Høgskulen på Vestlandet

Fakultet for ingeniør- og naturvitskap

Institutt for datateknologi, elektroteknologi og

Tofaktorautentisering (2FA), ved bruk av Tidsbasert engangspassord (TOTP), i MinID.

Two-factor authentication (2FA) using time-based onetime password (TOTP) in MinID.

Systemdokumentasjon

Versjon 1.0

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
15/04	0.1	Databasemodell	U.F.
11/04	0.2	Databasemodell	U.F
11/04	0.3	Sikkerhet	J.V.E
12/04	0.4	Prosjektstruktur	J.V.E
04/05	0.5	Arkitektur	U.F
09/05	0.6	Testing oppdatert	E.F
10/05	0.7	Arkitektur, Prosjektstruktur, Klassediagrammer og Databasemodell oppdatert	U.F
13/05	0.8	Innledning	J.V.E
14/05	0.9	Installasjon, Testing	E.F
16/05	1.0	Innledning, Arkitektur, Installasjon og Kjøring, Sikkerhet, Dokumentasjon av kildekode	J.V.E, U.F, E.F

INNHOLDSFORTEGNELSE

1 Innledning

Dette dokumentet består av systemdokumentasjonen for *Tofaktorautentisering (2FA) ved bruk av Tidsbasert engangspassord (TOTP) i MinID.*

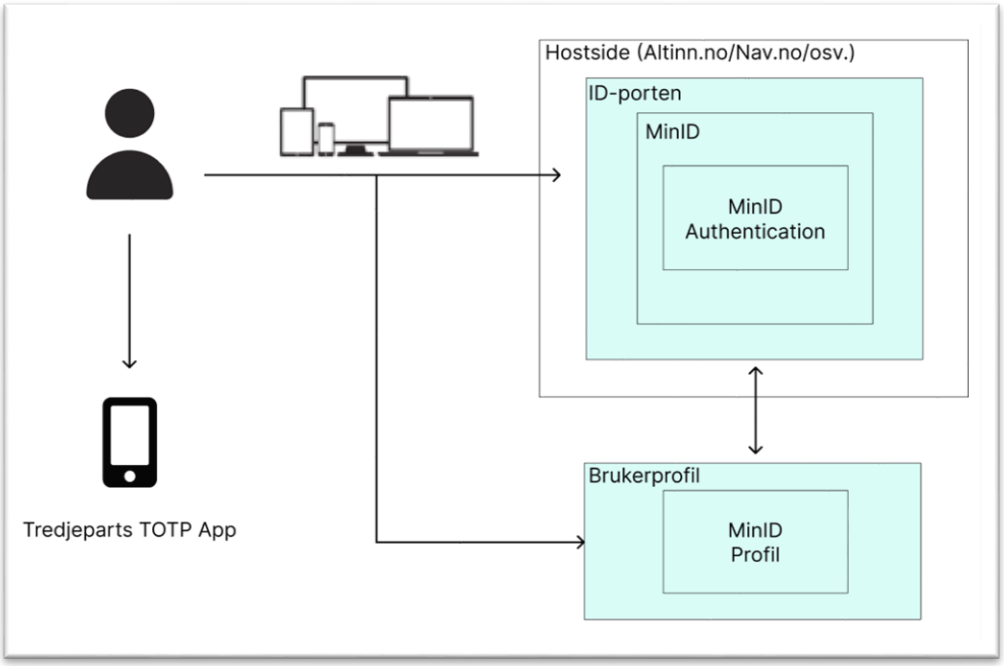
Dokumentet har som hensikt og forklare hvordan systemet ble designet, og hvordan det ble skapt. Det vil også gi innblikk i hvordan en kan vedlikeholde og videreutvikle produktet.

Systemdokumentasjonen er bygd i ulike kapitler:

- Arkitektur som beskriver de viktigste komponentene i systemet.
- Prosjektstruktur som visualiserer hvordan fil- og katalogstrukturen er for systemet.
- Klassestruktur som representerer de viktigste klassene og hvordan de samhandler.
- Databasemodell som viser endringer/nye tabeller i databasen, og hvordan de henger sammen.
- Sikkerhet som forklarer hvordan produktet er sikret mot utilsiktet angrep, hvordan passord blir håndtert og lignende.
- Installasjon og kjøring som forklarer hvordan en kan få produktet opp å kjøre.
- Dokumentasjon av kildekode som omhandler hvordan kildekode er skrevet.
- Testing som forklarer hvordan tester er skrevet og hvorfor de er skrevet slik som de er.

2 Arkitektur

MinID som er en av autentiseringsalternativene ID-porten tilbyr, kan nås ved innlogging hos en hostside. MinID-profil er en separat modul fra MinID-autentisering som har ansvaret for brukerinnstillinger. Systemet er bygd på Model-View-Controller sine arkitekturprinsipper, men med justert navngivning for å imøtekomme det som forventes fra Spring Boot og Thymeleaf.



Figur 1 Konseptuell oversikt

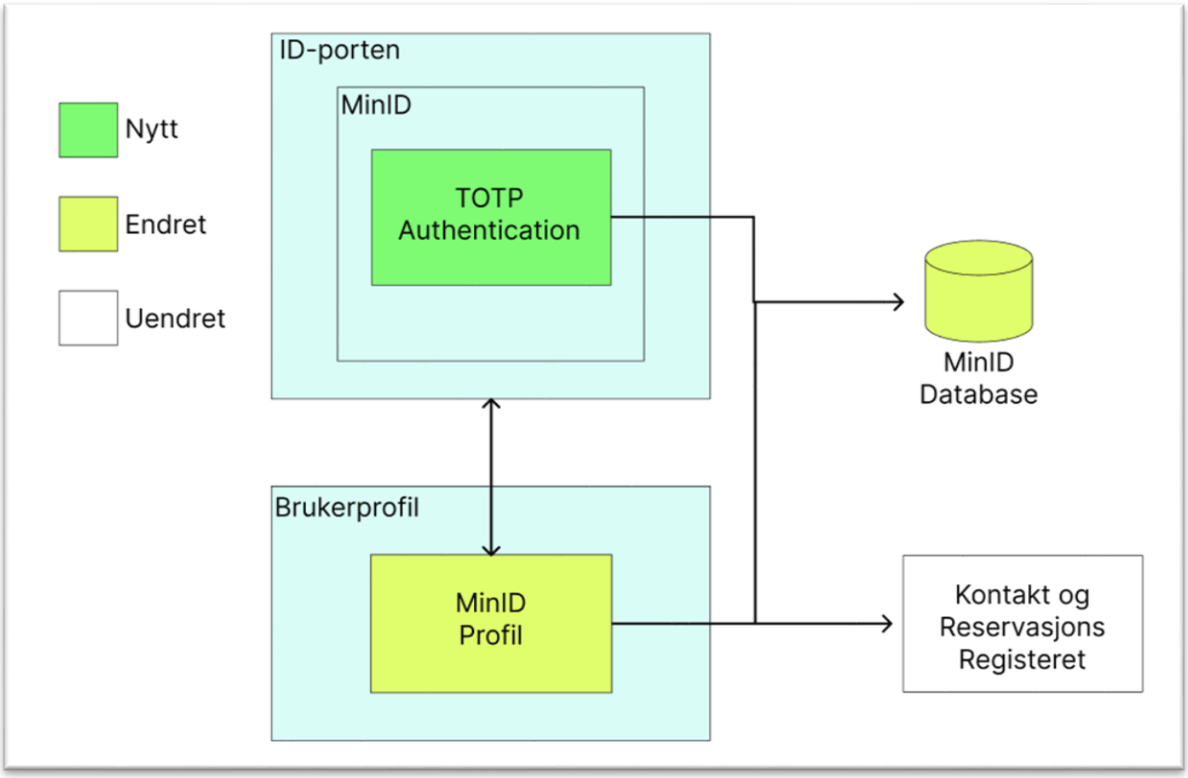
Systemer er oppdelt i to moduler som skilles basert på ansvarsområder (Figur 2):

Brukerprofil presenterer brukerinformasjon, og har ansvaret for å håndtere mulige endringer av brukerinnstillinger.

TOTP Authentication har ansvaret for innloggingsprosessen, med den nye TOTP MFA funksjonaliteten implementert.

Brukerprofil og TOTP Authentication bruker en delt Utils-klasse for håndtering av logikk.

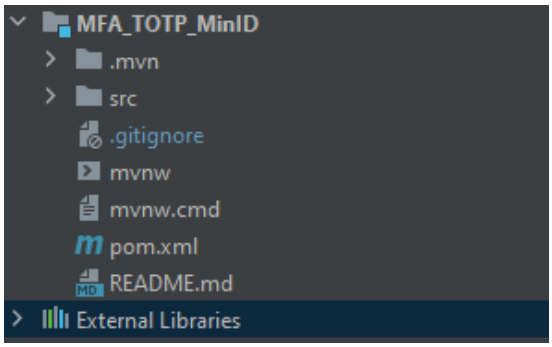
KRR representerer Kontakt og Reservasjonsregisteret sin API og brukes for å hente brukerinformasjon, spesifikt telefonnummer.



Figur 2 Oversikt over systemets moduler

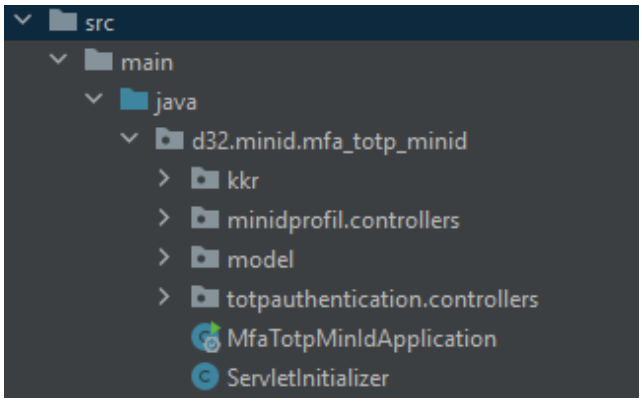
3 Prosjektstruktur

Fil-strukturen for prosjektet er satt opp på denne måten (Figur 3), hvor src er lokasjonen til kildekoden. Alle eksterne biblioteker og avhengigheter er dokumentert i pom.xml filen. Siden dette er et maven prosjekt håndterer maven alle avhengigheter som trengs for å kjøre programmet.



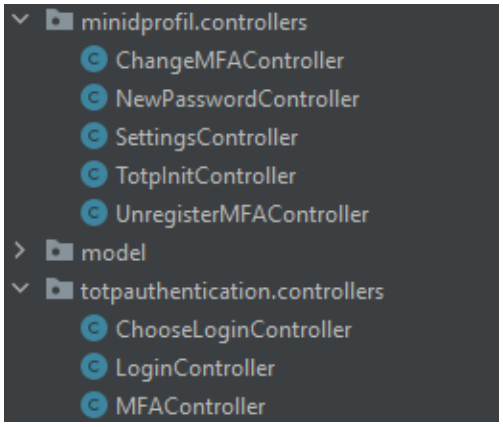
Figur 3 Overordnet struktur

Kildekoden er strukturert ut ifra modulene, med applikasjonen og servlet initialisereren for seg selv.



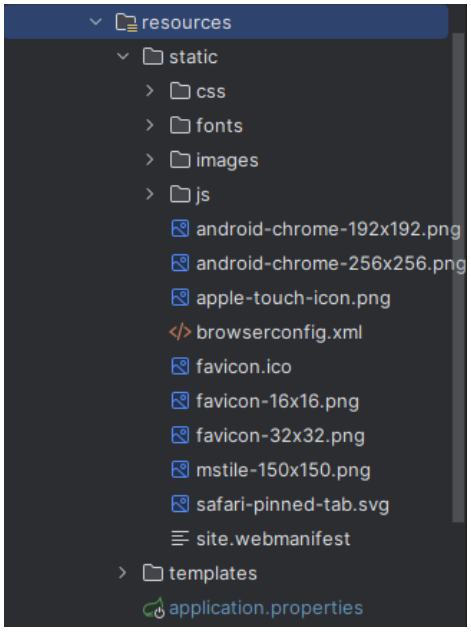
Figur 4 Moduloversikt

For eksempel er kontrollerne delt opp i sine separate moduler og med spesifikke ansvarsområder.



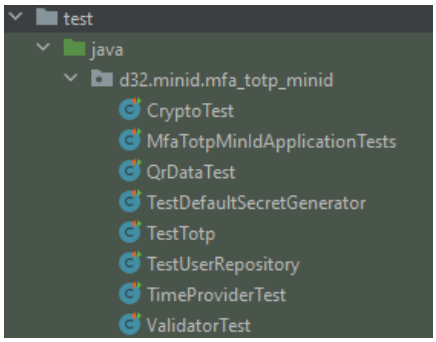
Figur 5 Kontrollere

Under resources ligger alt relatert til frontend, hvor static mappen har alle bilder, fonter, css og JavaScript som websidene trenger. I templates ligger alle html-filer. I resources ligger også filen application.properties, denne filen håndterer de ulike datakildene. Det vil si database url, brukernavn og passord, i tillegg beskrives det som trengs for klientregistrering og provider for ID-porten.



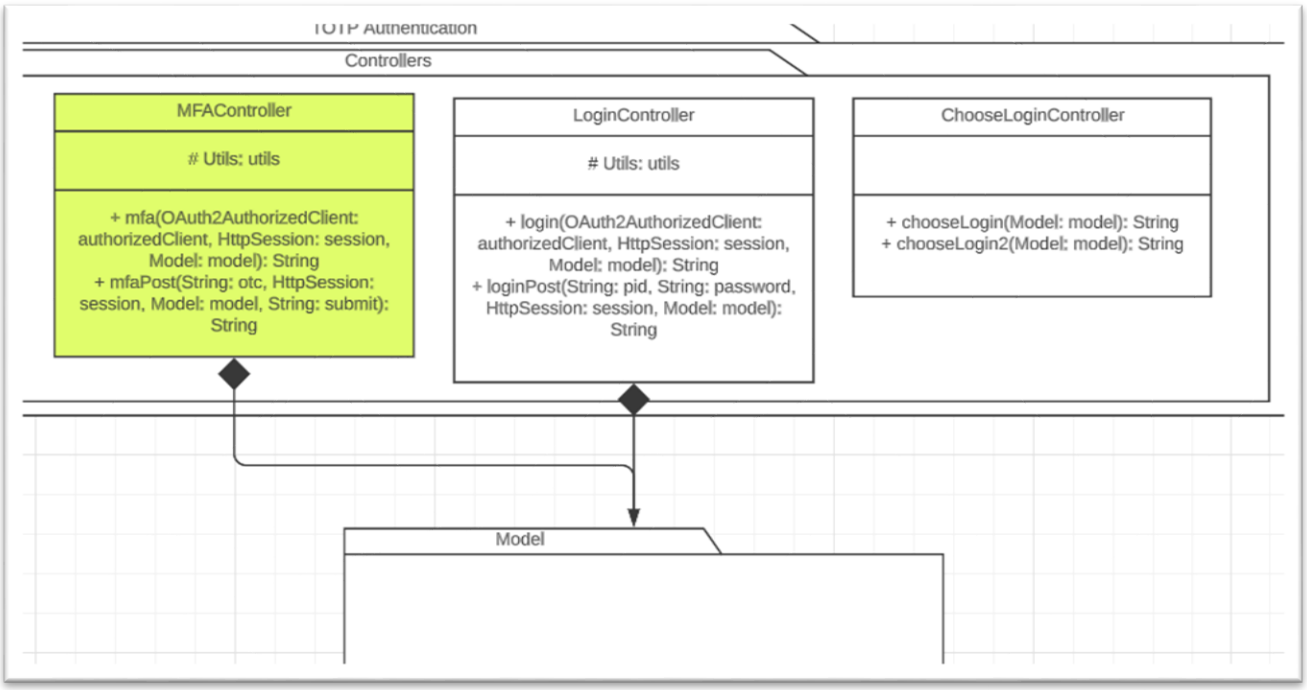
Figur 6 Resources

I test ligger de ulike testene som er skrevet for programmet, mer om dette under kapittel 9: Testing.

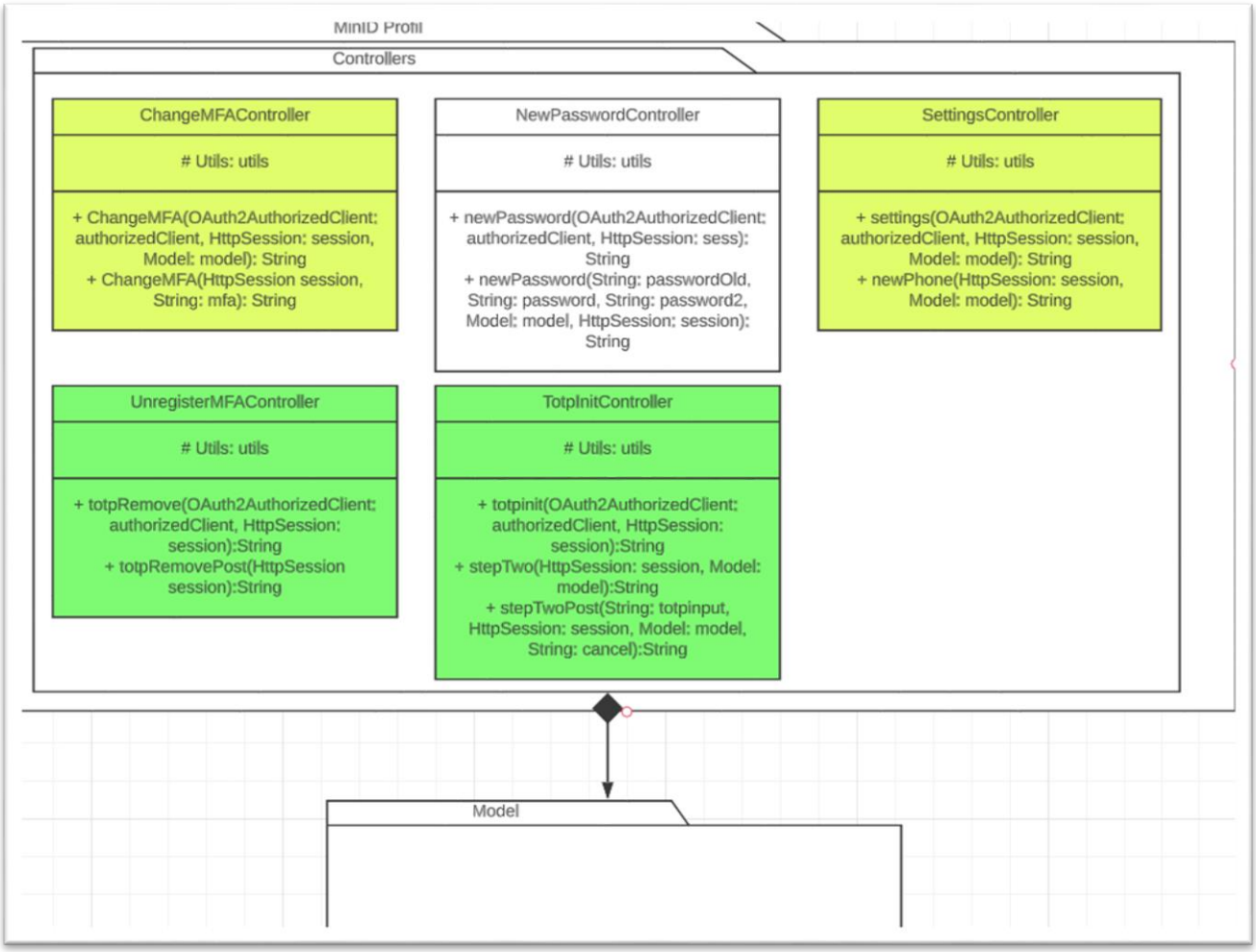


Figur 7 Tester

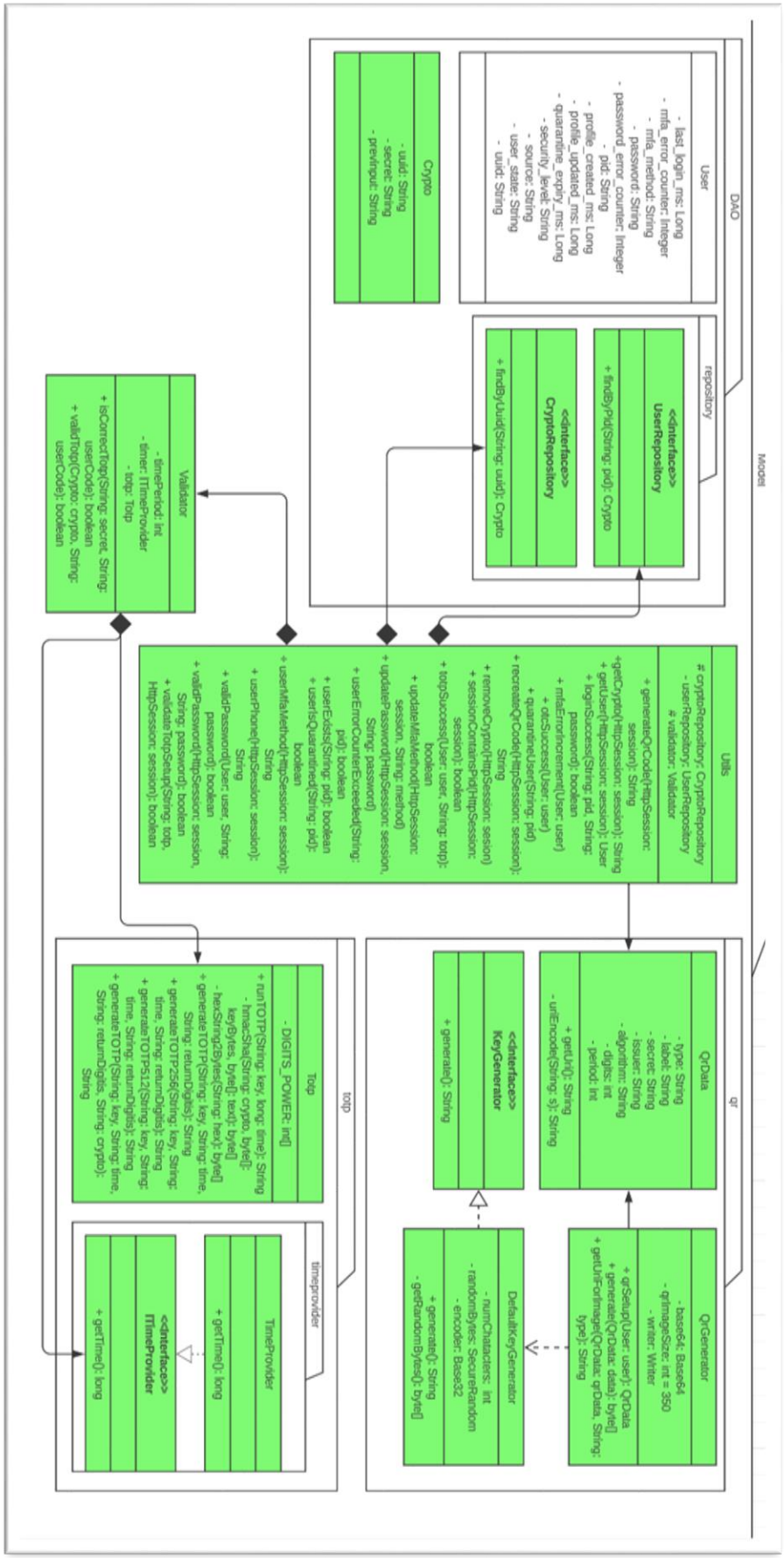
4 Klassediagrammer



Figur 8 TOTP Authentication klasser



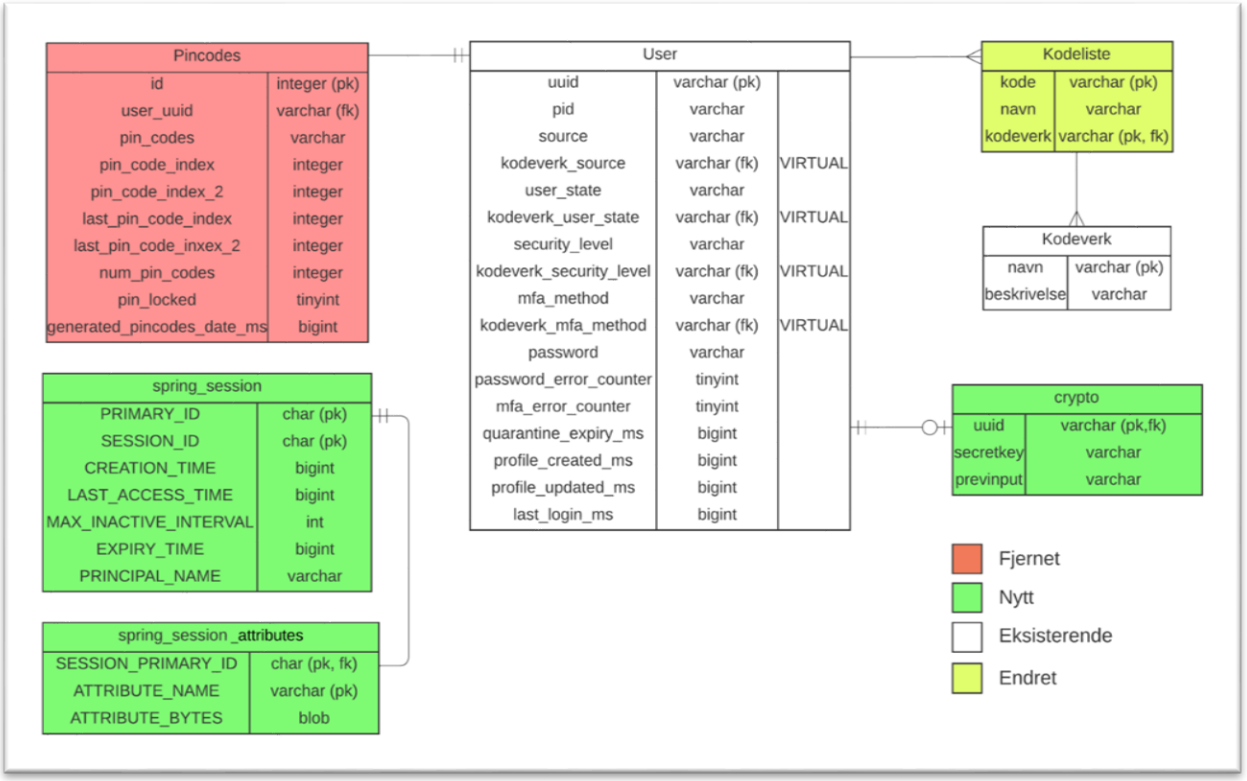
Figur 9 MinID Profil klasser



Figur 10 Klassediagram – Modelinnhold

5 Databasetabell

Da PIN-koder ble avviklet i januar (*Endring I Bruk Av Minid,* 2023) er det ikke inkludert som del av løsningen. User, Kodeliste og Kodeverk fungerer på sammenlignbar måte som DigDirs egen database hvor relevant. Det er f.eks. ikke prioritert å implementere funksjonaliteten bak Kodeliste og Kodeverk da dette ikke er relevant for prosjektet. Endringen i Kodeliste refererer til TOTP som autentiseringsalternativ. Spring_session og spring_session_attributes er automatisk generert av Spring Security, hvor PID blir lagt til som session attribute ved innlogging. Dette gjøres separat fra ID-porten for å kunne gjennomføre fullstendig TOTP implementasjon. Crypto inneholder uuid referanse fra relevant bruker, samt dens krypterte secretkey og den forrige TOTP-koden som ble brukt for autentisering i previnput.



Figur 11 ER-Diagram av databasemodell

6 Sikkerhet

Spring Boot har en del tilleggsfunksjonalitet som kan benyttes, blant annet Spring Security og Spring Session. Når en bruker logger inn, vil det bli laget en access token i spring_session tabellen i databasen. Dette tokenet følger Oauth 2.0 standarden. Det vil i tillegg konstrueres en csrf(Cross Site Request Forgery) token som lagres i spring_session. Spring_session er en tidsbasert tabell, og informasjonen som legges der er derfor gyldig i en gitt tidsperiode. Hvor lenge, er basert på informasjon i application.properties filen. Mye av dette gjøres automatisk av Spring Security, som sammen med Spring Session håndterer mye av de vanlige sikkerhetsutfordringer som finnes.

For å validere brukerinnlogging må brukeren logge inn med et unikt personnummer og passord, for å deretter validere med OTC eller TOTP kode. Det vil da legges til et attributt på spring_session_attribute-tabellen med en referanse til hvilken bruker det er som er autentisert og hvilken spring_session brukeren har.

Brukerens passord er hashet i databasen ved bruk av BCrypt(OWASP, n.d) med en "work factor" på 12, det vil si at det kjøres 2^{12} iterasjoner for å hashe passordet. Brukerens input vil bli hashet av serveren på samme måte for å så valideres opp mot databasen.

For å beskytte mot SQL-injections er det tatt i bruk form-handling som validerer at det som bruker kan sende inn ikke kan brukes til å sende ugyldige forespørsler. Det eneste stedet som det i teorien kunne være mulig å sende slike forespørsler er i passordfelt, men grunnet at passord blir kryptert før det sendes spørring til database, vil ikke SQL-injections være mulig.

7 Installasjon og kjøring

Prosjektet vil ligge i et GitHub-depot for de som har fått tilgang. En komprimert mappe av prosjektet er også med som vedlegg til innleveringen.

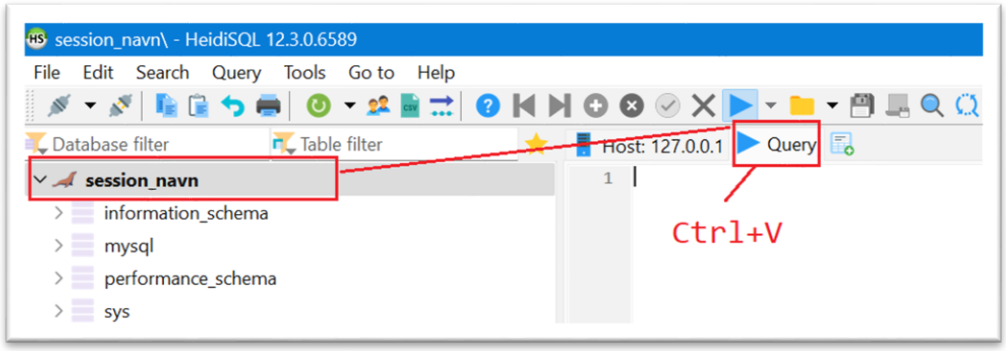
Det anbefales å bruke samme verktøy som prosjektgruppen selv har anvendt. Dette innebærer Intellij som IDE for front- og backend utviklingen, og HeidiSQL for arbeid opp imot database. Instruksene for installasjon og kjøring er skrevet med utgangspunkt i disse verktøyene og tar ikke høyde for uforutsette problemer med andre programvarer.

Database oppsett

Oppsettet av en database gjøres her vha. av HeidiSQL og tar utgangspunkt i at MariaDB er installert og kjører.

- Kopier innholdet i SQL-scipt.txt og kjør en query med det ved å lime inn scriptet i Query-vinduet i HeidiSQL (Figur 12),
- Trykk "Execute sql..." (Blå pil) eller press ned F9, scriptet vil kjøre og generere databasen.
- Det kan være behov for å gjøre en refresh for å se oppdateringene: Høyreklikk session_navn eller tilsvarende > og velg refresh.

Dette skal opprette en database med navn minid, under her skal crypto, kodeliste, kodeverk, spring_session, spring_session_attributes og user ligge.



Figur 12 Bilde av hvor script limes inn og kjøres

Application.properties

Med en database som er oppe og går, kan det nå kobles til i application.properties. Denne filen ligger i "src/main/resources/"

```
spring.datasource.url=jdbc:mariadb://<ip-adresse>/<database navn>
spring.datasource.username=<database brukernavn>
spring.datasource.password=<database passord>
...
spring.security.oauth2.client.registration.idporten.client-id=<client id>
spring.security.oauth2.client.registration.idporten.client-secret=<client secret>
```

Prosjektet nytter seg av en kobling opp imot DigDirs testsystem, dette krever client_id og client_secret tilsendt fra Digdir, og må legges til application.properties.

Maven

I prosjektet ligger en pom.xml fil som er fylt inn med flere avhengigheter prosjektet vil ha til rammeverk og biblioteker. Maven skal håndtere dette automatisk

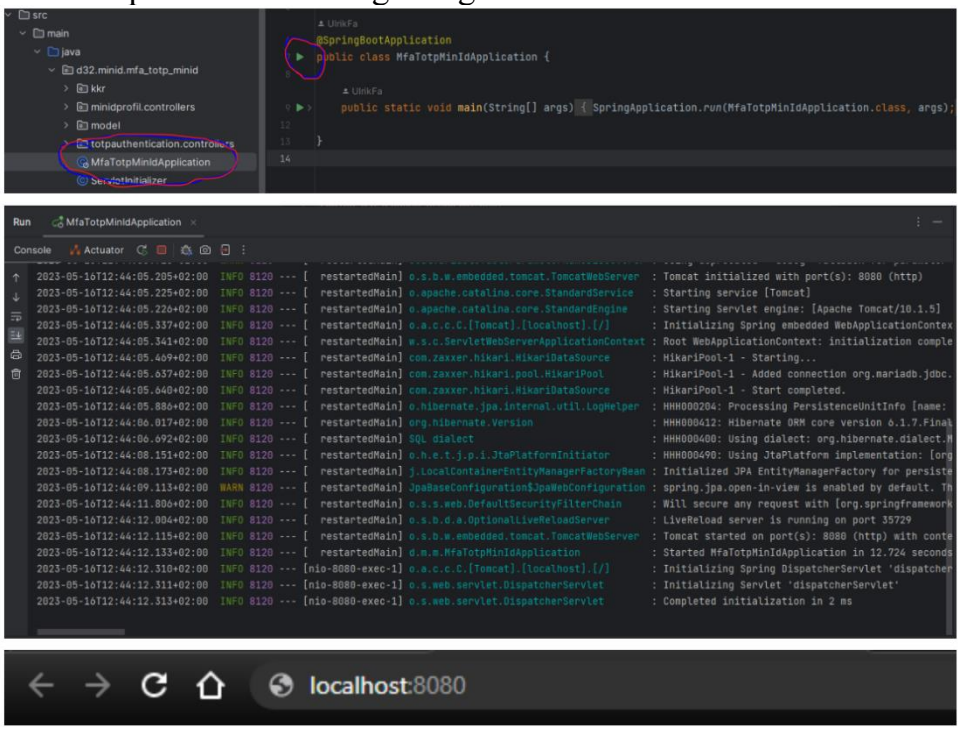
Generer en bruker

En av testene i prosjektet (TestUserRepository) genererer en bruker i databasen om den ikke eksisterer der fra før (Figur 13). Denne brukerkontoen vil kunne la seg nyttes til innlogging når programmet kjører, så det anbefales å kjøre gjennom testsettet før resten av programmet. Kjøring av testsettet vil også validere at maven har håndtert avhengigheter, og validere databasetilkobling.

Figur 13 Brukergenerering via testklasse

Kjøring

Programmet kjøres ved å navigere til MfaTotpMinIdApplication.java og trykke pil markert (figur 14). Terminalen vil da åpnes og vil indikere at programmet kjøres. Bruker må deretter åpne en nettleser og navigere til localhost:8080.



Figur 14 Kjøring av programmet

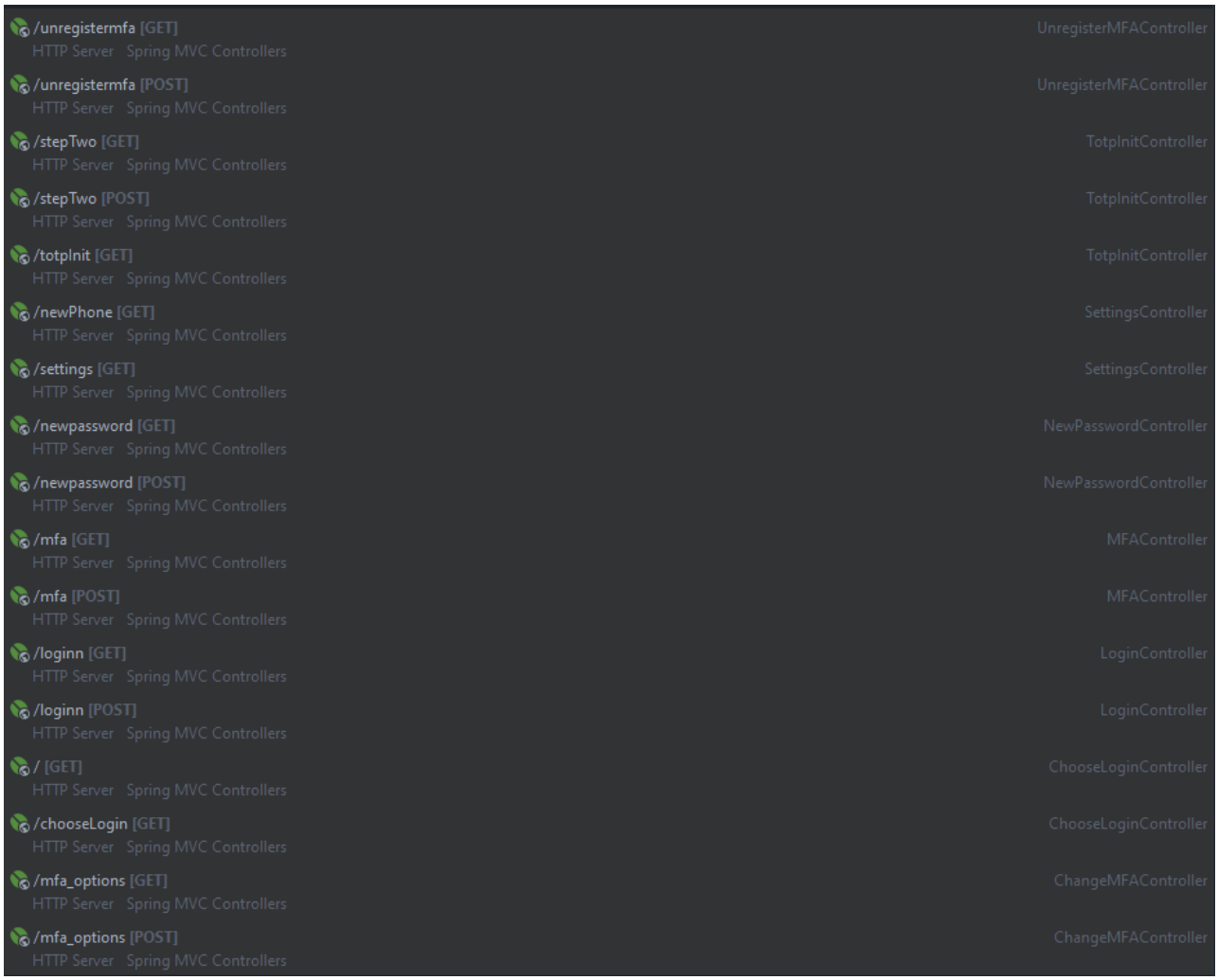
8 Dokumentasjon av kildekode

Det er skrevet Javadoc for metodene innenfor interesseområdet, som hovedsakelig dekker generering av TOTP og QR-kodene som brukes for førstegangsregistrering.

Ved eventuell endring i valgt SHA, må det også endres hvilken metode som blir brukt på linje 31 i Totp-klassen (i runTOTP metoden) og linje 48 i QrGenerator-klassen (i qrSetup metoden).

For bedre innsikt i metodenes funksjonalitet, kan testklassene inspireres for bedre kontekst.

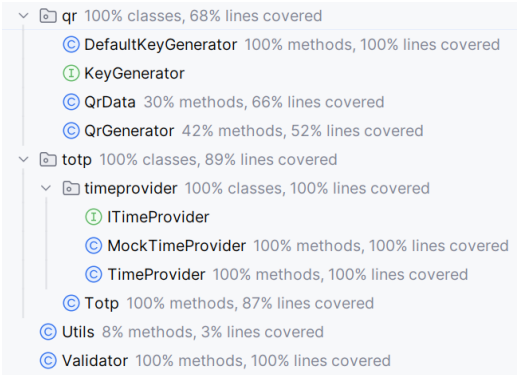
I figur 15 ligger en oversikt over de forskjellige endepunktene i løsningen, som også kan finnes i IntelliJ sin endepunktsoversikt.



Figur 15 Endepunktsdokumentasjon

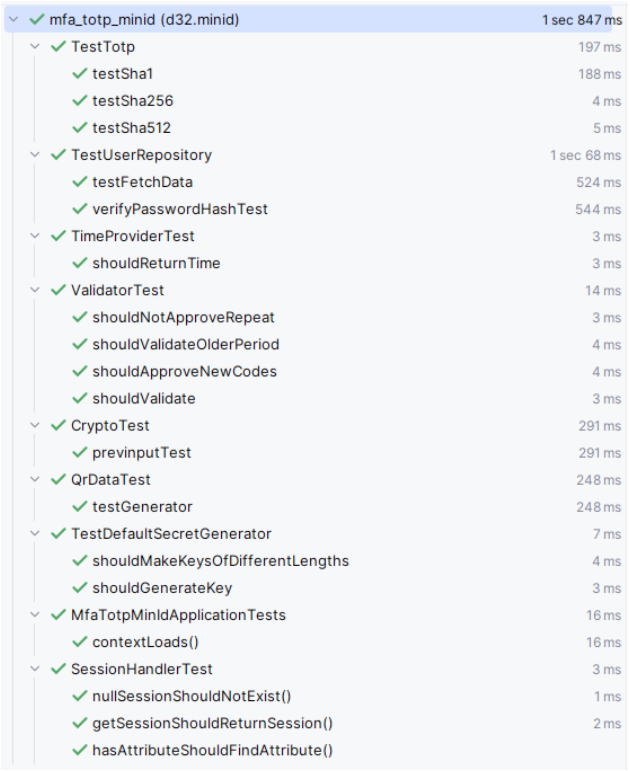
9 Testing

Enhetstester er skrevet for all kode som forholder seg den hovedsakelige funksjonaliteten i systemet. Hovedfokuset her er testing av klasser som forholder seg til TOTP-kode validering, generering av TOTP-koder, nøkler og QR-koder.



Figur 16 Testdekke av nøkkelkomponenter

På grunn av TOTP genereringens behov for å bruke tilfeldige (secret-keys) og bevegelige(timestamp) faktorer, vil testklassene ta nytte av konstante verdier og mock-objekter som gir forutsigbare resultater å veie opp imot.
Testsettet er vurdert ut ifra kodedekke av enkelte linjer/metoder/klasser for å gi en måleenhet for omfanget av testingen (Figur 15). Dette er en enkel fremgang for å måle testingen og tar ikke høyde for den mer sammenhengende strukturen i programmet, med flere logiske stier og uforutsett oppførsel.
Det er lagt fokus på å ha et sett med tester som dekker tilnærmet hundre prosent av kode som er selvskrevet, i motsetning til kode som er generert ved hjelp av en IDE. Typiske metoder for klasser som getters, setters, toString og equals er ikke inkludert i testing, da dette har forutsigbar oppførsel gitt at det ikke er vært skrevet om eller tilpasset. De tester som er blitt laget kan sees i Figur 16.



Figur 17 Alle tester skrevet for prosjektet

10 Referanser

KirstenS (n,d) *Cross Site Request Forgery (CSRF).* Tilgjengelig fra: https://owasp.org/www-community/attacks/csrf (Hentet: 11. april 2023)

OWASP (n,d) *Password Storage – OWASP Cheat Sheet Series.* Tilgjengelig fra: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html#bcrypt (Hentet: 11. april 2023)

Endring I Bruk Av Minid (2023) *Statsforvalteren i Oslo og Viken*. Tilgjengelig fra: https://www.statsforvalteren.no/oslo-og-viken/vergemal/nyheter---vergemal/2023/01/endring-i-bruk-av-minid/ (Hentet: 9. mai 2023).