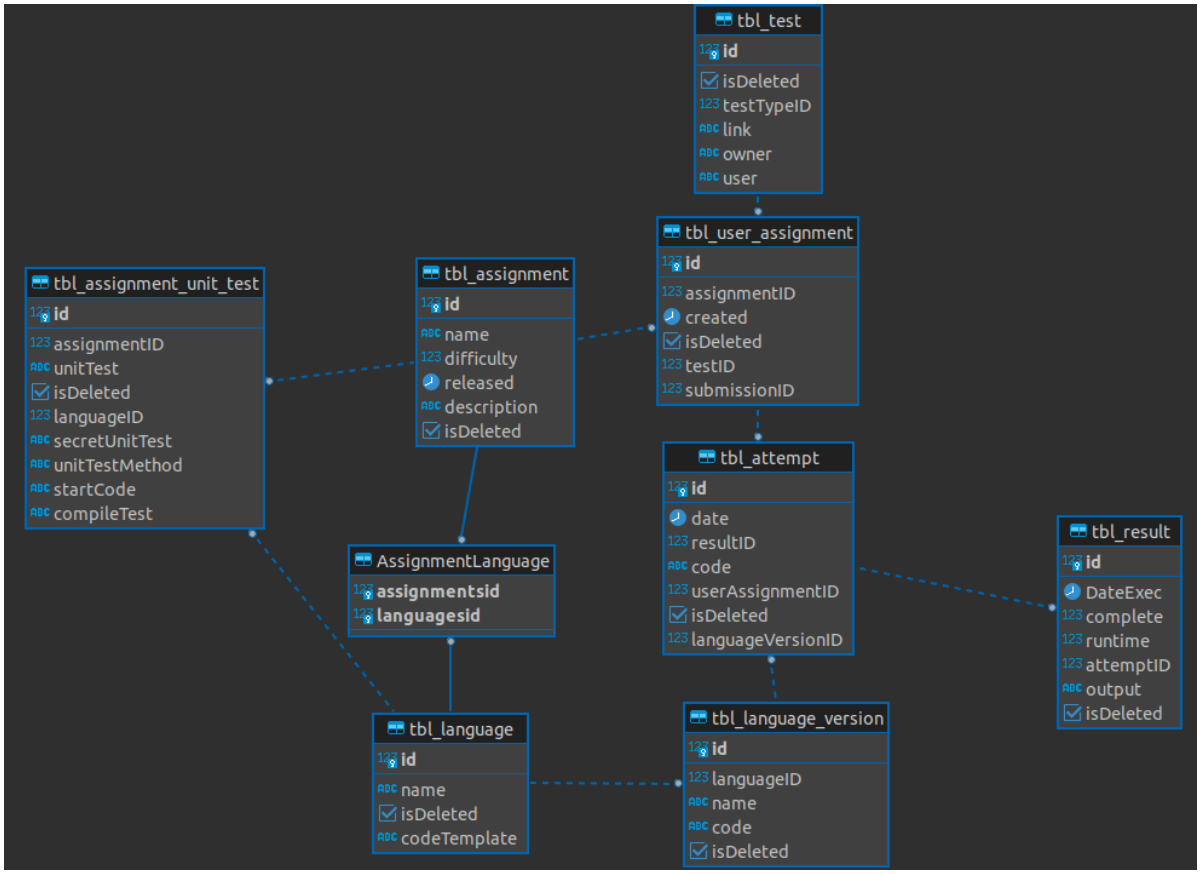


Figur 4-2: Detaljer i klassene

Videre er det framstilt i figur 4 en mer detaljert oversikt over innholdet objektene i klassediagrammet.

5 DATABASEMODELL

Prosjektet brukte Enitity Framework(Microsoft, 2021) til å omforme klassediagrammer til database, så behovet for videre modellering var unødvendig. Derimot er det inkludert et ER-diagram fra DBeaver. av har hentet data fra den faktiske databasen



Figur 5-1: Autogenerert ER-diagram fra DBeaver

6 Server-tjenester

Det ble opprettet totalt 30 REST-ressurser, her er en kort beskrivelse av dem:

- getAllAssignments(): en metode som henter alle oppgaver.
- newAssignment(AssignmentRequest assignment): en metode som oppretter en ny oppgave ved å bruke en HTTP POST-forespørsel med et "AssignmentRequest" objekt som inneholder informasjon om den nye oppgaven.
- deleteAssignment(int id): en metode som sletter en eksisterende oppgave ved å bruke en HTTP DELETE-forespørsel med en oppgave-id som parameter.
- updateAssignment(int id, AssignmentRequest assignmentRequest): en metode som oppdaterer en eksisterende oppgave ved å bruke en HTTP PUT-forespørsel med en oppgave-id og et "AssignmentRequest" objekt som inneholder den nye informasjonen om oppgaven.
- getSingleAssignment(int id): en metode som henter informasjon om en enkelt oppgave ved å bruke en HTTP GET-forespørsel med en oppgave-id som parameter.
- newAttempt(), deleteAttempt(), og getSingleAttempt(): Disse er metoder som definerer funksjonaliteten til API-en for å opprette, slette og hente en enkelt prøve.
- languages(), newLanguage(), deleteLanguage() og getSingleLanguage(): Dette er metoder som definerer funksjonaliteten til API-en for å hente, opprette, slette og hente en enkelt språk.
- languageVersions(), newLanguageVersion(), deleteLanguageVersion() og getSingleLanguageVersion(): Disse er metoder som definerer funksjonaliteten til API-en for å hente, opprette, slette og hente enkelt språkversjoner.
- tests(), newTest(), deleteTest() og getSingleTest(): Disse er metoder som definerer funksjonaliteten til API-en for å hente alle tester, opprette en ny test, slette en test, og hente en enkelt test.
- TestRequest: Dette er en klasse som representerer forespørselen om å opprette en ny test.
- unitTests() er en GET-ressurs som returnerer alle enhetstester.
- newUnitTest(AssignmentUnitTest unitTest) er en POST-ressurs som oppretter en ny enhetstest.
- deleteUnitTest(int id) er en DELETE-ressurs som sletter en enhetstest med gitt id.
- unitTest(int id) er en GET-ressurs som returnerer en enkelt enhetstest med gitt id.
- getAllUserAssignments(): Returnerer en liste over alle brukeroppdrag i systemet.

- newUserAssignment(userAssignment): Oppretter et nytt brukeroppdrag i systemet.
- deleteUserAssignment(id): Sletter et eksisterende brukeroppdrag i systemet basert på ID.
- updateUserAssignment(id, userAssignment): Oppdaterer et eksisterende brukeroppdrag i systemet basert på ID og ny informasjon om brukeroppdraget.
- getSingleUserAssignment(id): Henter informasjon om et enkelt brukeroppdrag i systemet basert på ID.

To websocket endepunkter:

- ExecuteTest som tar mot et forsøk og sender til hangfire.
- DistributeResult som kaller opp klienten med resultatet av forsøket.

7 SIKKERHET

Applikasjonen er i fare mot kodeinjeksjon. REST APIet er åpent uten restriksjoner for hvem som har tilgang. I tillegg er informasjonen over HTTP, en ukryptert kanal. Dette alle anbefalt områder for videre arbeid.

8 INSTALLASJON OG KJØRING

8.1 Avhengigheter

MSSQL server - Used by hangfire

POSTGRESQL server - Used by Entity Framework

DOTNET 7.0 SDK+RUNTIME

.NET cli to compile and serve backend.

NPM 18.0

Node package manager, building and serving react.

8.2 Linux setup

Applikasjonen skal integreres mot server, så guiden er bare laget for linux systemer.

Avhengiv av docker for MSSQL: Official Microsoft guide

```
sudo docker pull mcr.microsoft.com/mssql/server:2022-latest
sudo docker run -e "ACCEPT_EULA=Y" -e
"MSSQL_SA_PASSWORD=YourStrong@Passw0rd" \
   -p 1433:1433 --name sql1 --hostname sql1 \
   -d \
   mcr.microsoft.com/mssql/server:2022-latest
```

```
sudo docker ps –a
```

sudo docker start sql1

Så konfigurerer testplatform-backend/api/appsettings.json som er tilkoblingsstrenger til databasen med tilhørende postgresql og msssql database brukernavn og passord.
Så kan brukeren bygge siste migration fra Entity Framework:

```
cd testplatform-backend/api
dotnet ef database update
```

Staret backenden:

```
cd testplatform-backend/api
dotnet build
```

starte frontenden:

```
cd testplatform-frontend
npm i
npm start
```

9 DOKUMENTASJON AV KILDEKODE

Kildekoden ligger vedlagt som ZIP-fil.

10 KONTINUERLIG INTEGRASJON OG TESTING

Systemet er ikke deployert og har ingen kontinuerlig integrasjon, ingen unit tester eller integrasjonstester er skrevet for systemet.

11 REFERANSER

Microsoft. (2021). Enitity Framework Core.
Tilgjengelig fra: https://learn.microsoft.com/en-us/ef/core/
[Hentet 10.05.2023]

12 Figurliste