



# BACHELOROPPGAVE:

## Kartprogram for industriell anvendelse med .NET & SQL

André Torsvik  
Ole Sunde

30. mai. 2022

## Dokumentkontroll

<i>Rapportens tittel:</i> Kartprogram for industriell anvendelse med .NET & SQL <i>Rapportens tittel på engelsk:</i> Map program for industrial use with .NET & SQL	<i>Dato/Versjon</i> 30. mai. 2022/1.00
<i>Forfatter(e):</i> André Torsvik Ole Sunde	<i>Rapportnummer:</i> BO22EB-16
	<i>Studieretning:</i> AUTB
	<i>Antall sider m/vedlegg</i> 77
<i>Høgskolens veileder:</i> Adis Hodzic	<i>Gradering:</i> Åpen
<i>Eventuelle Merknader:</i> Vi tillater at oppgaven kan publiseres.	

<i>Oppdragsgiver:</i> Høyskolen på Vestlandet	<i>Oppdragsgivers referanse:</i> BO22EB-16
<i>Oppdragsgivers kontaktperson(er) (inklusiv kontaktinformasjon):</i> Adis Hodzic	

## Forord

Denne rapporten dokumenterer vår avsluttende bacheloroppgave på studiet «Automatisering med robotikk» på Høgskolen på Vestlandet avdeling Bergen. Prosjektet inneholder hvordan vi utviklet et kartprogram og utfordringer vi møtte på.

Dette har vært et svært lærerikt prosjekt, og vi takker for oppgaven vi fikk av Høgskolen på Vestlandet. Vi har lært mye rundt det å planlegge større prosjekter og gode programmeringskikker.

Vi vil rette en stor takk til vår veileder Adiz Hodzic, for tips rundt flere programmeringsutfordringene og for gode råd om hva rapporten burde inneholde. Vi har fått god oppfølging og hatt god kommunikasjon igjennom hele semesteret.

Høgskolen på Vestlandet

Bergen, mai 2022

---

André Torsvik

---

Ole Sunde

## Sammendrag

Høyskolen på Vestlandet har gitt oss i oppgave å utrede bruk av OpenStreetMap og relaterte kartverktøy i programmer. I tillegg skal det lages et program som kan tegne OpenStreetMap-kart.

Oppgaven har som utgangspunkt at den skal gi bedre forståelse for hvordan kartdata og verktøy kan brukes i programmering. Den skal knytte sammen teori som er relevant for studenter med samme utdanningsløp som oss selv, med praktisk anvendelser av programmering i et industrielt perspektiv.

Vi har laget et program, med tilhørende dokumentasjon og brukermanual, som kan vise kartdata fra OpenStreetMap. Vi har også opprettet en SQL-database hvor vi har data for tenkte punkter og områder, som vi tegner på kartet. Brukere kan selv opprette, endre og slette disse gjennom programmet. Programmet har i tillegg en del funksjoner knyttet til brukerbehandling og filtrering av hva som vises.

Gjennom prosjektet har vi blitt kjent med en del biblioteker og utviklerverktøy som vi også beskriver i vår dokumentasjon. Azure, GMAP og Dapper er de vi fokuserer mest på.

Prosjektet resulterte i et program og en rapport, som vi føler demonstrerer gode eksempler for å programmere med kartverktøy, og verktøy generelt knyttet til programutvikling.

## Abstract

Western Norway University of Applied Sciences has accepted our application to write our bachelor's thesis on the use of OpenStreetMap and related mapping tools used in programming. In our submission we must also include a program that can draw OpenStreetMap maps.

The main goal of the task is to provide a better understanding of how map data and programming tools for maps can be used to create programs. Students that have covered a similar curriculum can expect theory and practical application of relevant programming to be explained together with an industrial perspective.

We have made a program that can display map data from OpenStreetMap, along with documentation and a user manual. We also set up a SQL-database where we store data that make up points and areas that we draw on top of the map. Users can create, update and delete these by using the program. The program also has a number of functions that let users administer user accounts and filter what data is shown on top of the map.

Throughout the project we have gotten to know some libraries and developer tools that we describe in our documentation. The ones we concentrate on are Azure, GMAP and Dapper.

The result of our labor is a program and a report that we think contains good examples for programming with map related tools and tools for general program development.

# Innhold

Dokumentkontroll .....	40
Forord.....	41
Sammendrag .....	42
Abstract .....	43
Figuroversikt.....	44
Tabelloversikt .....	44
1 Innledning.....	44
1.1 Organisering av rapporten .....	45
1.2 Oppdragsgiver .....	46
1.3 Problemstilling.....	47
1.4 Hovedidé for løsningsforslag.....	48
2 Kravspesifikasjon .....	58
2.1 Liste over kravspesifikasjoner .....	59
2.1.1 Liste over tilleggskrav .....	61
3 Analyse av problemet.....	61
3.1 Utforming av mulige løsninger .....	62
3.1.1 Løsningsalternativ 1 (Valgt løsning) – Bruke et egnet bibliotek .....	63
3.1.2 Løsningsalternativ 2 – Egen versjon .....	64
3.1.3 Løsningsalternativ 3 – Mobilapplikasjon .....	65
3.1.4 Vurderinger i forhold til verktøy og HW/SW komponenter .....	66
3.2 Konklusjon for valg av løsning .....	67
4 Teori .....	67
4.1 Utviklerverktøy - Visual Studio & .Net.....	67
4.1.1 IDE - Visual Studio.....	68
4.1.2 Språk - C#.....	68
4.1.3 Utviklerplattform - .NET .....	69
4.1.4 Plattform – Windows & Rammeverk - Windows Forms.....	69
4.2 Database - SQL & Azure.....	18
4.2.1 Azure .....	18
4.2.2 SQL & T-SQL.....	18
4.3 Biblioteker .....	18
4.3.1 GMAP.....	18

4.3.2	Dapper .....	19
4.4	OpenStreetMap .....	19
4.4.1	Rådata i OpenStreetMap .....	19
4.4.2	Viktige funksjoner .....	23
5	Realisering av valgt løsning .....	24
5.1	Utviklerverktøy .....	24
5.1.1	Plattform, språk, IDE, utviklerplattform og rammeverk .....	24
5.2	Database (Lagring av data) .....	24
5.2.1	Database – design .....	25
5.3	Biblioteker .....	27
5.3.1	GMAP .....	27
5.3.2	Dapper .....	28
5.4	Utvikling av programvare .....	29
5.4.1	Brukergrensesnitt .....	29
5.4.2	Programkode .....	35
5.5	Sikkerhet .....	37
5.5.1	Brannmur .....	37
5.5.2	Sikkerhetsklarering .....	37
5.5.3	Bruker .....	37
5.5.4	Admin kontroll .....	37
5.5.5	Starte programmet .....	39
5.6	Filter .....	41
5.6.1	Database-filter .....	41
5.6.2	Kategori-filter .....	41
5.6.3	Tag-filter .....	42
6	Testing .....	44
6.1	Testing underveis .....	44
6.2	Logge feil .....	44
6.3	Brukervennlighetstest .....	45
7	Diskusjon .....	47
7.1	Forslag til forbedring .....	47
7.1.1	Forbedring 1 – Mer detaljerte kart .....	47
7.1.2	Forbedring 2 – Lage nettside med .ASP .....	48
7.1.3	Forbedring 3 – Kryptering av kommunikasjon .....	48

7.1.4	Forbedring 4 – Innstillingsside før innlogging.....	48
7.1.5	Forbedring 5 – Forbedre og samkjøre SQL-spørringene .....	48
7.2	Kjente feil/bugs .....	49
7.3	Ting vi har lært gjennom vårt arbeid med oppgaven.....	49
7.3.1	GitHub .....	49
7.3.2	Bibliotekene GMAP og Dapper .....	49
7.3.3	Azure og Transact-SQL.....	49
7.3.4	SQL Stored Procedures og functions .....	49
7.3.5	BindingList .....	49
8	Konklusjon.....	50
	Referanser.....	51
	Appendikser .....	56
Appendiks A	Forkortelser og ordforklaringer.....	56
Appendiks B	Prosjektledelse og styring.....	57
B.1	Prosjektorganisasjon .....	57
B.2	Fremdriftsplanning .....	57
Appendiks C	Brukerdokumentasjon.....	58
C.1	Oppsett av programvare .....	58
C.2	Ikoner .....	58
C.3	Brukerdokumentasjon.....	59
C.4	Drifts- og vedlikeholdsdocumentasjon.....	61
C.5	Oppretting av Azure Database .....	61
C.6	Koble Visual Studio prosjektet ditt mot Azure Databasen .....	66
C.7	Opprette en «Google Cloud API key».....	68
Appendiks D	Liste over vedlegg .....	71
Appendiks E	Creative Commons Attribution 4.0 International Public License .....	72

## Figuroversikt

Figur 1 - Relasjon mellom valgte teknologier .....	37
Figur 2 - Oversikt og sammenheng mellom verktøy som finnes i vårt prosjekt .....	39
Figur 3 - Eksempel på kode som bruker Dapper til å utføre en spørring. [38] .....	41
Figur 4 - Struktur av data i OpenStreetMap [62].....	41
Figur 5 - «Node» fra OpenStreetMap [30] .....	41
Figur 6 - Eksempel på en node .....	42
Figur 7 - «Way» fra OpenStreetMap [30].....	44

Figur 8 - «Relation» fra OpenStreetMap [30].....	44
Figur 9 - ER-modell for «Objekt» og tilhørende tabeller.....	44
Figur 10 - ER-modell for «Område» og tilhørende tabeller.....	45
Figur 11 - ER-modell for Bruker og for Log_feil.....	47
Figur 12 - Klassen «Bruker».....	47
Figur 13 - Databasetabellen «Bruker».....	47
Figur 14 - Tidlig skisse av programmet.....	48
Figur 15 - Vanlig design mønster.....	48
Figur 16 - Windows Mail desktop applikasjon.....	48
Figur 17 - Teams desktop applikasjon .....	48
Figur 18 - Mal for vinduer i programmet.....	49
Figur 19 - Visningsvinduet .....	49
Figur 20 - Redigeringsvinduet.....	49
Figur 21 - Kontrollpanelvinduet .....	49
Figur 22 - «Light Theme», «Dark Theme», «Original Theme».....	49
Figur 23 - Admin: Brukere som venter på godkjenning.....	49
Figur 24 - Admin: Brukere som ikke har skrevet inn verifiseringskode.....	49
Figur 25 - Admin: Oppgradering og nedgradering av sikkerhetsklarering, og fjerning av brukere. ....	50
Figur 26 - Flytskjema for å åpne visningsvinduet. ....	51
Figur 27 - Utklipp av filter i programmet.....	56
Figur 28 - Flytskjema for database og kategori-filter .....	14
Figur 29 - Flytskjema for tag-filteret mot objekter (identisk for områder) .....	16
Figur 30 - Logging for ikke-statiske metoder .....	19
Figur 31 - Logging for statiske metoder, her i klassen: «Filbehandling» .....	20
Figur 32 - Utklipp av feilloggen i databasen .....	20
Figur 33 - Test for person uten forkunnskaper .....	21
Figur 34 - Områdevelger i aktivert og deaktivert tilstand. ....	22
Figur 35 - Kart over HVL, hentet fra Mazemap.....	22
Figur 36 - Kart over HVL, hentet fra OpenStreetMap.....	25
Figur 37 - Husikon med hvit bakgrunn   Figur 38 - Husikon fra Pichon .....	25
Figur 39 - Hovedseksjonene i vårt program. ....	26
Figur 40 - Opprette «Azure SQL» database: Ressurstype.....	28
Figur 41 - Opprette «Azure SQL» database: Databasetype.....	28
Figur 42 - Opprette «Azure SQL» database: «Basics» .....	29
Figur 43 - Opprette «Azure SQL» database: Opprette en server .....	<b>Error! Bookmark not defined.</b>
Figur 44 - Opprette «Azure SQL» database: Oversikt og bekreft oppretting .....	30
Figur 45 - Azure: Dine ressurser .....	<b>Error! Bookmark not defined.</b>
Figur 46 - Utfør spøringer .....	<b>Error! Bookmark not defined.</b>
Figur 47 - Tilkoblingsstrenger .....	33
Figur 48 - Tilkoblingsstreng i Visual Studio prosjektet .....	33
Figur 49 - Eksempelkode for å hente ut tilkoblingsstreng.....	34
Figur 50 - Eksempelkode for å utføre en spørring.....	34
Figur 51 - Legge inn din IP i «Azure SQL server» sin «Whitelist» .....	38
Figur 52 - Google Maps Platform .....	38
Figur 53 - Google Cloud API key .....	39

## Tabelloversikt

Tabell 1 - Eksempel med noen «nøkkel og verdi» data.....	22
Tabell 2 - Filtreringsrekkefølge for vår kode. ....	41
Tabell 3 - Tabellen fra "Settings" som vist i figur 48.....	67

## 1 Innledning

### 1.1 Organisering av rapporten

Alle bilder og figurer i rapporten er også å finne som vedlegg, i full oppløsning. Forkortelser og begreper forklares i Appendiks A. Første gang de blir brukt vil de være markert med stjerne \*, for eksempel: HVL\*.

### 1.2 Oppdragsgiver

Vår oppgave er en intern oppgave gitt av HVL\* ved institutt for data, elektro og realfag. Vår kontaktperson og veileder for oppgaven er Adis Hodzic [1].

### 1.3 Problemstilling

Oppgaveteksten slik den ble gitt oss: [2]

***Innledning:***

*«Det skal utredes for hvordan en kan bruke OpenStreetMap (og relaterte kart-verktøy) i programmer. Et programsystem som kan tegne OpenStreetMap-kart med brukerposisjon (GPS) skal utvikles/testes.»*

***Faglige elementer:***

*«Programmering med C#, systemutvikling, programmering for nettverk, GPS, mobilenhetsprogrammering, ...»*

I tillegg fortalte oppdragsgiver følgende om oppgaven:

*«Jeg synes teknologien bak kartdata og kartprogram er interessant, og har mange potensielle bruksområder innen industriell IT. Jeg vil derfor ha noen til å utforske dette området innen programmering og lage et program som kan fungere som et godt eksempel og referanse for andre.»*

### 1.4 Hovedidé for løsningsforslag

Sammen med oppdragsgiver kom vi fram til en visjon som grunnlag for løsningen av oppgaven. Hovedidé for løsningsforslag var å utarbeide et program, med tilhørende rapport (dokumentasjon), som kan bruke kartdata fra OpenStreetMap sammen med egenprodusert data. Konseptet vi så for oss var et program som kan brukes av en bedrift/organisasjon til å organisere ressurspunkter i et område de arbeider i.

## 2 Kravspesifikasjon

Som det kommer frem i problemstillingen vi gikk igjennom i kapittel 1.3, så har oppgaven ingen lang liste med krav. Etter å ha diskutert oppgaven med oppdragsgiver/veileder ble vi enige om følgende definisjon for oppgaven:

Oppdragsgiver ønsker å få en bedre forståelse for programmering med kartdata, fortrinnsvis ved bruk av OpenStreetMap. Vår oppgave er å utforske bruk av OpenStreetMap i programmering, dokumentere funnene i en rapport og å demonstrere bruken i et program. Det er en fordel om programmet er utformet som om det var i bruk i industri/næringsliv. Nivået bør ligge slik at studenter med samme bakgrunn og erfaring som oss selv (automatiseringsingeniør 6. semester) kan forstå hvordan programmet virker ved å lese dokumentasjonen og kildekoden. Så lenge løsningene er skalerbare så trenger ikke eksempladata å være av et omfang tilsvarende bruk i industri/næringsliv. Tillegg som vi ønsker å utforske, men som kan tenkes å være tidskrevende, bør defineres som tilleggskrav og bare utføres dersom det viser seg at det er tid til overs. Opgaven bør kunne fungere helt fint som en oppgave, helt uten tilleggskravene.

Dette la vi til grunn for følgende kravspesifikasjon.

### 2.1 Liste over kravspesifikasjoner

- Opgaven skal gi innsikt i en valgt løsning for kartdata, og demonstrere bruk i programmering.
- Det er ønskelig at oppgaven løses ved bruk av OpenStreetMap. Andre løsninger kan brukes dersom det finnes gode argumenter for valget.
- Det skal lages ett program som demonstrerer bruk av kartdata til å vise et kart. I tillegg skal programmet kunne vise data på toppen av kartet, i en ordnet form. Denne dataen skal kunne komme fra et annet sted enn kartdata.
- Programmet skal lages slik at det kunne vært i bruk hos en industriell aktør. Løsningene i programmet bør tåle skalering på mengde data og antall brukere.
- Programmet skal skrives med leselig kode, og ha en ryddig struktur, slik at studenter med tilsvarende erfaring kan sette seg inn i prosjektet og forstå hvordan programmet fungerer.
- Prosjektet og programmet skal ha brukerdokumentasjon/manual/kildekode som gjør at andre kan sette opp prosjektet og prøve det ut.

#### 2.1.1 Liste over tilleggskrav

- Muligheten for å vise kartet med data, eller hele programmet i en nettside, slik at det kan lastes inn på ulike mobile enheter.
- Muligheten for å vise programmet, eller en versjon av programmet i AR-enheter (eller VR-enheter).
  - Presis forankring i bredde og lengdegrader pluss høyde kan være en mulighet for å knytte data til en lokasjon i 3D.

### 3 Analyse av problemet

Etter å ha gjennomgått kravspesifikasjoner med veileder, så begynte vi å få en idé om hvordan vi ønsket å løse oppgaven vår. Ettersom det er vanskelig å anslå tidsbruken på et slik program når man er veldig tidlig i planleggingsfasen valgte vi også ut en del tilleggsløsninger som vi planla å arbeide med, dersom tiden tillot det. Oppgaven måtte allikevel kunne stå alene uten noen av disse tilleggene, ettersom det var en mulighet for at all tiden går med til å utføre hovedoppgaven.

#### 3.1 Utforming av mulige løsninger

Oppgaven var veldig åpen, slik at vi kunne selv bestemme hvordan vi ville anvende OpenStreetMap i programmet og hvilken retting vi ønsket å rett programmet mot. Som tidligere nevnt, ønsker vi å knytte programmet vårt opp mot industrien. Vi kunne selv velge programmeringsspråk, slik at vi kunne velge det vi var mest komfortable med eller hvis det var ett som skilte seg ut med gode løsninger.

En komponent som var nevnt i oppgavebeskrivelsen i kapittel 1.3, var bruk av GPS\*, som kan brukes for å avgjøre brukerens lokasjon og tilpasse kartet i forhold til dette. Det er derimot slik at det ikke er vanlig å ha GPS eller annen GNSS\* (satellittnavigasjon) i enheter som ikke benytter mobilkommunikasjon. Dette kan være en bærbar datamaskin, stasjonær datamaskin eller maskinvare som skytjeneste. Vi bestemte oss allikevel for å definere noen forslag til løsninger som ikke inneholdt mobilenhetsprogrammering og GPS. Etter å ha diskutert dette med oppdragsgiver kom vi fram til at disse andre løsningene også inneholdt stor grad av faglig relevans og framstod som like gode løsninger. Dermed anså vi ikke «mobilenhetsprogrammering» eller «GPS» som kriterier for en løsning.

##### 3.1.1 Løsningsalternativ 1 (Valgt løsning) – Bruke et egnet bibliotek

Lage en Windows Forms\* applikasjon ved hjelp av egnet bibliotek og data fra OpenStreetMap. Programmet skal kunne kartlegge hvor forskjellige objekter er. Objektene kan være hva som helst, men hovedtanken er sensorer og andre tekniske ting. Objektene som blir laget vil være tilgjengelig gjennom programmet, men lagret uavhengig av OpenStreetMap sine data. Vi ønsker også å ha med andre kjernefunksjoner som klassiske kart-programmer har, som å finne lokasjon med hjelp av adresse.

##### 3.1.2 Løsningsalternativ 2 – Egen versjon

Lage vår egen versjon av et kartprogram basert på XML\* formatet til «.OSM» filene. Løsningen må kunne tegne enkle kart for et relativt lite område. Løsningen må kunne nytte egne data i tillegg til data fra OpenStreetMap. Løsningen vil i stor grad bygge på løsning 1, men uten hjelp av et egnet bibliotek.

##### 3.1.3 Løsningsalternativ 3 – Mobilapplikasjon

Lage en applikasjon som kan brukes på mobiltelefon. Her måtte vi ha valgt om vi skulle fokusere på iOS\* eller Android\*. Denne ville også i stor grad bygge på løsning 1, men enheten som brukere applikasjonen ville vært en mobil istedenfor en PC.

##### 3.1.4 Vurderinger i forhold til verktøy og HW/SW komponenter

Vi valgte å programmere oppgaven vår med programmeringsspråket C#\*. Vi har fra tidligere erfaring med dette språket igjennom emnene ELE102\*, ELE205\* og ELE301\*. Gjennom emnene erfarte vi at Visual studio\* er brukervennlig når det gjelder å lage GUI\*.

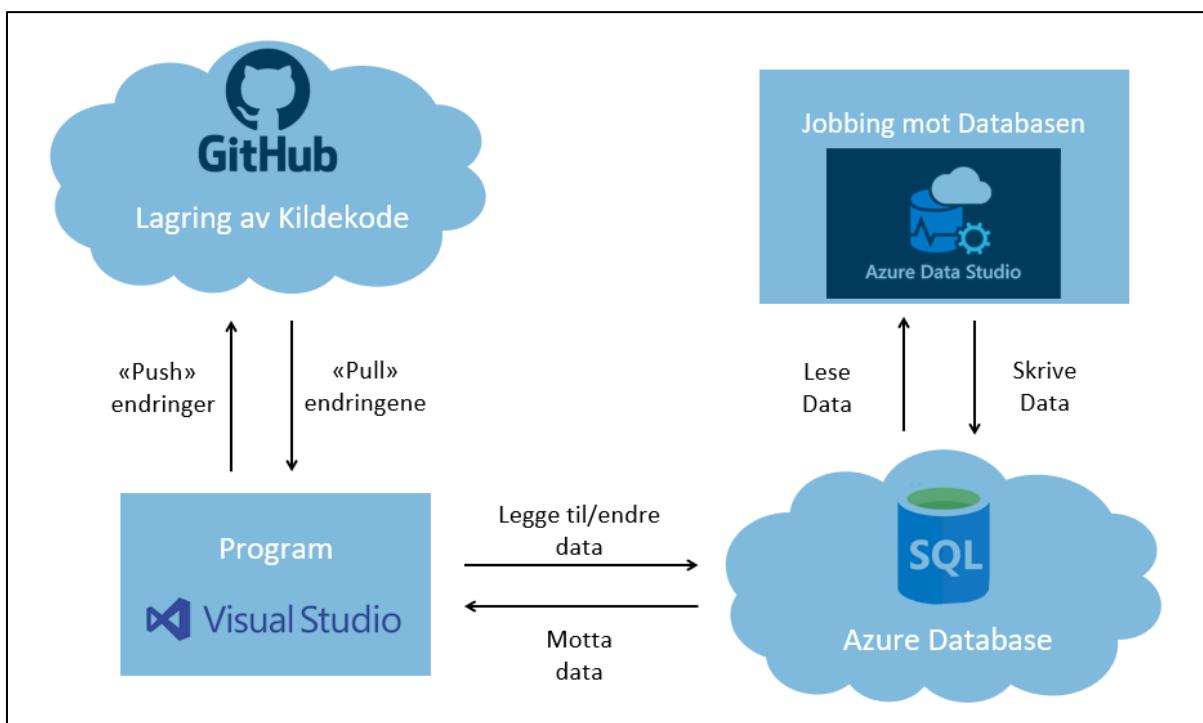
Et annet alternativ kunne være å programmere oppgaven i Python\*. Python er et språk vi har liten erfaring med fra før av, kun litt fra emnet ELE306\*. Derfor ville dette være tidkrevende å lære, spesielt med tanke på at det å utvikle GUI er annerledes fra hvordan vi har gjort det i Visual Studio. Derfor

foretrakk vi C# siden dette vil spare oss tid og vi har større forutsetning for å kunne lage et program med høyere kvalitet. Hvis dette programmet skulle bli brukt som referanse i undervisningen, er det i tillegg bedre å bruke et språk som studentene har et forhold til.

GitHub\* [3] er en skylagringstjeneste som hører til Git [4], og er svært godt egnet til lagring av kildekoden til programmer. Derfor er det et naturlig valg når det gjelder kildekontrollsysteem. I tillegg til lagring av koden får vi også versjonskontroll, som vil si at vi lett kan hente tilbake tidligere versjoner av programmet dersom det skulle oppstå problemer med koden. GitHub gjør det også lettere å jobbe samtidig med programmet. Vi har fra før av liten erfaring med GitHub, og det vil derfor kreve opplæring.

Når det gjelder lagring av data, har vi fra tidligere av erfaring med PostgreSQL fra emnet ELE305. Vi valgte å bruke en sky database\* fra Azure\*. Den databasen som dekket vårt behov, bruker Transact-SQL (T-SQL) som spørrespråk. Det er kun små endringer i syntaksen som skiller språkene, og det er derfor ikke tidkrevende å bytte om. Det er også andre forskjeller mellom PostgreSQL og T-SQL, men de har liten eller ingen påvirkning for vårt bruk i denne oppgaven [5]. I tillegg er Azure sin database enkel å koble opp mot, sammenlignet med å drive egen server.

Når vi skal lage tabeller, gjøre endringer, fjerne eller andre ting knyttet til databasen, bruker vi programmet: Azure Data Studio [6]. Vi kan også jobbe direkte mot databasen på nettsiden til Azure, men brukergrensesnittet er bedre i Azure Data Studio.



Figur 1 - Relasjon mellom valgte teknologier

### **3.2 Konklusjon for valg av løsning**

Løsningsalternativ nr.1 er den løsningen som vi valgte å gå for. Vi fant et bibliotek som var svært godt egnet til de funksjonene vi ønsket at programmet vårt skulle inneholde. Dette gjør at vi sparer mye tid, og kan fokusere på å gjøre koden så god, leselig og robust som mulig. I tillegg, hvis programmet skulle bli brukt som referanse i undervisningen, ser vi for oss at dette er den løsningen som ville gi størst læringsutbytte for studentene. De fleste har tilgang til en PC med Windows som operativsystem [7] og derfor ble valget å lage en Windows Forms applikasjon.

Løsningsalternativ nr.2 ville gitt oss mest kunnskap rundt kartprogrammering og hvordan kart fungerer generelt. Det ville tatt mye tid og ressurser å sette seg inn i programmer som tegner, og nødvendige API-er\* for kommunikasjon. Her finnes det allerede gode og eksisterende løsninger lett tilgjengelig. Siden vi ønsket at løsningsalternativ nr.1 og nr.2 skulle gå i samme retning, ville kvaliteten på produktet i løsnings nr.2 bli svakere. Store deler av tiden ville gått til å få programmet til å klare å tegne kart, og derfor ville vi fått mindre tid til de andre funksjonene i kravspesifikasjonen. I tillegg har vi ingen erfaring med å tegne i programmer fra tidligere av, så dette ville være tidkrevende med tanke på opplæring.

Løsningsalternativ nr.3 kunne være et godt alternativ siden vi da får tilgang til GPS-en til mobil enheten, og dermed hadde hatt mulighet til å bruke den aktiv i programmet. Siden vi må velge om vi skal lage appen til iOS, Android eller andre systemer, falt denne løsningen bort. Selv om mesteparten av den norske befolkningen bruker iOS som operativsystem [8], ville fremdeles et stort antall ikke få tilgang til applikasjonen. I tillegg har vi ingen erfaring med å lage mobilapper fra før og dette ville kreve opplæring.

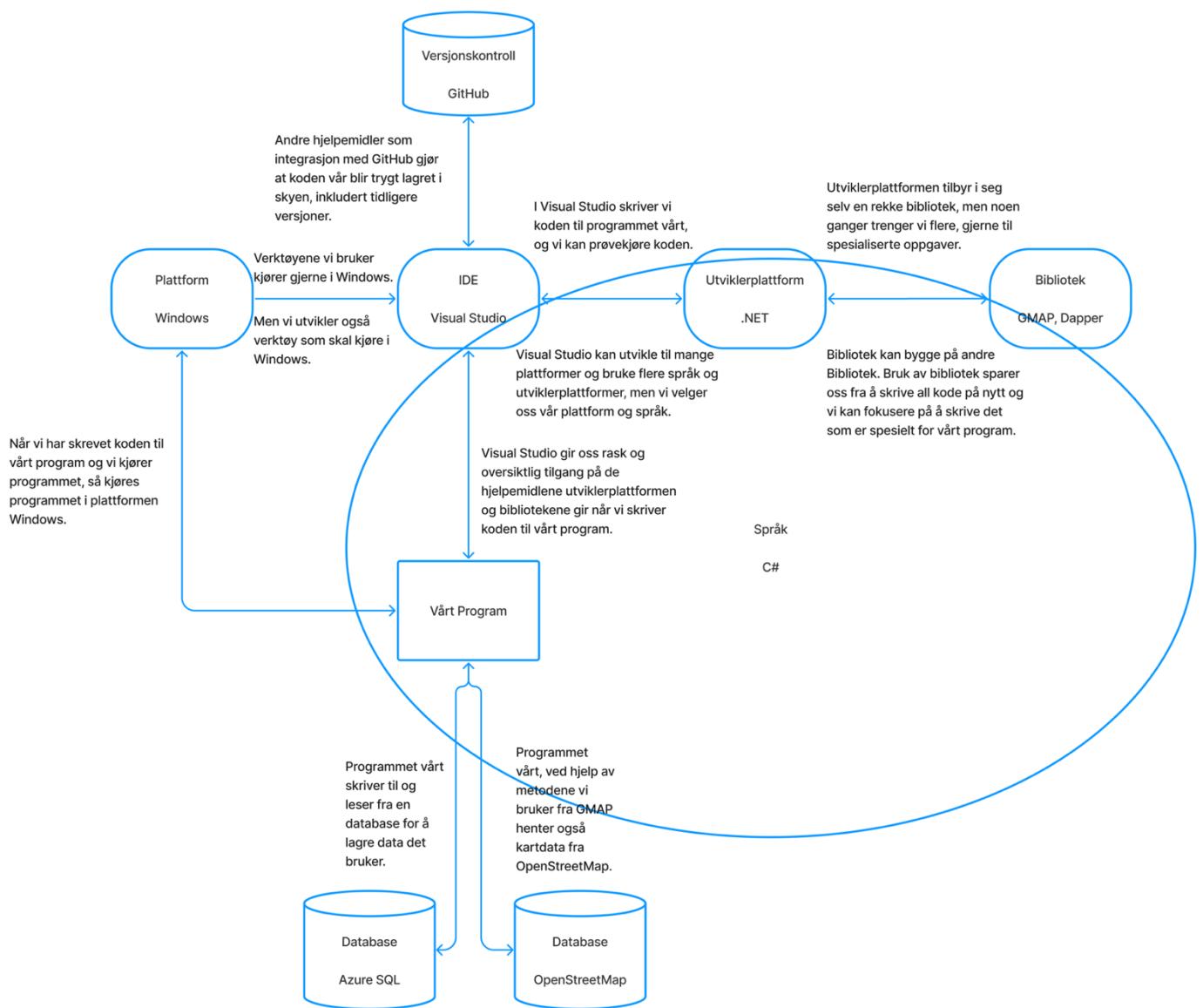
Etter vurdering av fordeler og ulemper passet løsning nr.1 best.

## 4 Teori

I dette kapittelet går vi igjennom teori knyttet til utviklerverktøy, OpenStreetMap, database og biblioteker. Her presenterer vi kort om de forskjellige elementene som er relevant til vårt prosjekt, knyttet til temaene.

### 4.1 Utviklerverktøy - Visual Studio & .Net

Plattform, utviklerplattform, «Integrated development environment» (IDE\*), språk, biblioteker og database. Det kan av og til være vanskelig å forstå seg på hva som er hva. Selv om vi gjerne kjenner verktøyene slik vi bruker de, kan det være vanskelig å ha oversikten på hva som passer inn hvor i det store bildet, spesielt når man er i startfasen med å lære seg programutvikling. Vi skal gjøre vårt beste for å belyse aktuelle verktøy og miljø for dette prosjektet, og sette de i sammenheng.



Figur 2 - Oversikt og sammenheng mellom verktøy som finnes i vårt prosjekt

#### **4.1.1 IDE - Visual Studio**

Visual Studio som integrated development environment (IDE) er en av de mest brukte verktøyene ifølge den årlige undersøkelsen til Stackoverflow. [9] [10] [11] [12]

Visual studio er utviklet av Microsoft og blir derfor et naturlig valg for å utvikle programvare til Windows. Visual Studio tilbyr en rekke verktøy utviklet av Microsoft for å dra nytte av de teknologiene som er tilgjengelig, .NET, GitHub og Azure, men også muligheten for å bruke verktøy utviklet av andre aktører og personer som f.eks. Dapper. Visual Studio Community Edition er tilgjengelig gratis fra Microsoft. Den fullspekkede versjonen Visual Studio Enterprise Edition er tilgjengelig gratis for studenter sammen med «Azure for Students». [13] [14]

#### **4.1.2 Språk - C#**

C# er utviklet av Microsoft ble opprinnelig sluppet sammen med .NET. [15] Det er et programmeringsspråk laget for å være et enkelt, moderne og allsidig objekt-orientert språk. Språket danner grunnlaget for koden vi skriver, og måten den henger sammen. Enkelte verktøy støtter også bruk via andre språk.

Fra C# får vi elementer som klasser (Class), strukturer (Struct) og egenskaper (Properties\*) for å nevne noen.

#### **4.1.3 Utviklerplattform - .NET**

.NET (dot net) er en utviklerplattform. En utviklerplattform består av programmeringsspråket og biblioteker, satt sammen for å danne et grunnlag for å utvikle programvare til en rekke plattformer. [16] [17]

Eksempler på biblioteker i .NET som vi gjerne kjenner og bruker er «System.Collections», som gir oss lister som ArrayList og SortedList og «System.Math» som gir oss Min(), Max(), Pow() og Sqrt(). I tillegg finnes det mange flere. [18]

#### **4.1.4 Plattform – Windows & Rammeverk - Windows Forms**

Programmer skrives ofte til en plattform. Det finnes løsninger som gir muligheten til å skrive til flere plattformer samtidig, men koden blir gjerne generell og ikke tilpasset plattformen. Det er ikke uhørt å skrive størsteparten av koden for flere plattformer og så tilpasse den delen av koden som er spesifikk for plattformen. [19]

Windows Forms (.NET Framework) er et rammeverk\* for brukergrensesnitt (UI) som forenkler oppgaven med å lage en applikasjon for bruk til plattformen Windows. Rammeverket gjør det at man visuelt kan utvikle brukergrensesnittet ved å dra elementer fra en verktøykasse (toolbox) til dit man ønsker i programmet. [20]

## 4.2 Database - SQL & Azure

Det å bruke en database til å lagre data tilhørende et program har mange klare fordeler. Det er lett å dele data mellom flere brukere når det er ønskelig, det er lettere å håndtere data på en trygg måte og det er lett å skalere programmet. Det finnes mange ulike tilbydere av databaseløsninger, både for bruk lokalt på egen maskinvare og som skytjenester. Det å sette opp en database på en lokal maskin kan være en billig løsning, men krever en del tilpasning for at alt skal virke knirkefritt. I dag har det blitt veldig lett og populært å opprette databaser i skyen via skytjenestetilbydere. En av de største tilbyderne av databaseløsninger via skyen er Microsoft. [24]

### 4.2.1 Azure

Azure er navnet på Microsoft sine skyløsninger og tilhørende rammeverk. En viktig del av Azure er muligheten til å leie løsninger som tjenester. En rekke løsninger tilbys, deriblant databaser, virtuelle maskiner og infrastrukturløsninger. Studenter har tilgang på å prøve ut en rekke av tjenestene i Azure via kampanjen «Azure for Students». [14]

### 4.2.2 SQL & T-SQL

Structured Query Language (SQL) er et mye bruk språk for å håndtere data i databasesystemer. SQL er bygget på konseptet at data i en database har relasjoner til annen data. Det finnes flere varianter basert på SQL. En av variantene, som er utviklet av og blir brukt av Microsoft, er Transact-SQL (T-SQL). Varianten bygger på SQL, men har små forskjeller i syntaks. [25] [26]

#### 4.2.2.1 T-SQL Prosedyrer

Prosedyrer i SQL fungerer på samme prinsippet som en metode i C#. Prosedyren opprettes, og du kan kalle på prosedyren ved navn. Prosedyren kan ha argumenter og kan returnere et resultat. I tillegg til å være mer effektiv kan riktig konstruerte prosedyrer også gjøre det vanskeligere å gjennomføre SQL Injection. [27]

##### 4.2.2.1.1 SQL Injection

Kort forklart er det en sårbarhet som gjør at tekst som skrives inn av en bruker kan oppfattes som kode og utføre uønskede handlinger i databasen.

#### 4.2.2.2 T-SQL Funksjoner

Funksjoner kan brukes til å bearbeide verdier. En funksjon utfører en operasjon på en input og returnerer en output. En prosedyre kan kalle på en funksjon, men en funksjon kan aldri kalle på en prosedyre. [28] [27]

#### 4.2.2.3 T-SQL Sekvenser

En sekvens brukes for å telle lags i en forhåndsdefinert tallrekke. F.eks kan du bruke en sekvens til å tilordne hver rekke et nytt tall i området 10001-99999. Sekvensen henter neste verdi den ikke har brukt når den kalles på, og returnerer verdien. Du kan også velge hvor mye du vil inkrementere sekvensen med for hver bruk, og du kan definere ulike tall-datatyper (smallint, int, bigint osv.). [29]

## 4.3 Biblioteker

Et bibliotek i programmering er en samling av ressurser som kan brukes i programkoden slik at den som programmerer ikke alltid må skrive all kode fra grunnen av. Mange bibliotek har åpen kildekode og er gjerne tilgjengelig for fri bruk, eller fri ikke-kommersiell bruk.

### 4.3.1 GMAP

GMAP er et bibliotek til .NET som har forhåndsdefinerte metoder for å hente og vise kartdata fra forskjellige tilbydere. GMAP biblioteket er åpen kildekode og er tilgjengelig for fri bruk [30]. Med biblioteket opprettes et objekt som viser kart fra en av tilbyderne som støttes, med kun få linjer med kode.

GMAP bygger på .NET og er tilpasset blant annet Windows Forms, men har også tilpasninger for flere rammeverk. Til Windows Forms har biblioteket allerede definert grunnleggende funksjoner som, å dra i kartet for å flytte området du ser på, zoome inn og ut, og hente ut bredde og lengdegradene for et område du klikker på.

Biblioteket definerer objekter basert på lengde og breddegrader, så man kan enkelt selv legge til markører og områder på kartet og forankre de ved bruk av bredde- og lengdegrader.

#### 4.3.2 Dapper

Dapper er et bibliotek til .NET som hjelper med «Object-relational mapping», dvs. knytte databaseobjekter til objekter i programkoden. Fordelen med å bruke Dapper framfor f.eks. «Entity Framework Core», er at vi kan enkelt utføre spøringer i SQL kode. Når vi har opprettet klasser som er like tabellene i databasen, knyttes resultatet av spørringen til objekter av den klassen. De enkelte dataverdiene knyttes til egenskaper i klassen. Vi kan også koble spørringen direkte til en liste av objekter slik at hver rad(tuppel) utgjør et objekt i listen. [31]

```
1 reference
public List<Person> GetPeople(string lastName)
{
    using ( IDbConnection connection = new System.Data.SqlClient.SqlConnection(Helper.CnnVal("SampleDB")))
    {
        var output = connection.Query<Person>($"select * from People where LastName = '{ lastName }'").ToList();
        return output;
    }
}
```

Figur 3 - Eksempel på kode som bruker Dapper til å utføre en spørring. [38]

Når man bruker Dapper til å utføre en spørring kan man bruke erklæringen «using», da lukkes tilkoblingen til databasen automatisk når spørringen er ferdig utført.

Dapper er svært ressurseffektivt. [32]

#### 4.4 OpenStreetMap

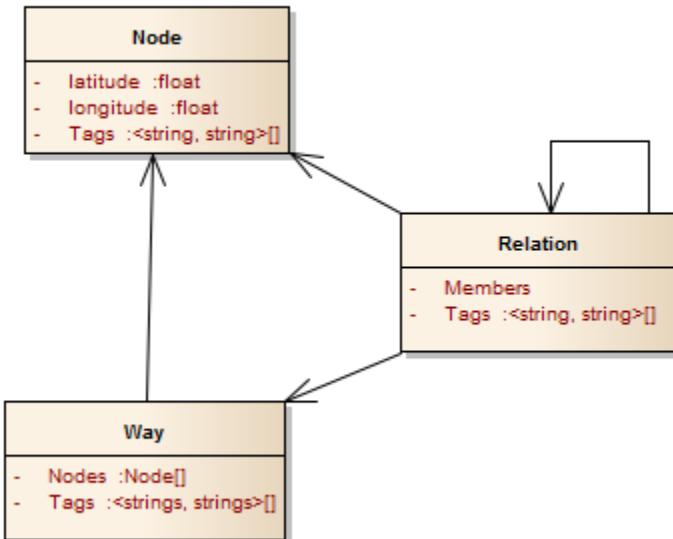
OpenStreetMap [21] er et prosjekt hvor frivillige arbeider for å holde en oppdatert database med geografisk informasjon fra hele verden. Frivillige oppfordres til å bidra med lokal informasjon.

Prosjektet støttes av den ideelle organisasjonen «The OpenStreetMap Foundation» [22].

OpenStreetMap sin data er gratis og åpen for alle å bruke, så lenge du krediterer OpenStreetMap.

##### 4.4.1 Rådata i OpenStreetMap

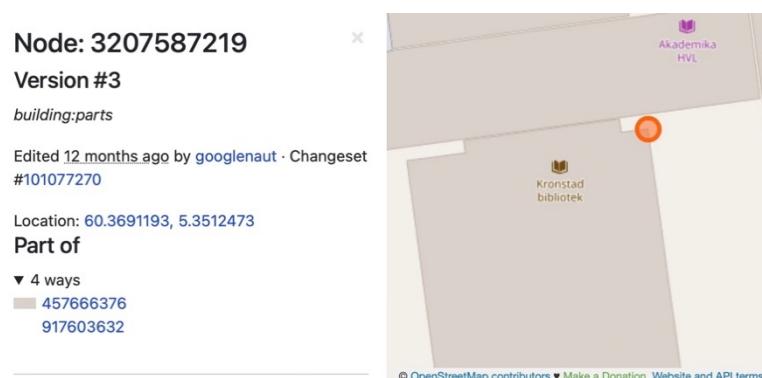
Informasjon lagret i OpenStreetMap kan eksporteres som «.osm» filer. Hvis vi ser nærmere på disse kan vi få et godt inntrykk av hvordan dataene blir lagret og kategorisert i OpenStreetMap. Filene inneholder data i XML format. OpenStreetMap organiserer data i 3 hovedelementer [23]; Node (definerer punkter i planet), Way (definerer lineære objekter eller områdegrenser) og Relation (knytter sammen objekter som er del av noe større).



Figur 4 - Struktur av data i OpenStreetMap [62]

#### 4.4.1.1 Node

Nesten all informasjon i OpenStreetMap blir lagret i elementet «node». I tillegg til metadata tilhørende versjon og endringer for OpenStreetMap inneholder en node alltid følgende: En unik id som identifiserer noden, breddegrad og lengdegrad som plasserer noden på kartet. Noder er alltid et enkelt bestemt punkt på kartet med en gitt bredde- og lengdegrad.



Figur 5 - «Node» fra OpenStreetMap [30]

En komplett liste over data som lagres i en node er:

- id(int)
- visible(true)

- version(int)
- changeset(int)
- timestamp(timestamp)
- user(string)
- uid(int)
- lat (float)
- lon (float)
- tag ([String, String]) – Ikke alle noder inneholder tag(s).

```
1  <node id="3938248119"
2    |  visible="true"
3    |  version="4"
4    |  changeset="110857021"
5    |  timestamp="2021-09-07T12:07:20Z"
6    |  user="Martials"
7    |  uid="11927357"
8    |  lat="60.3690739"
9    |  lon="5.3510373">
10   <tag k="addr:city" v="Bergen"/>
11   <tag k="addr:housenumber" v="28"/>
12   <tag k="addr:postcode" v="5063"/>
13   <tag k="addr:street" v="Inndalsveien"/>
14   <tag k="amenity" v="library" />
15   <tag k="email" v="biblioteket@hib.no"/>
16   <tag k="internet_access" v="yes"/>
17   <tag k="name" v="Kronstad bibliotek"/>
18   <tag k="website" v="https://www.hvl.no/bibliotek"/>
19  </node>
```

Figur 6 - Eksempel på en node

#### 4.4.1.2 Tags

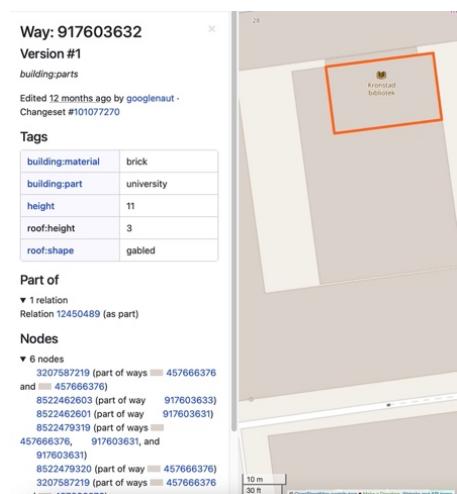
De tre elementene, Node, Way og Relation bruker alle tags til å lagre det meste av informasjon som ikke er standardisert for alle elementer av samme type. Tags inkluderer all annen informasjon om elementet i en «nøkkel og verdi»-tabell, med tekststrenger. Et eksempel på en slik tabell kan være som følger.

addr:city	Bergen
addr:housenumber	28
addr:postcode	5063
addr:street	Inndalsveien

Tabell 1 - Eksempel med noen «nøkkel og verdi» data

#### 4.4.1.3 Way

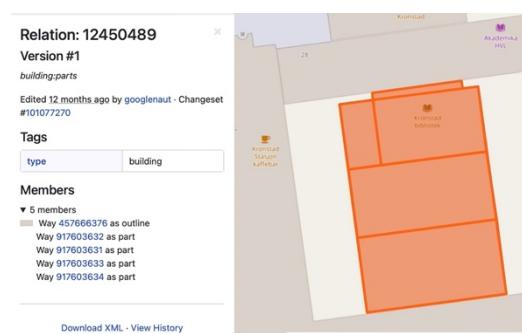
Elementet Way knytter sammen flere Noder til et lineært objekt eller et avgrenset område. Eksempler på dette kan være en vei, en elv eller et hus. Way knytter lineære linjer mellom nodene, som f.eks hvor en elv svinger, eller hjørnet på et hus. Way inneholder en liste av Nodes og kan også inneholde tags med mer informasjon om elementet. I et kartprogram vil dette gjerne utgjøre et polygon\*. Da har rekkefølgen på nodene noe å si for hvordan figuren blir tegnet av programmet.



Figur 7 - «Way» fra OpenStreetMap [30]

#### 4.4.1.4 Relation

Elementet Relation brukes som et element som binder sammen flere objekter som del av noe større. Relation kan inneholde en eller flere andre Relation(s), Way(s) og eller Node(s), og kan ha egne Tags.



Figur 8 - «Relation» fra OpenStreetMap [30]

#### 4.4.2 Viktige funksjoner

Det er flere andre funksjoner som definerer OpenStreetMap. Siden prosjektet baserer seg på åpen data, har alle brukere muligheter for å registrere seg for så å bidra med data til OpenStreetMap for å gjøre tjenesten bedre. Endringer kan spores igjennom versjonshistorikken og det er tilgjengelig informasjon om hvem som foreslo hvilke endringer og når. OpenStreetMap legger også til rette for nedlastning av rådata fra tjenesten. Ved å bruke eksporteringsfunksjonen kan du definere et område av kartet og laste dette ned. Data blir eksportert som en «.osm» fil og inneholder rådata i XML formatet. [Se figur 6.](#)

## 5 Realisering av valgt løsning

Oppgaven som ble gitt oss (se kapittel 1.3 og vedlegg; «BO22EB-16 Bruk av OpenStreetMap») var ganske åpen i formuleringen, så en av tingene vi måtte ta stilling til var å finne ut hvordan vi ønsket å løse oppgaven. Som beskrevet i «Hovedidé for løsning» (se kapittel 1.4) kom vi fram til en plan som vi synes dannet et godt grunnlag.

Basert på grunnlaget bestemte vi oss for plattform og verktøy for vår løsning. Siden en av målsettingene våre var at programmet skal kunne være en referanse for andre studenter med samme studieløp som oss, så ble det naturlig å velge plattform og verktøy som vi kjenner fra pensum.

### 5.1 Utviklerverktøy

Med det valget, å demonstrere verktøyene vi kjente fra pensum, ble det raskt avgjort hvilke verktøy vi ville bruke.

#### 5.1.1 Plattform, språk, IDE, utviklerplattform og rammeverk

Windows, C#, Visual Studio, og .NET er godt kjent fra tidligere pensum og det er verktøy vi er komfortable med å bruke til å løse oppgaver. Selv om vi har vurderte andre rammeverk for grafisk oppbygging av vårt program, så var det bare Windows Forms som var kjent fra tidligere pensum. (Se kapittel 3.1.4) Dermed var det bare Windows Forms som ville la programmet virke som referanse for andre studenter med samme utdanningsløp.

### 5.2 Database (Lagring av data)

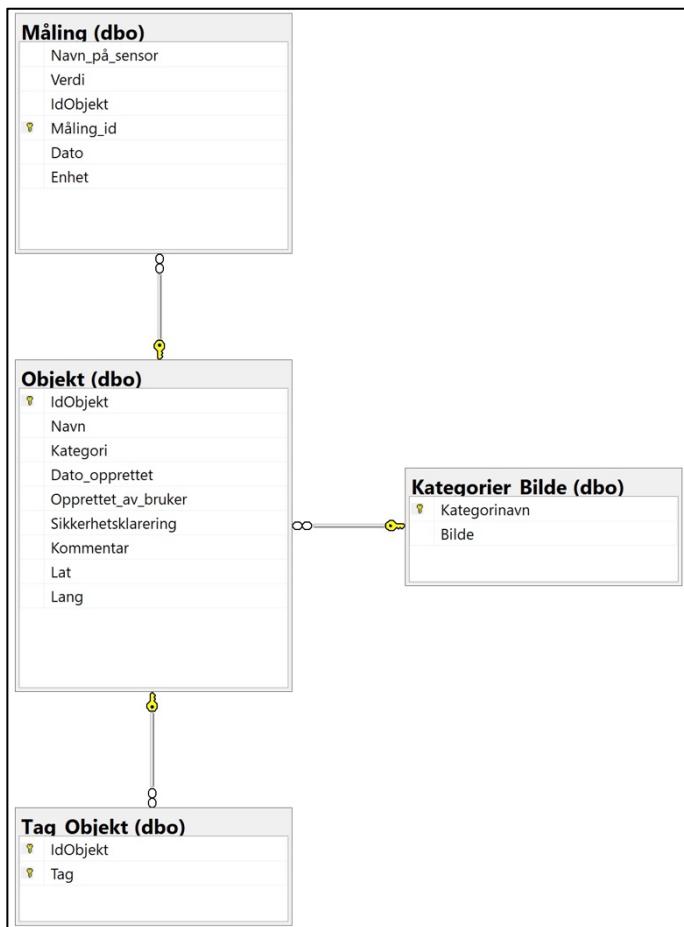
For lagring av data ønsket vi at all nødvendig data skulle være tilgjengelig for alle brukere og ikke være begrenset av en lokal fil e.l. som ikke nødvendigvis er synkronisert med andre brukere. Vi valgte derfor å opprette en ekstern database. Siden vi har noe erfaring med SQL, valgte vi å gå for en SQL-server løsning. [33] Utenom all felles data som er lagret på SQL-serveren så har vi enkelte variabler tilknyttet brukerens ønske om programmets tilstand ved oppstart, som lagres i en lokal fil.

Ettersom høyskolen allerede har en avtale med Microsoft bestemte vi oss for å undersøke tilbudet hos dem. Behovet vårt for datalagring ble mer en dekket av serverene vi kan leie der, og som studenter har vi tilgang på 100 USD credits fra Microsoft Azure. Siden vi allerede er en del av systemet via skolens konto (epost og Active Directory [34]), og Azure SQL(Transact-SQL) er veldig likt PostgreSQL kjent fra pensum, så var det lett å velge denne løsningen. For ca. 5 USD i måneden fikk vi opprettet en SQL-server med 2GB lagring som vi kunne administrere sammen ved å knytte den opp mot våre skolekontoer. Siden serveren leveres som en skytjeneste, vil den alltid være tilgjengelig fra hvor som helst så lenge man har tilgang til internett.

Vi valgte å lage et SQL-script som kan fjerne gamle tabeller, opprette nye tabeller og fylle inn data nødvendig for å demonstrere vårt program, slik at vi enkelt kunne gjenopprett vår database til en kjent tilstand.

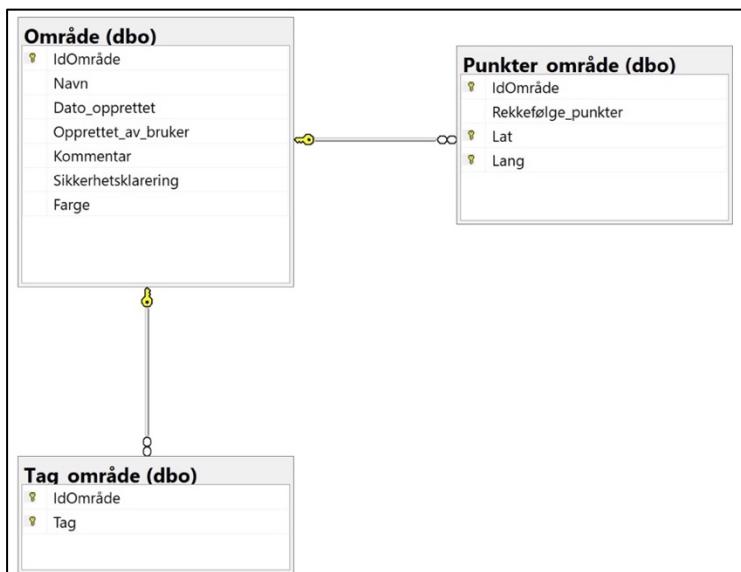
### 5.2.1 Database - design

Databasedesignet fulgte naturlig ut ifra de funksjonene vi ønsket at programmet vårt skulle ha. Nødvendige tabeller for objekter, områder, brukere, målinger m.m ble opprettet.



Figur 9 - ER-modell for «Objekt» og tilhørende tabeller

#### 5.2.1.1 Tabeller



Figur 10 - ER-modell for «Område» og tilhørende tabeller



Figur 11 - ER-modell for Bruker og for Log\_feil.

#### 5.2.1.1.1 Bruker

Tabellen «Bruker» inneholder informasjon om brukeren som personalia, passord og nødvendige variabler for godkjenning av brukere.

#### 5.2.1.1.2 Log\_feil

«Log\_feil» Inneholder informasjon om feilmeldinger som oppstår i programmet. Lagrer feilmeldinger med tilhørende klasse og metode.

#### 5.2.1.1.3 Objekt

«Objekt» er navnet vi har gitt på ressurspunkter som programmet viser. I tabellen «Objekt» lagrer vi informasjon om objekter som vises, som navn og posisjon, hvem som opprettet objektet, når objektet ble opprettet, sikkerhetsklarering, kategori og kommentarer.

#### 5.2.1.1.4 Kategorier\_Bilde

«Kategorier\_bilde» brukes til å lagre bilde (ikon) tilhørende et objekt som vises i programmet.

#### 5.2.1.1.5 Tag\_Objekt

«Tag\_Objekt» brukes til å lagre tags (nøkkelord) som knyttes til et objekt.

#### 5.2.1.1.6 Måling

Tabellen «Måling» lagrer data for objekter som fungerer som sensorer med registrering av måledata. Selv om lagring av måledata, og påfølgende uthenting av informasjonen, ikke er mulig i real-time med denne teknologien, så kan informasjonen som vises være tilstrekkelig fersk for å presenteres til en bruker via et kartprogram. Navn, verdi, enhet, dato og id lagres.

#### 5.2.1.1.7 Område

«Område» er navnet vi har gitt til polygon som vises i programmet, når ressursene strekker seg over et område (ikke et enkelt punkt). I tabellen «Område» lagrer vi informasjon om polygonet som vises, som Id, navn, hvem som opprettet objektet, når området ble opprettet, sikkerhetsklarering, farge og kommentarer.

#### 5.2.1.1.8 «Tag\_område

«Tag\_Område» brukes til å lagre tags (nøkkelord) som knyttes til et objekt.

#### 5.2.1.1.9 Punkter\_område

«Punkter\_område» brukes til å lagre punkter som utgjør et område, samt rekkefølgen punktene skal tegnes i.

### 5.2.1.2 T-SQL Prosedyrer

Vi bruker en rekke prosedyrer i vårt program, spesielt når brukeren skal skrive inn en tekst som blir en del av søkebegrepet.

Dette kan være mer effektivt for en SQL-server og kan bidra til sikkerhet. (Se kapittel 4.3.2.1.1), [35]

#### 5.2.1.3 T-SQL Funksjoner

I oppgaven vår bruker vi en prosedyre for å endre dato til å tilsvare vår tidssone, i stedet for tidssonen til serveren, når vi skal legge inn nå-verdier for dato i tabeller. [36]

#### 5.2.1.4 T-SQL Sekvenser

I vårt program brukes sekvenser for å gi en unik verdi (løpenummer) som Id, hvor vi kan være sikre på at verdien ikke har blitt brukt til Id tidligere. Id er primærnøkkelen for både objekt, område og målinger, og er derfor viktig at de får en ny og unik verdi som id. Hver av disse tabellene har sin egen sekvens som gir en ny id.

## 5.3 Biblioteker

I programmet bruker vi to tilleggsbiblioteker: GMAP og Dapper. GMAP brukes til å generere kartet, og Dapper til å kommunisere med vår database.

### 5.3.1 GMAP

GMAP [37] er et bibliotek som oppfyller det meste av ønsker vi har i forhold til generering av kart for vår oppgave. Vi bruker GMAP til å hente kartdata fra OpenStreetMap, og presentere det som et kart. I tillegg bruker vi GMAP til å hente ruteinformasjon imellom to posisjoner, som kan projiseres på kartet. Ulike funksjoner kan bruke ulike tilbydere. Pga. begrensninger med OpenStreetMap så bruker vi «Google maps» til å hente ruteinformasjon mellom to punkter og så presentere det på kartet fra OpenStreetMap.

GMAP sin mulighet til å plotte objekter og områder på et kart, ved hjelp av bredde- og lengdegrad, er noe vi benytter mye i vårt program. Til disse objektene knytter vi informasjon vi selv har definert og som vi ønsker å presentere.

Det at GMAP er tilpasset .NET og Windows Forms gjør det lett for oss å bruke biblioteket i vårt program. Et objekt av typen «GMap.NET.WindowsForms.GmapControl» kan plasseres i et form og settes opp ved at man definerer properties som:

```
«map.MapProvider = GMapProviders.OpenStreetMap;»
```

### 5.3.2 Dapper

Vi bruker Dapper i vårt program til å skrive spørninger til databasen direkte i SQL kode. Vi kan også kalle på lagrede prosedyrer. (Se kapittel 4.3.2.1)

I programmet vårt har vi opprettet klasser som tilsvarer de tabellene vi har i Databasen.

```
9  public class Bruker
10 {
11     public string Fornavn { get; set; }
12     public string Etternavn { get; set; }
13     public int Telefonnummer { get; set; }
14     public string Epost { get; set; }
15     public int Sikkerhetsklarering { get; set; }
16     public string Passord { get; set; }
17     public int Tallkode { get; set; }
18     public bool Godkjent { get; set; }
19     public bool Verifisert { get; set; }
20 }
```

Figur 12 - Klassen «Bruker»

Bruker (dbo)	
Fornavn	
Etternavn	
Telefonnummer	
Epost	
Sikkerhetsklarering	
Passord	
Tallkode	
Godkjent	
Verifisert	

Figur 13 - Databasetabellen «Bruker»

Når vi bruker Dapper til å utføre spørninger mot tabellen «Bruker» får vi da tilbake et objekt av typen «Bruker» eller en liste av objekter av typen «Bruker». [38]

For å se hvordan metodene blir strukturert i programmet vårt. (se kapittel 5.4.2.5 – Kommunikasjon med databasen) [39]

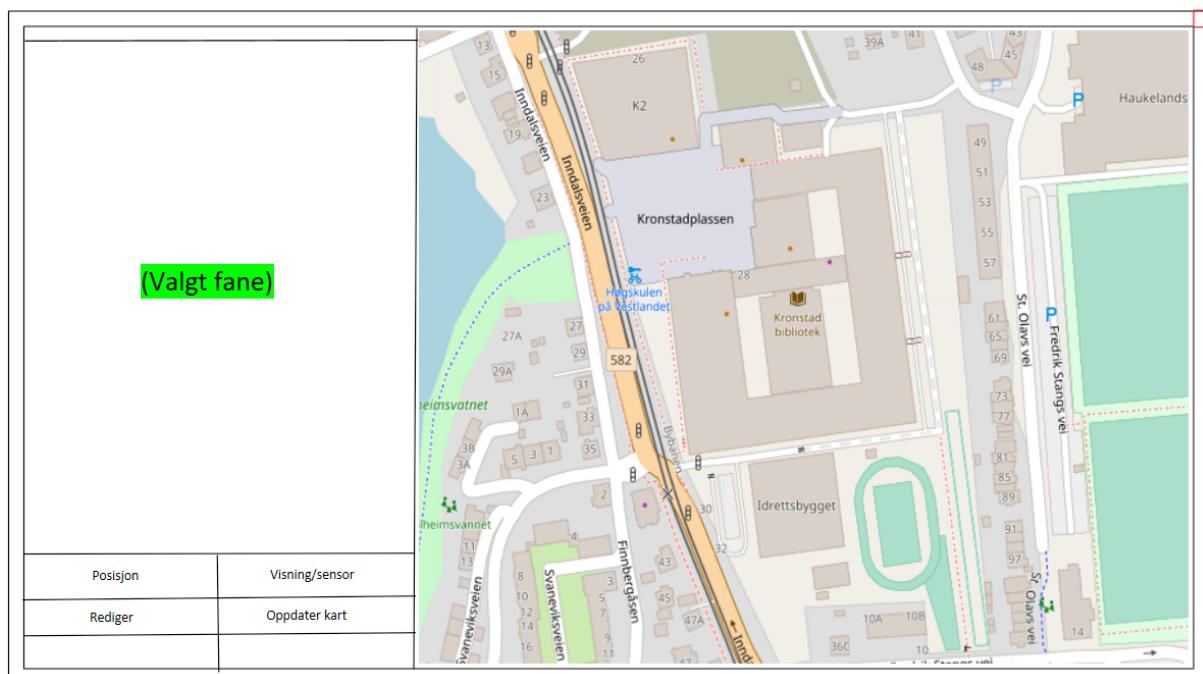
## 5.4 Utvikling av programvare

Med utgangspunkt i de valgte utviklerverktøyene (se kapittel 5.1 – Utviklerverktøy) opprettet vi et nytt prosjekt av typen «Windows Forms App (.NET Framework)» og koblet prosjektet mot GitHub som versjonskontroll.

### 5.4.1 Brukergrensesnitt

#### 5.4.1.1 Første utkast

Etter den første veiledningstimen tegnet vi en skisse over hvordan vi så for oss programmet med den informasjonen vi fikk da, se figur 14. Den inneholder en grov beskrivelse av det vi så for oss, og hadde med ønskede egenskaper vi ville at programmet skulle ha. Når vi begynte å jobbe mer med innholdet på programmet, innså vi fort at det ville være for lite plass til å få alle funksjonene inn i et vindu. Derfor introduserte vi flere vinduer som man kan åpne fra hovedvinduet.



Figur 14 - Tidlig skisse av programmet

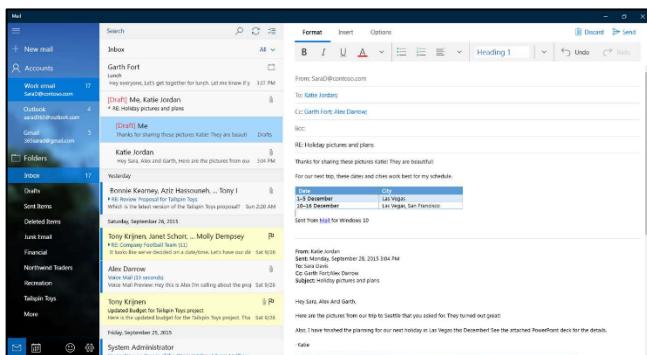
#### 5.4.1.2 Undersøkelse av eksisterende løsninger

Vi ønsket at brukergrensesnittet skulle være brukervennlig, samtidig som designet skulle være moderne og behagelig å se på. Derfor begynte vi å studere andre desktop applikasjoner som vi bruker mye. Det vi undersøkte var hvordan de organiserte forskjellige faner og funksjoner innad i appen. Det var ett design som gikk igjen, og som ble brukt av flere store og mye brukte apper, se figur 15.

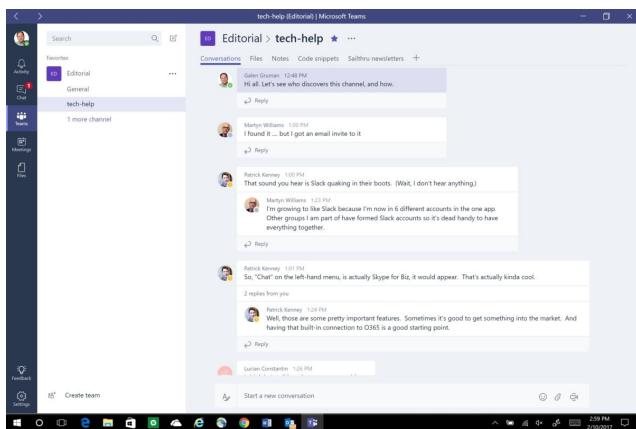


**Figur 15 - Vanlig design mønster**

Mønsteret som gikk igjen var at det har tre seksjoner. Seksjonene kunne variere i størrelse men inneholdt stort sett det samme. Seksjon 1 inneholdt normalt sett bare ikoner, men hadde noen ganger et beskrivende navn også. Seksjon 1 blir oftest brukt til å skifte mellom de forskjellige fanene man har tilgjengelig, ved hjelp av musetrykk på ikonene. Seksjon 2 viste alternativene i forhold til den valgte fanen i seksjon 1. Seksjon 3 var der fokuset til brukeren var mest og der appens funksjoner og oppgaver oftest ble utført. Noen av apper som bruker denne måtte å organisere på er Windows Mail [40], Teams [41] og Discord [42].



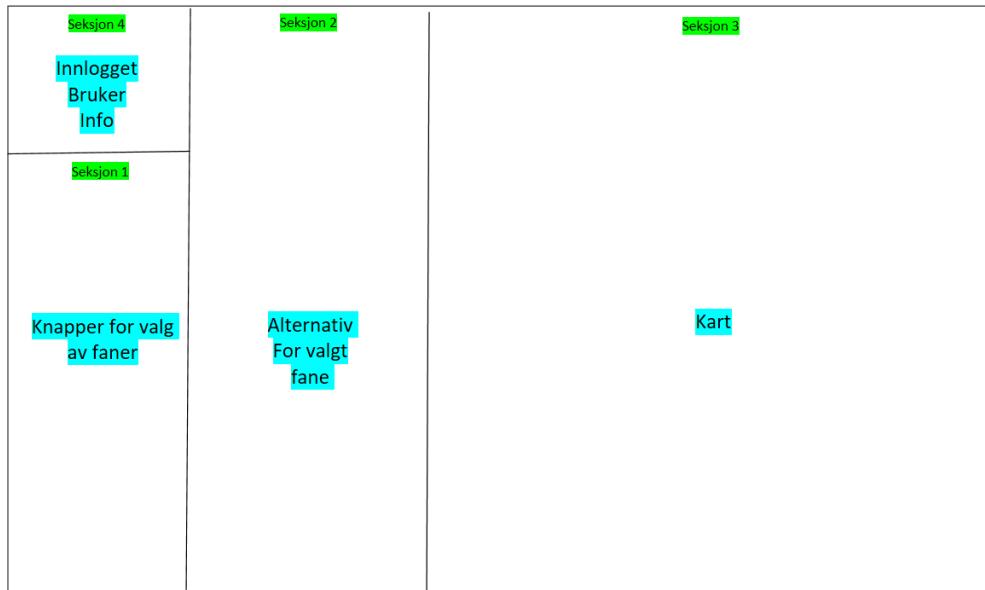
**Figur 16 - Windows Mail desktop applikasjon**



Figur 17 - Teams desktop applikasjon

#### 5.4.1.3 Vårt design

Vi gikk for et lignende design og måtte å organisere appen vår på, se figur 18. Dette gjorde vi for at brukeren skulle føle at appen er kjent, selv første gang personen bruker den. Vi selv syns dette er en fleksibel og oversiktlig måtte å organisere ting på. Hvis man senere skulle ønske å tilføye nye funksjoner eller faner i programmet, vil det være enkelt å legge til. Man kan enkelt legge til en ny knapp i seksjon 1 som inneholder funksjonen eller den nye fanen. Mye av grunnen til av vi valgte å designe programmet slik, er nettopp denne fleksibiliteten som vi fikk.



Figur 18 - Mal for vinduer i programmet

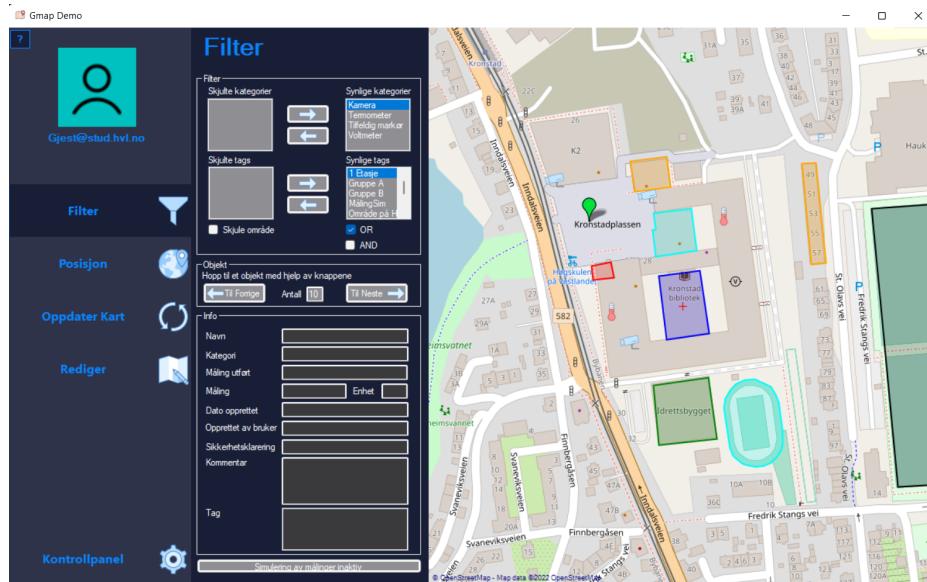
Når skissen var på plass, begynte vi å lete på YouTube [43] for å se om noen allerede hadde laget noe lignende. Dette gjorde vi for å få inspirasjon til mulige løsninger. Vi fant ganske fort en video [44] som hadde laget et lignende design, men hadde flere mangler i forhold til det vi ønsket. Vi bestemte oss for å ta utgangspunkt i det vi likte fra dette eksemplet. Mye av brukergrensesnittet vi laget er derfor inspirert av videoen, men vi gjorde en del endringer.

I planleggings fasen av prosjektet diskuterte vi om vi skulle lage to separate program, et for presentasjon av dataene og et annet for å legge til og redigering dataene. Vi valgte å legge alt i et program. Grunnen til dette er hovedsakelig at det blir enklere å jobbe med prosjektet siden man har alt på en plass istedenfor i to forskjellige prosjekter. En annen grunn er at det blir mer oversiktlig for en eventuell bruker/testperson. Skulle dette programmet ut til en kunde, ville vi selvsagt ha undersøkt hvordan de ønsket å ha det.

Slik programmet er nå har vi tre forskjellige vinduer, og de er: visning, redigering og kontrollpanel. Her er en kort forklaring på hva det forskjellige vinduene gjør og hvordan de ser ut:

#### 5.4.1.3.1 Visning

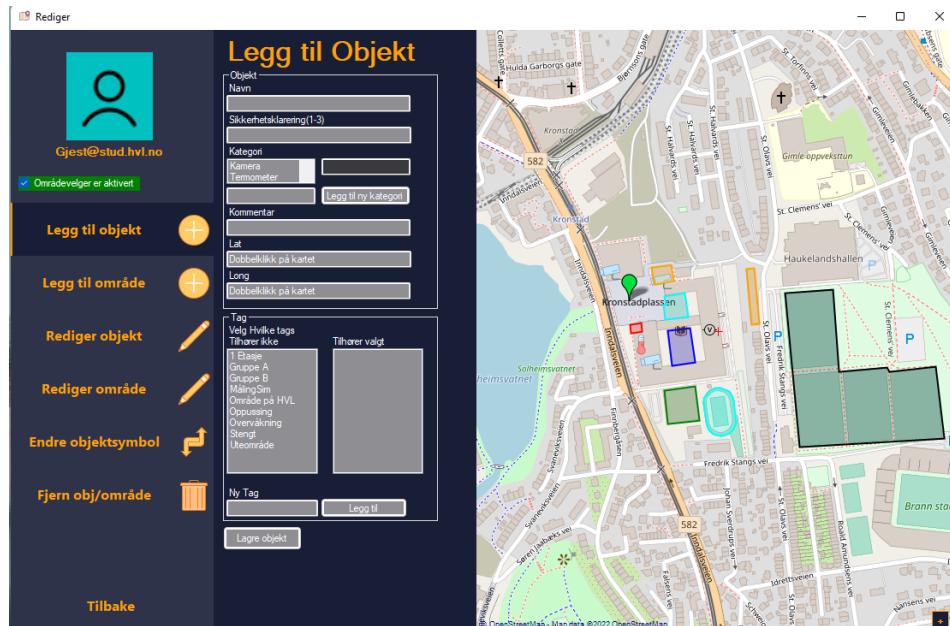
Når man logger inn kommer man inn i visningsvinduet, se figur 19. Her har man mulighet til å flytte kartet til ønsket posisjon enten ved hjelp av å bevege kartet, eller å søke. I tillegg blir dataene som er lagret i databasen vist, og de kan filtreres på ønsket måte. Hvordan filtret fungerer snakker vi mer om i delkapittelet «[5.6 Filter](#)». Man har også mulighet til å se informasjon om ressursen som blir vist.



Figur 19 - Visningsvinduet

#### 5.4.1.3.2 Redigering

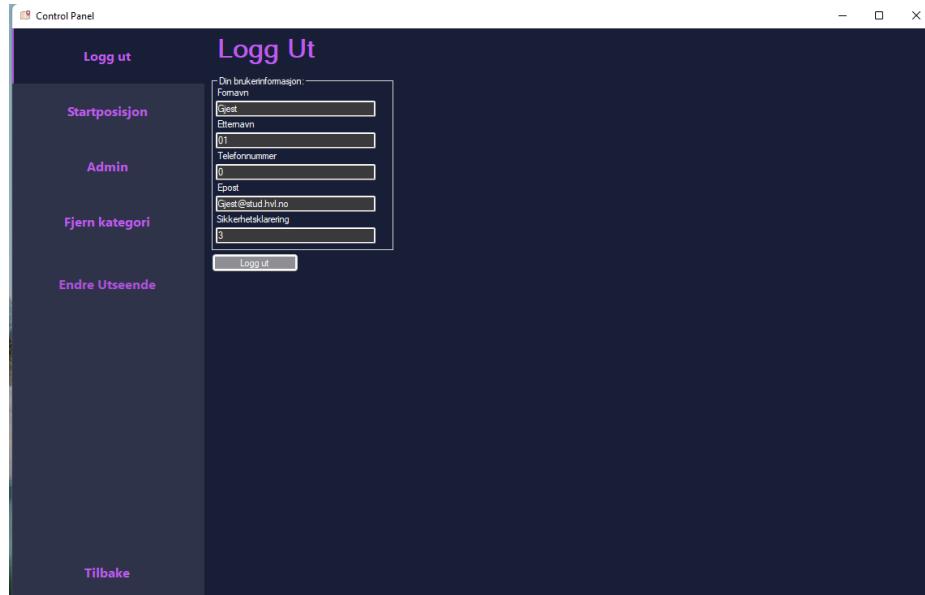
For å åpne redigeringsvinduet må man trykke på knappen «Rediger» i visningsvinduet. I vinduet «Rediger» kan man redigere hva som blir vist på kartet av ekstra ressurser. Man kan legge til, endre og fjerne ressurser på kartet. I tillegg kan man endre hvordan objekter blir presentert på kartet. Slik kan objektene bli presentert med ønskede ikoner, og ikke bare en markør.



Figur 20 - Redigeringsvinduet

#### 5.4.1.3.3 Kontrollpanel

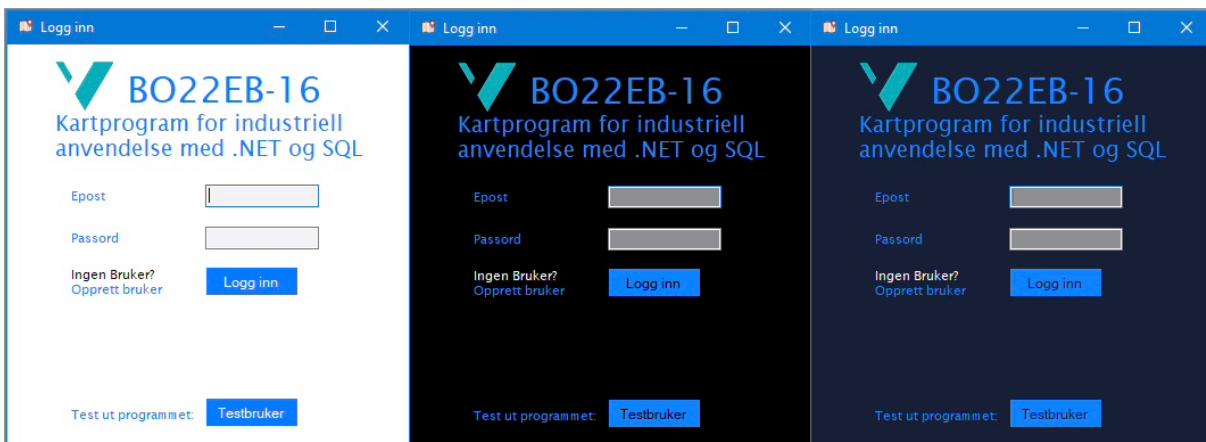
For å åpne kontrollpanelvinduet må man trykke på knappen «Control Panel» i visningsvinduet. Her har man mulighet for å endre personlig preferanser. Endringer som f.eks startposisjonen til kartet når programmet starter opp og fargetema. Andre funksjoner som man kan finne i dette vinduet er admin fanen, men denne går vi mer innpå i delkapittelet «[5.5.4 Admin Kontroll](#)».



*Figur 21 - Kontrollpanelvinduet*

#### 5.4.1.4 Farge tema

Først brukte vi samme fargevalg som videoen fra Youtube [44], og tenkte at dette skulle være vårt fargevalg. Senere ønsket vi fleksibilitet rundt fargevalget. Dermed introdusert vi muligheten til å endre fargetema. Nå har vi følgende temaer: dark theme, light theme og original theme. Også her kan man enkelt legge til flere alternativ i koden om man skulle ønske det. Valgt fargetema blir lagret i en fil på hver enkelt enhet, slik at hver enhet kan ha forskjellig. Alternativet hadde vært å lagre det som en attributt i «bruker» tabellen i databasen. I figur 22 viser vi hvordan innloggingsvinduet ser ut i de forskjellige temaene.



*Figur 22 - «Light Theme», «Dark Theme», «Original Theme»*

Muligheten til å enkelt endre fargetema kan være et nyttig salgstriks når man skal selge et program til en bedrift. Da har man muligheten til å legge til bedriftens farger og logoer i programmet for at det skal se tiltalende ut. [45]

### **5.4.2 Programkode**

Når designet og GUI-komponentene begynte å komme på plass, og vi var fornøyde med hvordan ting ble visuelt, begynte vi med backend-delen av programmet. Prosjektet inneholder en del kode, og derfor går vi ikke inn på spesifikke detaljer rundt den. Videre i delkapittelet skriver vi generelt om hvordan vi jobbet og tankegangen under skriving av koden. Vi nevner også hvordan vi jobbet med informasjonen fra databasen.

#### **5.4.2.1 Organisering**

Etter hvert som vi hadde skrevet litt kode og laget flere nye klasser, ble det vanskelig å holde oversikt over hvilken del av programmet de hørte til. Som en konsekvens av dette bestemte vi oss for å organisere klassene og formene inn i en mappestruktur, i tillegg til bruk av beskrivende navn. Slik fikk vi bedre oversikt, og tiden vi brukte på leting gikk betraktelig ned. Dette var også med tanke på å gjøre koden mer oversiktlig for andre som ønsket å bruke koden vår som referanse til sine prosjekter.

#### **5.4.2.2 Leselig kode**

Under kodingen av programmet fokusert vi på å gjøre koden så leselig som mulig. Som følger av dette ble det brukte lengre tid på navngivning og kommentering av koden enn vi vanligvis ville gjort. Vi ønsket at navnene på klassene, metodene og variablene skulle gi mening og være forklarende. Programmet inneholder også en del kommentar. Noen kommentar er kun for å forklare hva en enkel funksjon gjør, mens andre forklarer hva hele metoden gjør. Med hjelp av navnene og kommentarene vil en annen person lettere kunne forstå hva koden gjør. Vi føler at mye av koden er skrevet på en forståelig måte som ikke trenger kommentarer, men vi ønsket å legge ekstra fokus på forståelighet.

#### **5.4.2.3 Klasser og metoder**

Vi jobbet etter tanken om at de metodene vi opprettet skulle være så konkrete som mulig, og helst kun gjøre en ting. Slik at metodene kunne bli brukt på nytt flere steder i programmet. For metodene var det også viktig med gode navn, slik at man er sikker på hva metoden gjør uten å måtte sjekke koden.

Flere av metodene hadde en logisk plassering i en klasse. For eksempel så har vi en «kart» klasse, som inneholder alle metodene som har med kartet å gjøre. Videre hadde vi andre som var vanskeligere å plassere, siden metodene kunne blir brukt av flere forskjellige klasser. Da prøvde vi å plassere dem etter beste evne i en passende klasse.

De fleste klassene som er opprettet inneholder flere metoder. Det fleste metodene i klassene er statiske. Når metodene er statiske trenger vi ikke å opprette en instanse av klassen for å bruke metodene. Dette sparar oss for noen linjer med kode, og i tillegg ser koden ryddigere ut.

#### **5.4.2.4 Rutine rundt metoder**

Når vi jobbet med metoder vi skulle opprette, hadde vi en fast arbeidsrutine. Første steg var å få koden til å gjøre oppgaven. Når man hadde fått koden til å virke skulle man prøve å forbedre koden med å både gjøre den mer effektiv og på færre linjer. Her brukte vi ofte nettsider som stackoverflow [46] for å finne de beste løsningene. Når alle trinnene var gjennomført, kunne den andre på gruppen se igjennom koden for å se om han så noen forbedringsmuligheter.

Flere av de første metodene som vi skrev, måtte vi endre på eller skrive på nytt, ettersom vi fikk mer erfaring og fant bedre løsninger i programmet.

#### **5.4.2.5 Kommunikasjon med databasen**

For hver tabell i databasen har vi to tilhørende klasser i programmet. Den ene klassen har alle attributtene til den aktuelle tabellen som tilgangsmedlemmer\*. Den andre er for å kommunisere med databasen. Når vi skal hente ned et noe fra databasen blir begge klassene brukt. Et eksempel på hvordan programmet henter en spesifikk bruker ved hjelp av databasekommunikasjonsklassen «DBComBruker» og klassen med alle attributtene som tilgangsmedlemmer for «Bruker», kan være:

Først vil DBComBruker bruke en av sine metoder for å hente brukeren og dens attributter. Videre bruker den klassen Bruker, for å lagre Brukeren. For at den skal klare å lagre alle attributtene som databasen kommer med, må klassen Bruker sine tilgangsmedlemmer og kolonner i tabellen ha identiske navn. Grunnen til dette er for at Dapper skal vite hvilket attributt som skal lagres i hvilket tilgangsmedlem.

Når vi skal legge til noe nytt, lese, endre eller slette noe, er det disse klassene som kommuniserer med databasen som blir brukt. Når vi ønsker å bruke informasjonen i programmet er det klassene som har tilgangsmedlemmer som blir brukt.

#### **5.4.2.6 Målinger og simulering av målinger**

I programmet vårt bestemte vi oss for at objekter skal ha muligheten til å vise målinger. Tanken bak dette var at enkelte objekter kunne være lyssensorer, temperaturmålere, luftfuktighetsmålere, trafikktellere osv. Disse kan sende sine data direkte til databasen via f.eks. mikrokontrollere. Ettersom vårt prosjekt ikke har noen fysiske komponenter la vi inn en funksjon i programmet som simulerer målinger for to objekter ved å sende tilfeldige nye verdier til databasen. Intervallet er satt til 10 sekunder ved hjelp av en «System.Threading.Timer» funksjon. [47]

## 5.5 Sikkerhet

Med tanke på at programmet skal ha industriell anvendelse er det naturlig å legge vekt på sikkerhet. Her har vi gjort flere tiltak for å få applikasjonen så sikker som mulig.

### 5.5.1 Brannmur

Azure sin database har en brannmur som kun tillater IP-adresser som er på whitelist\*. Dvs. hvis IP-adressen til enheten ikke er på whitelisten vil ikke databasen kommunisere med den aktuelle enheten. I tillegg har vi en test når programmet startet. Testen sjekker om programmet klarer å opprette kontakt med databasen, og viss den klarer det er IP-adressen på whitelisten. Hvis testen feiler, gir programmet feilmeldingen og avslutter.

Det er mulig å sette IP-range for det aktuelle nettverket som appen skal bli brukt i. Under utviklingen av appen gjorde vi dette for deler av skolen sitt nettverk, slik at vi ikke trengte å whitelist IP-adressen vår hver gang vi jobbet på skolen sitt område.

### 5.5.2 Sikkerhetsklarering

Innad i programmet ønsket vi ha forskjellige sikkerhetsgrader, slik at selv om man har tilgang til programmet har man ikke adgang til alle ressurser som er lagret. Vi definerte derfor en sikkerhetsklarering i programmet hvor brukere og ressurser kan ha en klarering fra 1-3. Tallrekken kan endres eller utvides. Alle objekter og områder må ha en sikkerhetsklarering, brukere har også denne attributten. Brukere kan bare se og jobbe med ressurser som er lik eller lavere enn brukerens sikkerhetsklarering.

### 5.5.3 Bruker

For å bruke programmet må man ha en bruker. Dette kan man opprette inne i programmet når innloggingsvinduet kommer opp. Når man oppretter bruker vil man få beskjed om at de må vente til en administrator har godkjent brukeren. Hvis man blir godkjent vil man motta en epost med en verifiseringskode som man må fylle inn når man prøver å logg inn. Det er tre grunner for at vi velger å sende koden på epost:

1. Sjekke om epost adressen er skrevet riktig.
2. Personen er den han/hun utgir seg for å være.
3. For å ha en konkret måtte å kontakte brukeren på, om det skulle oppstå spørsmål rundt det person har lagt til.

### 5.5.4 Admin kontroll

Som tidligere nevnt må en ny bruker bli godkjent av en administrator. Dette gjorde vi for at de som er ansvarlig for applikasjonen skal ha full kontroll på hvem som har tilgang til appen, og dermed ha muligheten til å godkjenne og avslå brukere, se figur 23). Dette krever maks sikkerhetsklarering. Velger man å godkjenne brukere som er i listen «Venter på godkjenning», vil de komme på listen over personer som ikke er verifisert enda, se figur 24. Her kan man også se verifiseringskoden som ble sendt på epost. Vi ønsket å vise den om det skulle oppstå komplikasjoner med eposten. Anbefalingen er fremdeles at den aktuelle brukeren bruker koden som ble sendt på epost, av grunnene som ble nevnt i delkapittel «6.5.3 bruker».



Figur 23 - Admin: Brukere som venter på godkjenning.



Figur 24 - Admin: Brukere som ikke har skrevet inn verifiseringskode.

En annen mulighet innad i programmet er å oppgradere eller nedgradere brukers sikkerhetsklarering. I tillegg har man mulighet til å fjerne brukere, se figur 25. Kravende for de forskjellige mulighetene er noe vi har bestemt, og er mulig å endre. For oppgradering, nedgradering og fjerning gjelder følgende krav:

Oppgradere:

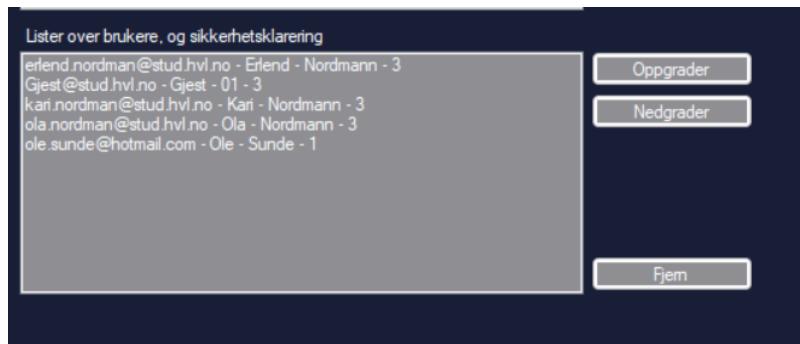
- Den som skal oppgradere en bruker må ha høyre sikkerhetsklarering enn brukeren. Det vil si at man kan oppgradere en annen bruker til sin egen sikkerhetsklarering.
- Kan ikke oppgradere seg selv.

Nedgradering:

- Den som skal nedgradere en bruker må ha høyre sikkerhetsklarering enn brukeren.
- Kan ikke nedgradere om brukeren som allerede har laveste sikkerhetsklarering, altså 1.
- Kan nedgradere seg selv.  
NB: Hvis det kun er en med maks sikkerhetsklarering og personen prøver å nedgradere seg selv, får personen ikke tillatelse til dette. Personen må da oppgradere enn annen før han kan nedgradere seg selv.

Fjerning:

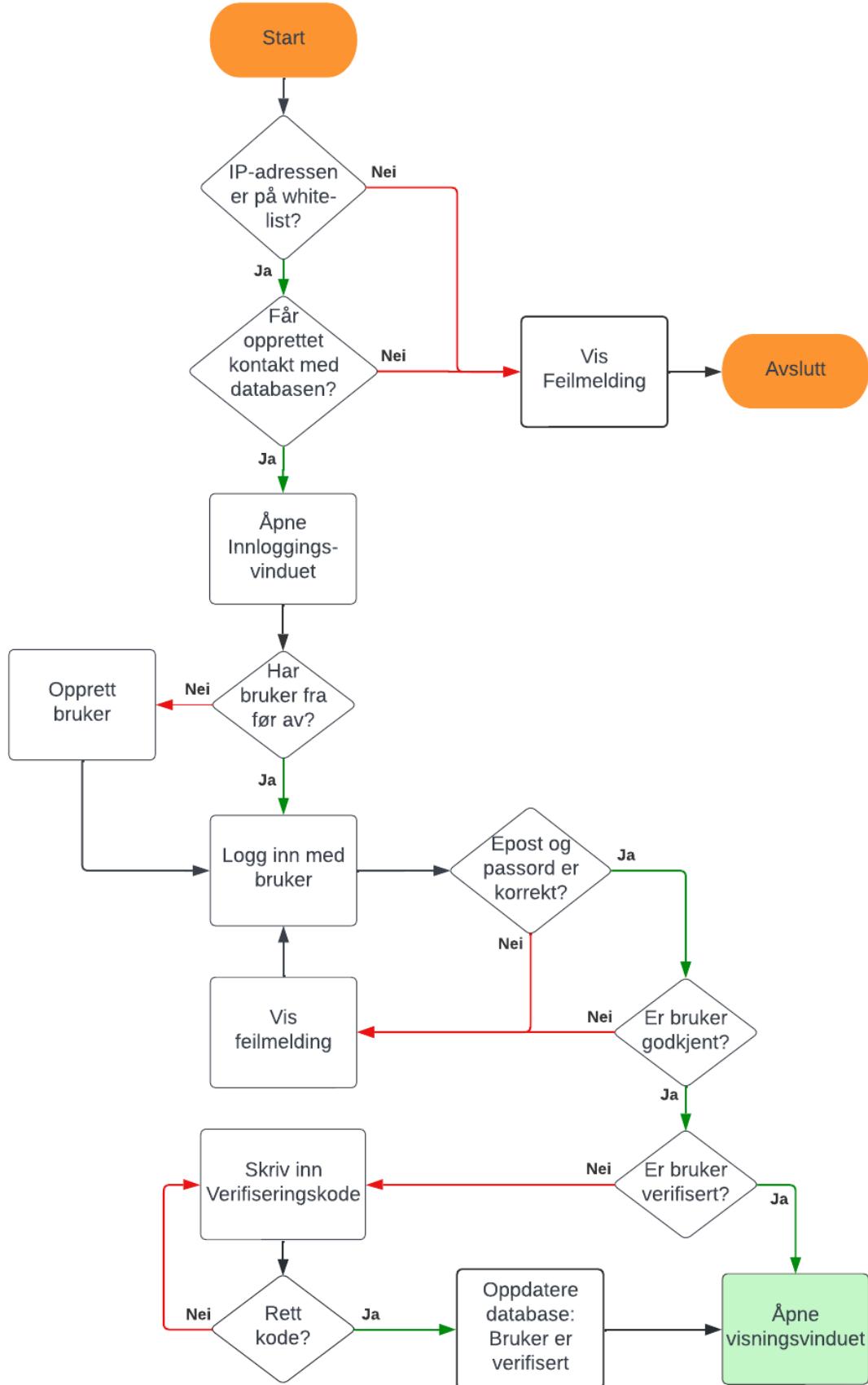
- Man må være maks sikkerhetsklarering for å kunne fjerne andre brukere.
- Den som skal fjerne en bruker må ha høyre sikkerhetsklarering enn brukeren.
- Kan ikke fjerne seg selv.



Figur 25 - Admin: Oppgradering og nedgradering av sikkerhetsklarering, og fjerning av brukere.

### 5.5.5 Starte programmet

For å starte programmet må man igjennom en del nødvendige sjekker/tester. I figuren 26, kan man se hva som kreves for å kunne åpne visnings vinduet. Alle sjekkene og testene er beskrevet tidligere i dette delkapittelet. Alle disse tiltakene er der for å gjøre at den informasjonen som er lagt inn i programmet er så sikker som mulig. Dermed har ikke hvem som helst mulighet til å se informasjonen som er lagret.



Figur 26 - Flytskjema for å åpne visningsvinduet.

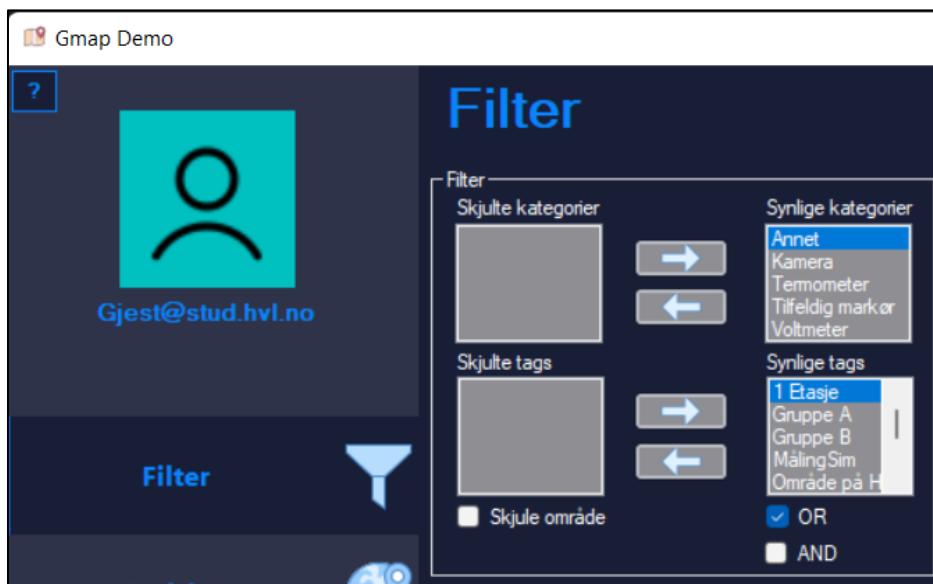
## 5.6 Filter

Vi ønsket at brukeren skulle ha mulighet til å filtrere hva som blir vist på kartet. Derfor introduserte vi en filtreringsfane, se figur 27). Hovedfokuset i programmet er objekter/sensorer, og derfor kan man spesifisere filteret mer nøyaktig for objekter enn områder. Hvilke filter objekter og områder går igjennom kan du se i tabell 2. Filtrene fungerer på samme måte for både objekter og områder.

Videre i delkapitlet vil diagrammene fokusere på objekter, men filtrene som blir brukt på både objekter og områder har identisk virkemåte.

Rekkefølge	Filtrene objekter går igjennom:	Filtrene områder går igjennom:
1	Database-filter	Database-filter
2	Kategori-filter	Tag-filter
3	Tag-filter	

Tabell 2 - Filtreringsrekkefølge for vår kode.



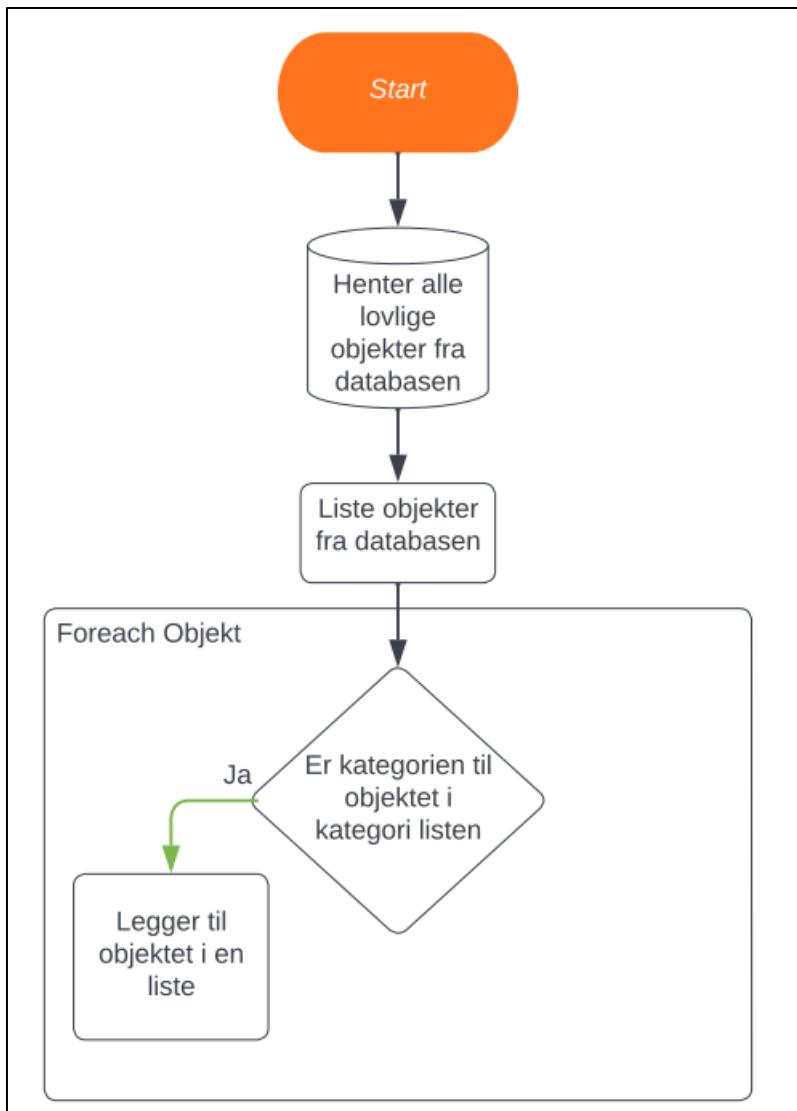
Figur 27 - Utklipp av filter i programmet.

### 5.6.1 Database-filter

Det første filteret finner man allerede i spørringen til databasen når man henter ressursene ned til programmet. Den henter den alle objektene eller områdene som har sikkerhetsklarering lik eller lavere enn det den innloggede brukeren har. Ressurser med høyre sikkerhetsklarering vil ikke bli hentet ut av databasen.

### 5.6.2 Kategori-filter

Kategori-filteret brukes kun på objekter, siden områder ikke har attributtet «kategori». Når programmet har hentet de objektene som den innloggede brukeren har tillatelse til å se, sjekker det hvilke kategorier brukeren ønsker å ha på kartet. Da bruker den listen: «Synlige kategorier» (se figur 27) for å sjekke om objektene har kategorien sin i listen, se figur 28.



Figur 28 - Flytskjema for database og kategori-filter.

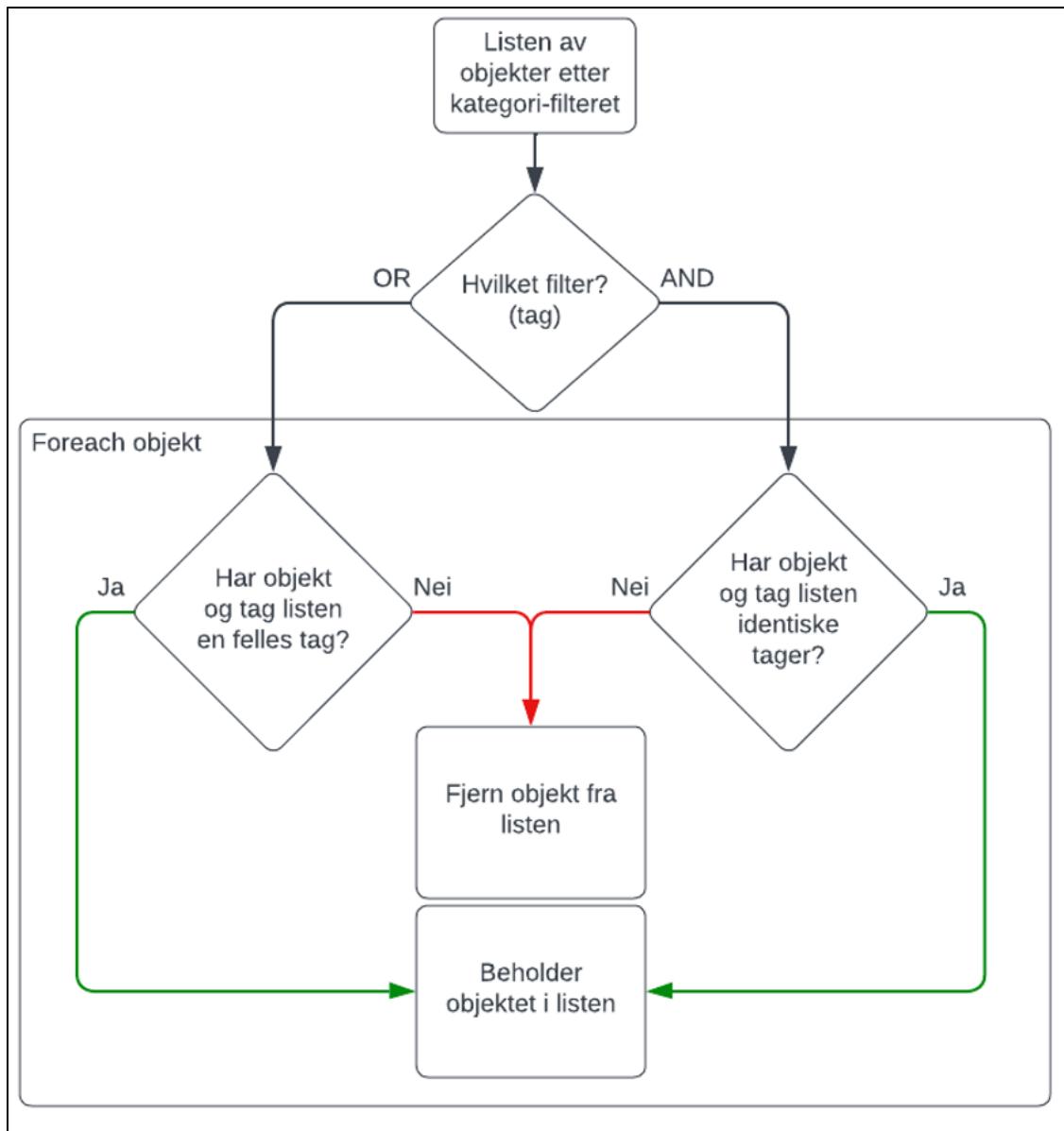
### 5.6.3 Tag-filter

Tag er det engelske ordet for merkelapp. I vårt program har vi opprettet «tags» som nøkkelord, som kan settes på ressurser og brukes for filtrering av hva som skal vises. Dette filteret gjelder for både objekter og områder. Dette er det siste filteret før ressursene vises på kartet. Den tar utgangspunkt i listen «Synlig tags» (se figur 27) og hvordan den filtrerer er brukerens valg. Valget er mellom «OR» eller «AND», og de finner man under «Synlig tags» listen. En av disse vil alltid være aktiv. De har forskjellige måtte å filtrere på, og de er som følger: (fokuserer på objekter)

**OR:** Hvis listen «synlig tags» og tagene knyttet til objektet har en felles tag, vil objektet gå igjennom filteret.

**AND:** Hvis listen «synlig tags» og tagene knyttet til objektet er identiske, altså at alle er like, vil objektet gå igjennom filteret.

De objektene som gikk igjennom filteret (se figur 29) og er i listen, vil bli presentert på kartet.



Figur 29 - Flytskjema for tag-filteret mot objekter (identisk for områder).

## 6 Testing

I dette kapitlet tar vi for oss hvordan vi testet programmet.

### 6.1 Testing underveis

Under utviklingen av programvaren, gjorde vi mange tester underveis. Vi testet alltid nye funksjoner og metoder når de ble lagt til. Det som ble testet da var om den klarte oppgaven som den skulle utføre og om den påvirket andre eksisterende funksjoner. Med jevne mellomrom tok vi en mer omfattende test der vi testet alle funksjonene programmet kunne utføre på daværende tidspunkt. Under testingen noterte vi ned det som ble oppdaget av feil, mangler og forbedringer, og fikset det etterpå. Da måtte vi igjen kontrollere at endringen fikset feilen/mangelen.

### 6.2 Logge feil

Nesten alle try-catch-blokkene i programmet inneholder en metode som logger catch-en, slik som i figur 30 og 31. Metoden logger feilen i databasen med følgende attributter: dato, hvilken klasse, hvilken metode, og feilmeldingen. Etter hvert som vi hadde gjennomført flere tester av programmet, sjekket vi databasen for å se hvilke feil som var blitt logget, se figur 32. Da oppdaget vi at flere metoder ikke var robuste nok og ofte feilet. Dermed kunne vi forsterke de metodene som databasen hadde i loggen, basert på feilmeldingene som var lagret.

```
catch (Exception feilmelding)
{
    DBComLog_feil.LogFeil(GetType().Name, System.Reflection.MethodBase.GetCurrentMethod().Name, feilmelding.Message);
}
```

Figur 30 - Logging for ikke-statiske metoder.

```
catch (Exception feilmelding)
{
    DBComLog_feil.LogFeil(typeof(FilBehandling).Name, System.Reflection.MethodBase.GetCurrentMethod().Name, feilmelding.Message);
}
```

Figur 31 - Logging for statiske metoder, her i klassen: «Filbehandling».

	Dato	Klasse	Metode	Feilmelding
2	2022-04-12 16.03.07.990	frm_R_LeggTilOmråde	btnLeggTilPunktIListe_Click	Inndatastrengen hadde ugyldig format.
3	2022-04-19 10.29.47.453	frmPosisjon	btnStart_Click	Input string was not in a correct format.
4	2022-04-19 11.50.42.590	Frm_S_StartPosisjon	Fylltekstfelt	Inndatastrengen hadde ugyldig format.
5	2022-04-27 16.14.58.700	frmPosisjon	btnSlutt_Click	Inndatastrengen hadde ugyldig format.
6	2022-04-27 16.15.00.190	frmPosisjon	btnStart_Click	Inndatastrengen hadde ugyldig format.
7	2022-04-27 16.15.01.147	frmPosisjon	btnStart_Click	Inndatastrengen hadde ugyldig format.
8	2022-04-27 16.15.04.940	frmPosisjon	btnStart_Click	Inndatastrengen hadde ugyldig format.
9	2022-04-27 16.17.38.943	frmPosisjon	btnSlutt_Click	Inndatastrengen hadde ugyldig format.
10	2022-05-02 16.16.18.707	frm_R_RedigerOmråde	btnLeggTilPunkt_Click	Inndatastrengen hadde ugyldig format.
11	2022-05-02 16.16.21.230	frm_R_RedigerOmråde	btnLeggTilPunkt_Click	Inndatastrengen hadde ugyldig format.

Figur 32 - Utklipp av feilloggen i databasen.

### 6.3 Brukervennlighetstest

Vi ville få personer med liten eller ingen forkunnskaper om programmet til å teste det ut. Dette ønsket vi for å teste brukervennligheten og for å finne feil og mangler. Det er ikke en uvanlig praksis å gi programmer ut til personer med liten eller ingen forkunnskaper om programmet, siden de oppdager oftere mangler og feil som den som laget programmet ikke har oppdaget [48]. Grunnen til dette er at de kan bruke ting på andre måter enn det utvikleren har tiltenkt og at de ikke vet hvordan alle ting fungerer, og derfor prøver seg fram. I tillegg ser man om de bruker ting på tiltenkt måtte.

Når programmet nærmet seg ferdig, rundt påsketider (17-25 april), fikk vi familiemedlemmer og venner til å teste programmet. Vi brukte den samme forhåndsskrevne testen på alle, testen kan du se i figur 33. Vi leste opp teksten uten at de fikk se teksten selv. Videre ga vi kun hint når de var på villspor og hadde prøvd en stund uten å komme noen vei. Vi prøvde å ikke si mer enn det som var nødvendig, men av og til er det mangler på kunnskap som er essensielt for programmet og dens bruk, som: «hva er en tag?» eller «hva menes med sikkerhetsklarering?». Dette er eksempler på spørsmål vi besvarte.

#### Test

##### Info om appen:

"Dette er en app som bruker OpenStreetMap sitt kart, og kan kartlegge forskjellige ting, som objekter og områder. Dvs At man kan legge til forskjellige ting på kartet som: kamera, termometere, osv. Disse tingene er kun synlig for det som har tilgang til appen og har bruker. "

Gi dem appen, og få dem til å logg inn, UTEN å bruke testbrukeren.  
Dvs: de må opprett bruker, og du må godkjenne den.

Når personen er logget inn:

Gi 2 min på å bli kjent med appen

##### Oppgaver:

1. Finne: Oslo Sentrum med hjelp av søk
2. Finn rute mellom Bergen sentrum og Åsane
3. Bruk filteret til å kun vise voltmeteret på kartet, som er på HVL
4. Flyttet kartet til der vi er nå
5. Legg til tv-en, der den ca. står

##### Info:

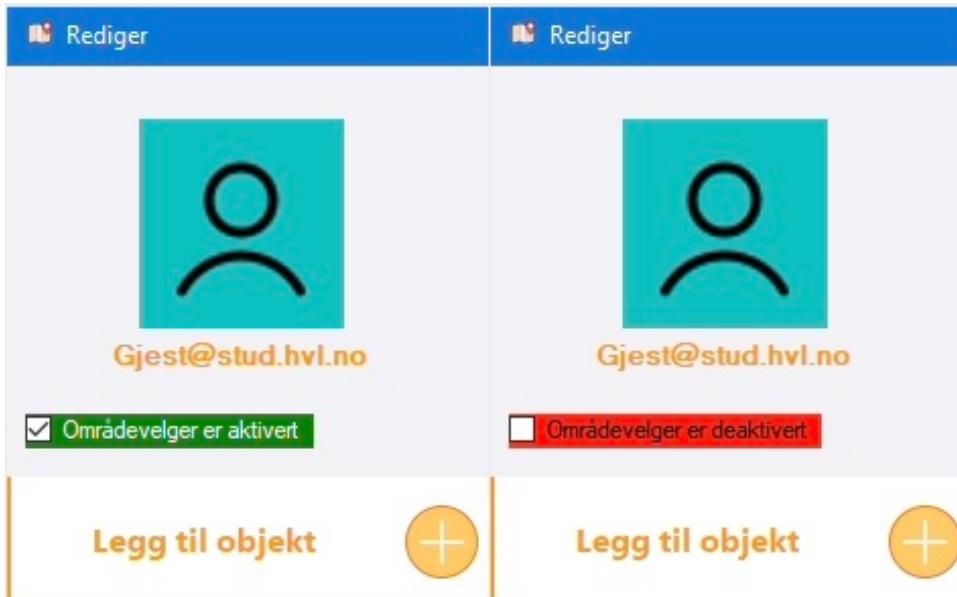
nå ser du at markøren du la til er grønn, men kan du få den til  
Å være en tv istedenfor en grønn markør slik som det er nå?

6. Bytte objektsymbol til tv
7. Lag et område som dekker huset vi befinner oss i nå, lagre
8. Tv-en som ble lagt til i oppgave 6 er nå flyttet ut av huset i "virkeligheten", kan du også flytte den på kartet?
9. Kan du endre fargen på området som ble laget over huset?(oppgave 8)
10. Jeg har plassert flere "tilfeldig markører" ut på kartet, hvor mange klarer du å finne?  
(Må oppgi land de befinner seg i)
11. Kan du simulere en måling på HVL? På det sensorene som de går på, må selv finne de.
12. Legge til et punkt/objekt inni det området du la til (over huset). (**Vansklig**)
13. Kan du fjerne alt du har lagt til?
14. Nå er det et farge "theme" kan du endre det til noe annet?
15. Logg ut

Figur 33 - Test for person uten forkunnskaper.

Basert på testingen ble det gjort en del små endringer, som å endre navn på noen av overskriftene siden de kunne være misvisende. I tillegg ble det lagt til mer informerende tekst flere plasser og verktøytips der det var nødvendig. En annen endring som ble gjort var at fanen som kan endre objektsymbol ble flyttet til redigeringsvinduet. Opprinnelig var denne plassert i kontrollpanelet, men når testpersonene skulle utføre oppgave 5 og 6 (se figur 33), fant de ikke fanen som kunne utføre oppgave 6 og ga opp til slutt. På spørsmålet hvor de synes den burde være, mente de at den burde være i redigeringsvinduet. På de testene som ble gjennomført etter flyttingen, hadde ingen problemer med å utføre oppgave 6.

Videre ble de fleste oppgavene gjennomført uten store problemer, med unntak av oppgave 12 (se figur 33). Denne oppgaven er ikke veldig intuitivt, fordi man treffer på et problem som krever en spesiell og spesifikk løsning. Problemet man møter på er: når man prøver å legge til et objekt eller punkt inni et område, hopper programmet til «Rediger område» fanen og legger inn informasjonen om området du klikket på. Når dette hoppet skjer, blir all informasjon du har fylt ut om objektet fjernet, og det må gjøre det på nytt. For å løse dette problemet må man trykke på «Områdevelger er aktivert» for å få den til å bli deaktivert. (Se figur 34 )



Figur 34 - Områdevelger i aktivert og deaktivert tilstand.

Når man har deaktivert områdevelgeren kan man plassere objekter og punkter inni eksisterende områder uten problemer. Områdevelgeren blir aktivert igjen når man legger til noe, lagrer en endring, eller åpner en ny fane. Selv om dette problemet er vanskelig å løse, klarte flertallet å løse oppgaven uten hint.

Generelt sett fikk vi gode tilbakemeldinger på programmet, og testpersonene synes at programmet var brukervennlig. Det var totalt ti personer som utførte testen. To personer før flyttingen av fanen der man kan bytte objektsymbol, og åtte etter.

## 7 Diskusjon

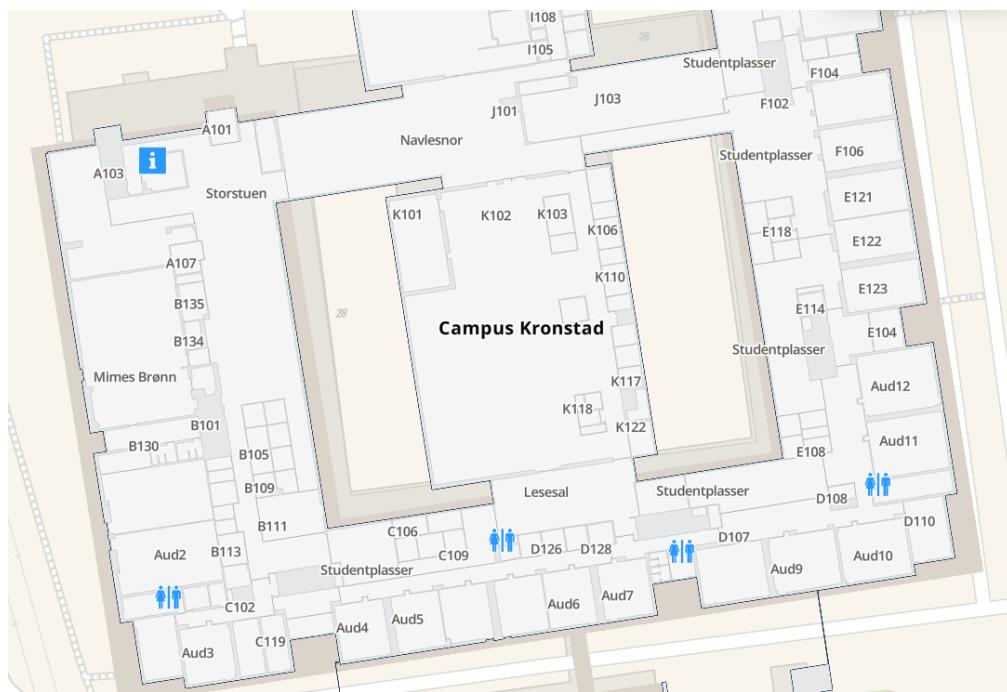
Det vil alltid være ting man i ettertid tenker at man kunne løst på en annen måte eller ting man ville gjort mer med, hvis tiden tillot det. Vi drøfter derfor noen forslag til videre forbedringer til programmet og reflekterer over ting vi har lært gjennom arbeidet.

### 7.1 Forslag til forbedring

Det finnes en uendelig lang liste med ting man kan gjøre dersom man har all tid i verden, men her forsøker vi å trekke fram nyttige og tydelige forbedringer som vi hadde prioritert dersom vi hadde ekstra tid.

#### 7.1.1 Forbedring 1 – Mer detaljerte kart

Et program som vårt vil bli bedre hvis man har et mer detaljert innendørs kart over den aktuelle bygningen, med detaljer som romfordeling og annen nyttig info. En eksisterende app som gjør det mulig å tegne kart med mer detaljer, er Mazemap [49]. Mazemap er en app som høgskolen bruker for å vise hvor de forskjellige rommene er, og veien til dem fra din posisjon. En forbedring ville være å få inn en løsning som kan legge til et mer detaljert innendørs kart, som viser romfordeling inn i vårt program. Gjerne slik mazemap gjør det. For sammenligning se figur 35 og 36.



Figur 35 - Kart over HVL, hentet fra Mazemap.



Figur 36 - Kart over HVL, hentet fra OpenStreetMap.

### 7.1.2 Forbedring 2 – Lage nettside med .ASP

Tidligere i rapporten (kapittel 3.1.3) snakket vi om å lage en mobil app. Dette ble ikke gjennomført siden vi måtte velge et bestemt operativsystemene å fokusere på. Dermed kunne en forbedring være å lage en nettside med hjelp av .ASP [50]. Det vil da være mulig å lage en app som kan bli brukt på tvers av de mest brukte operativsystemene (Microsoft Windows, Linux og MacOS). Hvis vi skulle gjennomføre dette måtte vi lage et nytt prosjekt. Mye av koden fra programmet vårt ville vært mulig å bruke om igjen, men mye måtte endres. I tillegg måtte vi designet det visuelle på nytt med hjelp av en html-fil, noe vi ikke har erfaring med. På grunn av gjenstående tid ble ikke dette gjennomført.

### 7.1.3 Forbedring 3 – Kryptering av kommunikasjon

En funksjon vi så på å legge til vårt program, men ikke fikk tid til å gjennomføre, var kryptering av data mellom server og program. Med tanke på at programmet blir laget for industriell bruk er det å kunne kryptere data absolutt av interesse for flere brukergrupper.

Det ligger i dag en kryptering gjennom «Transport Layer Security (Encryption-in-transit)» (TLS 1.2). Vi har ikke undersøkt i detalj hvor omfattende denne krypteringen er, eller hva som er potensielle svakheter og forbedringspunkter. Vi har heller ikke innført noen form for kryptering for «Data at rest» (forenklet; data som blir lagret på en enhet mellom/etter bruk). [51]

Innføring av forskjellige passende former for kryptering er en forbedring som kan tillegges vårt program.

### 7.1.4 Forbedring 4 – Innstillingsside før innlogging

En side (form) som kan brukes før programmet kobler til databasen, hvor du kan erstatte innloggingsstrengen slik at programmet kan koble til en annen database, uten at endringen må gjøres direkte i programkoden.

### 7.1.5 Forbedring 5 – Forbedre og samkjøre SQL-spørringene

Noen av SQL-spørringene\* kan tenkes at kunne slåes sammen med riktig bruk av variabler. En mulig forbedring er derfor å gå igjennom alle SQL-spørringene og se på hvordan de kan forbedres og hvilke av de som kan slåes sammen.

## 7.2 Kjente feil/bugs

For øyeblikket har vi ingen kjente bugs som vi ikke har utbedret, i vårt program.

Det er allikevel enkelte ting som vi ikke har tatt høyde for i vårt program som i spesielle tilfeller kan føre til at vårt program krasjer.

- Overflow av integer, f.eks. ved at vi har for mange objekter + områder.
- Overfylling av database.

Som nevnt i kapittel 7.1.3 finnes det også en del forbedringer som vi kan tilføre vårt program.

## 7.3 Ting vi har lært gjennom vårt arbeid med oppgaven

I tillegg til å bli mer fortrolig med .NET og Visual Studio er det et par nye ting vi har lært gjennom vårt arbeid med oppgaven.

### 7.3.1 GitHub

Vi har lært oss å bruke versjonskontroll ved hjelp av GitHub, og å utvikle et program sammen ved å laste opp oppdateringer til skyen.

### 7.3.2 Bibliotekene GMAP og Dapper

Vi har blitt kjent med bibliotekene GMAP og Dapper og har lært å bruke disse.

### 7.3.3 Azure og Transact-SQL

Vi har utforsket Microsoft Azure og har lært å sette opp en database i Azure systemet. Vi har også lært å utføre spørninger i Transact-SQL.

### 7.3.4 SQL Stored Procedures og functions

Vi har blitt introdusert for «Store procedures» og «Function» i Transact-SQL og lært å bruke disse.

### 7.3.5 BindingList

Vi har oppdaget BindingList som gir oss en toveis kommunikasjon mellom en liste og data f.eks. fra en listBox.

## 8 Konklusjon

Gjennom arbeidet med dette prosjektet har vi i gruppen tilegnet oss mange gode erfaringer. Vi har fått bedre kjennskap med bruk av forskjellige verktøy tilknyttet programmering, planlegging av prosjekter tilknyttet programmering og vi har fått mye praktisk erfaring med å programmere.

Det er vår oppfatning at prosjektet har oppfylt alle kravene. Vi har utredet bruk av OpenStreetMap (og relaterte kart-verktøy) i programmer, og dokumentert dette i rapporten. Vi har også laget et program som demonstrerer mange nyttige funksjoner og verktøy for studenter som jobber med programmering og har likt utgangspunkt som vi hadde etter 5. semester. Selv om programmet har funksjon knyttet til kart og kartverktøy, så kan eksemplene demonstrere nyttige funksjoner og verktøy for programmer med mange andre bruksområder.

Vi håper også at prosjektet kan bidra til å øke kvaliteten ytterligere, på fag som ELE301 - Industriell IT, som vi fikk mye av grunnlaget for oppgaven fra. Det er med tanke på dette at vi har fokusert på å gjøre koden oversiktlig og har jobbet med å skrive gode manualer for oppsett av systemet og verktøyene.

## Referanser

- [1] Høgskulen på Vestlandet, «Adis Hodzic - Høgskulen på Vestlandet,» [Internett]. Available: <https://www.hvl.no/person/?user=Adis.Hodzic>. [Funnet 21 02 2022].
- [2] A. Hodzic, *Bruk av OpenStreetMap*, Bergen: Høyskolen på Vestlandet, 2021.
- [3] GitHub, «GitHub,» GitHub, [Internett]. Available: <https://github.com/>. [Funnet 6 Mai 2022].
- [4] Git, «Git,» Microsoft, [Internett]. Available: <https://git-scm.com/>. [Funnet 6 Mai 2022].
- [5] TablePlus, «What is TSQL and how is it different from SQL, PL/SQL, and PL/pgSQL?,» TablePlus, 12 September 2018. [Internett]. Available: <https://tableplus.com/blog/2018/09/what-is-tsql-how-is-it-different-from-sql-plsql-plpgsql.html>. [Funnet 6 Mai 2022].
- [6] Microsoft, «What is Azure Data Studio?,» Microsoft, 23 03 2022. [Internett]. Available: <https://docs.microsoft.com/en-us/sql/azure-data-studio/what-is-azure-data-studio?view=sql-server-ver16>. [Funnet 6 05 2022].
- [7] statcounter.com, «Statcounter,» statcounter.com, [Internett]. Available: <https://gs.statcounter.com/os-market-share/desktop/worldwide>. [Funnet 06 Mai 2022].
- [8] statcounter.com, «Statcounter,» statcounter.com, [Internett]. Available: <https://gs.statcounter.com/os-market-share/mobile/norway>. [Funnet 6 Mai 2022].
- [9] Stack Overflow, «Developer Survey Results 2017 - Technology - most popular developer environments by occupation,» Stack Overflow, 2017. [Internett]. Available: [https://insights.stackoverflow.com/survey/2017/#technology\\_-\\_most-popular-developer-environments-by-occupation](https://insights.stackoverflow.com/survey/2017/#technology_-_most-popular-developer-environments-by-occupation). [Funnet 20 05 2022].
- [10] Stack Overflow, «Developer Survey Results 2018,» Stack Overflow, 2018. [Internett]. Available: <https://insights.stackoverflow.com/survey/2018/#development-environments-and-tools>. [Funnet 20 05 2022].
- [11] Stack Overflow, «Developer Survey Results 2019,» Stack Overflow, 2019. [Internett]. Available: <https://insights.stackoverflow.com/survey/2019/#development-environments-and-tools>. [Funnet 20 05 2022].
- [12] Stack Overflow, «2021 Developer Survey,» Stack Overflow, 2021. [Internett]. Available: <https://insights.stackoverflow.com/survey/2021#technology>. [Funnet 20 05 2022].
- [13] Microsoft, «Visual Studio Developer Tools for Students,» Microsoft, [Internett]. Available: <https://visualstudio.microsoft.com/students/>. [Funnet 20 05 2022].

- [14] Microsoft, «Azure for Students,» Microsoft, [Internett]. Available: <https://azure.microsoft.com/nb-no/free/students/>. [Funnet 20 05 2022].
- [15] Microsoft, «The history of C#,» Microsoft, 11 03 2022. [Internett]. Available: <https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history>. [Funnet 20 05 2022].
- [16] Microsoft, «Visual Studio IDE with .NET,» Microsoft, [Internett]. Available: <https://visualstudio.microsoft.com/vs/features/net-development/>. [Funnet 20 05 2022].
- [17] Microsoft, «What is .NET?,» Microsoft, [Internett]. Available: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>. [Funnet 20 05 2022].
- [18] Microsoft, «.NET API browser,» Microsoft, [Internett]. Available: <https://docs.microsoft.com/en-us/dotnet/api/?view=net-6.0> . [Funnet 20 05 2022].
- [19] S. Pauly, «1 Codebase. 3 Platforms. Here's How.,» medium.com, 5 10 2020. [Internett]. Available: <https://medium.com/swlh/1-codebase-3-platforms-heres-how-c524e0be8c63>. [Funnet 20 05 2022].
- [20] Microsoft, «Desktop Guide (Windows Forms .NET),» Microsoft, [Internett]. Available: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-6.0>. [Funnet 20 05 2022].
- [21] Stack Overflow, «Developer Survey Results 2017 - Most popular technologies platform,» Stack Overflow, 2017. [Internett]. Available: <https://insights.stackoverflow.com/survey/2021/#most-popular-technologies-platform>. [Funnet 20 05 2022].
- [22] D. Clark, «SQL vs T-SQL: Understanding the Differences,» Dataquest, 04 03 2021. [Internett]. Available: <https://www.dataquest.io/blog/sql-vs-t-sql/>. [Funnet 20 05 2022].
- [23] Microsoft, «Transact-SQL Reference (Database Engine),» Microsoft, 11 01 2022. [Internett]. Available: <https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15>. [Funnet 20 05 2022].
- [24] Microsoft, «SQL Injection,» Microsoft, 19 03 2021. [Internett]. Available: <https://docs.microsoft.com/en-us/sql/relational-databases/security/sql-injection?view=sql-server-ver15>. [Funnet 20 05 2022].
- [25] geeksforgeeks.org(@MKS075), «Difference between Function and Procedure,» geeksforgeeks.org, 18 12 2019. [Internett]. Available: <https://www.geeksforgeeks.org/difference-between-function-and-procedure/>. [Funnet 20 05 2022].

- [26] Microsoft, «CREATE SEQUENCE (Transact-SQL),» Microsoft, 30 01 2021. [Internett]. Available: <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-sequence-transact-sql?view=sql-server-ver16>. [Funnet 20 05 2022].
- [27] GitHub, «GMap.NET License,» GitHub, [Internett]. Available: <https://github.com/judero01col/GMap.NET/wiki/License>. [Funnet 9 Mai 2022].
- [28] Several inc Stack Overflow, «GitHub - Dapper,» [Internett]. Available: <https://github.com/DapperLib/Dapper>. [Funnet 20 05 2022].
- [29] M. Steele, «YouTube - Dapper - A Simple Object Mapper for .NET - Lightning Talk - Dot Net Sheff - July 2018,» Dot Net Sheff, 15 07 2018. [Internett]. Available: [https://www.youtube.com/watch?v=4Wmjo2zXt\\_M](https://www.youtube.com/watch?v=4Wmjo2zXt_M). [Funnet 20 05 2022].
- [30] The OpenStreetMap Foundation, «OpenStreetMap,» The OpenStreetMap Foundation, [Internett]. Available: [openstreetmap.org](http://openstreetmap.org). [Funnet 20 05 2022].
- [31] The OpenStreetMap Foundation, «The OpenStreetMap Foundation,» The OpenStreetMap Foundation, [Internett]. Available: <https://blog.openstreetmap.org/about/>. [Funnet 20 05 2022].
- [32] The OpenStreetMap Foundation, «Elements - OpenStreetMap Wiki,» The OpenStreetMap Foundation, [Internett]. Available: <https://wiki.openstreetmap.org/wiki/Elements>. [Funnet 20 05 2022].
- [33] Høyskolen på Vestlandet, «ELE301 Industriell IT,» Høyskolen på Vestlandet, [Internett]. Available: <https://www.hvl.no/studier/studieprogram/emne/37/ele301>. [Funnet 20 05 2022].
- [34] Microsoft, «Azure Active Directory (Azure AD),» Microsoft, [Internett]. Available: <https://azure.microsoft.com/en-us/services/active-directory/#features>. [Funnet 20 05 2022].
- [35] Microsoft, «When to Use Procedures,» Microsoft, 02 11 2021. [Internett]. Available: <https://docs.microsoft.com/en-us/sql/odbc/reference/develop-app/when-to-use-procedures?view=sql-server-ver15>. [Funnet 20 05 2022].
- [36] J. M. J. Diaz, «Lesson Learned #4: Modifying the default time zone for your local time zone.,» Microsoft, 27 07 2016. [Internett]. Available: <https://techcommunity.microsoft.com/t5/azure-database-support-blog/lesson-learned-4-modifying-the-default-time-zone-for-your-local/ba-p/368798>. [Funnet 20 05 2022].
- [37] GitHub, «GitHub - GMap.NET,» GitHub, [Internett]. Available: <https://github.com/judero01col/GMap.NET>. [Funnet 20 05 2022].
- [38] iamtimcorey.com, «YouTube - How to connect C# to SQL (the easy way),» iamtimcorey.com, 28 02 2017. [Internett]. Available: <https://www.youtube.com/watch?v=Et2khGnrlqc>. [Funnet 20 05 2022].

- [39] Several inc Stack Overflow, «GitHub - Dapper - Readme,» [Internett]. Available: <https://github.com/DapperLib/Dapper/blob/main/Readme.md>. [Funnet 20 05 2022].
- [40] Microsoft, «Microsoft Store - E-post og Kalender,» Microsoft, [Internett]. Available: <https://apps.microsoft.com/store/detail/epost-og-kalender/9WZDNCRFHVQM?hl=nb-no&gl=NO>. [Funnet 28 Mai 2022].
- [41] Microsoft, «Microsoft Teams,» Microsoft, [Internett]. Available: <https://www.microsoft.com/nb-no/microsoft-teams/log-in>. [Funnet 28 Mai 2022].
- [42] Discord, «Discord,» Discord, [Internett]. Available: <https://discord.com/>. [Funnet 28 Mai 2022].
- [43] Alphabet Inc, «YouTube,» Alphabet Inc, [Internett]. Available: <https://www.youtube.com> . [Funnet 20 05 2022].
- [44] CodeCraks, «Youtube - How to Create a Modern Flat UI Design Dashboard in C# 2020,» CodeCraks, 29 Juni 2020. [Internett]. Available: <https://www.youtube.com/watch?v=vYDyGxoq9JU>. [Funnet 22 Februar 2022].
- [45] Microsoft, «Use a theme to create a custom look for your app,» Microsoft, 05 05 2022. [Internett]. Available: <https://docs.microsoft.com/en-us/power-apps/maker/model-driven-apps/create-themes-organization-branding> . [Funnet 20 05 2022].
- [46] Stackoverflow, «Stackoverflow,» Stackoverflow, [Internett]. Available: <https://stackoverflow.com/>. [Funnet 13 Mai 2022].
- [47] Microsoft, «Timer Class,» Microsoft, [Internett]. Available: <https://docs.microsoft.com/en-us/dotnet/api/system.threading.timer?view=netframework-4.8> . [Funnet 20 05 2022].
- [48] P. Peranzo, «Imaginovation,» 26 Oktober 2020. [Internett]. Available: <https://imaginovation.net/blog/successfully-conduct-app-user-testing/>. [Funnet 9 Mai 2020].
- [49] Mazemap, «Mazemap,» Mazemap, [Internett]. Available: <https://www.mazemap.com/>. [Funnet 20 Mai 2022].
- [50] Microsoft, «ASP.NET,» Microsoft, [Internett]. Available: <https://dotnet.microsoft.com/en-us/apps/aspnet> . [Funnet 20 05 2022].
- [51] Microsoft, «An overview of Azure SQL Database and SQL Managed Instance security capabilities,» Microsoft, 26 04 2022. [Internett]. Available: <https://docs.microsoft.com/en-us/azure/azure-sql/database/security-overview?view=azuresql>. [Funnet 20 05 2022].
- [52] Microsoft, «Application deployment prerequisites (Windows desktop),» Microsoft, 30 04 2022. [Internett]. Available: <https://docs.microsoft.com/en-us/visualstudio/deployment/application-deployment-prerequisites?view=vs-2022>. [Funnet 20 05 2022].

- [53] Icons8, «Microsoft Store - Pichon - Free Icons,» Icons8, [Internett]. Available: <https://apps.microsoft.com/store/detail/pichon-free-icons/9NK8T1KSHFFR?hl=nb-no&gl=NO>. [Funnet 27 Mai 2022].
- [54] Icons8, «Icons8,» Icons8, [Internett]. Available: <https://icons8.com/icons>. [Funnet 27 Mai 2022].
- [55] Microsoft, «Download and install Azure Data Studio,» Microsoft, 20 05 2022. [Internett]. Available: <https://docs.microsoft.com/en-us/sql/azure-data-studio/download-azure-data-studio?view=sql-server-ver15>. [Funnet 20 05 2022].
- [56] Microsoft, «Download SQL Server Management Studio (SSMS),» Microsoft, 25 05 2022. [Internett]. Available: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>. [Funnet 27 05 2022].
- [57] Microsoft, «Visual Studio: IDE and Code Editor for Software Developers and Teams,» Microsoft, [Internett]. Available: <https://visualstudio.microsoft.com>. [Funnet 20 05 2022].
- [58] Google, «Google Maps Platform,» Google, [Internett]. Available: <https://mapsplatform.google.com>. [Funnet 20 05 2022].
- [59] Google, «Pricing - Google Maps Platform,» Google, [Internett]. Available: <https://mapsplatform.google.com/pricing/>. [Funnet 20 05 2022].
- [60] Balaji S, «YouTube - How to Get Google Map API Key?,» Balaji S, 04 11 2018. [Internett]. Available: <https://www.youtube.com/watch?v=eXJ1qkTMLw8>. [Funnet 20 05 2022].
- [61] Creative Commons, «Attribution 4.0 International,» Creative Commons, [Internett]. Available: <https://creativecommons.org/licenses/by/4.0/legalcode>. [Funnet 30 05 2022].
- [62] OsmSharp, «OpenStreetMap Data Model,» OsmSharp, [Internett]. Available: <http://docs.itinero.tech/docs/osmsharp/osm.html>. [Funnet 20 05 2022].

## Appendikser

### Appendiks A Forkortelser og ordforklaringer

Android	Android er et mobilt operativsystem. Eies av Google Inc.
API	Forkortelse for «Application Programming Interface». Er en kode som brukes for å utveksle data mellom to forskjellige systemer eller apper.
Azure	Microsofts samling av skytjenester, leverer bl.a. virtuelle maskiner, infrastrukturløsninger, servere og databaser.
C#	Programmeringsspråk utviklet av Microsoft
Database	Strukturert samling av data
ELE102	Emnekode; programmering og mikrokontroller
ELE205	Emnekode; Videregående programmering
ELE301	Emnekode; Industriell IT
ELE306	Emnekode; Robotikk
GitHub	Skylagring og versjonskontroll av programmeringsprosjekter
GNSS	Forkortelse for «Global Navigation Satellite Systems», fellesbetegnelse for satellittbaserte navigasjon og posisjoneringssystemer.
GPS	Forkortelse for «Global Positioning System». GPS er et satellittnavigasjonssystem. På folketale kan av og til sikte til «GNSS».
GUI	«Graphical User Interface» Grafisk brukergrensesnitt
HVL	Forkortelse for: Høgskolen på Vestlandet
IDE	Integrated Development Environment
iOS	iOS er et mobilt operativsystem. Eies av Apple Inc.
Polygon	En polygon eller en mangekant er en lukket kurve i planet, bestående av rette linjestykker.
Properties	Egenskaper. I C# oppfører de seg som variabler men er implementert med tilgangsmedlemmer.
Rammeverk	Refererer til en underliggende struktur som brukes som et grunnlag for å bygge andre ting.
SQL-spørring	Forespørsel om data fra en database
Tilgangsmedlemmer	Innkapsling av datamedlemmer
Visual Studio	IDE brukt for å programmere prosjektet i C#
Whitelist	Forteller hva som skal være tillatt, og alt annet er implisitt ikke tillatt.
Windows Forms	Verktøy for å utvikle GUI
XML-format	Forkortelse for «Extensible Markup Language». XML er et verktøy for deling av strukturerte data mellom informasjonssystemer.

## **Appendiks B      Prosjektledelse og styring**

### **B.1            Prosjektorganisasjon**

Som en gruppe på to personer valgte vi bort å ha en definert leder for prosjektet. Istedentfor valgte vi å ha hyppig kommunikasjon, og å jobbe sammen på grupperom eller via Microsoft Teams. På den måten kunne vi hele tiden drøfte og diskutere elementer med hverandre og sørge for at vi jobbet i sammen retning. For oss fungerte dette som en bra løsning. Vi hadde laget en tydelig framdriftsplan som vi jobbet etter og tok for oss prosjektet del for del i tur og orden, framfor å jobbe på forskjellige arbeidsområder.

### **B.2            Fremdriftsplan**

Se gantt-diagram i vedlegg

## Appendiks C Brukerdokumentasjon

### C.1 Oppsett av programvare

For at programmet skal fungere krever det også tilkobling til en database med riktig oppsett. For oppskrift på dette se appendiks C.4 og C.5. Pass på at du bruker riktig tilkoblingssteng for databasen i programmet dersom du har satt opp databasen selv, og at din klients IP-adresse har tilgang til databasen.

Programmet, både som .exe fil og prosjektet fra Visual Studio, finnes i vedlegg.

Programmet som .exe fil kan startes på en maskin som kjører Windows 2000 eller nyere og har installert .NET v4.8. [52]

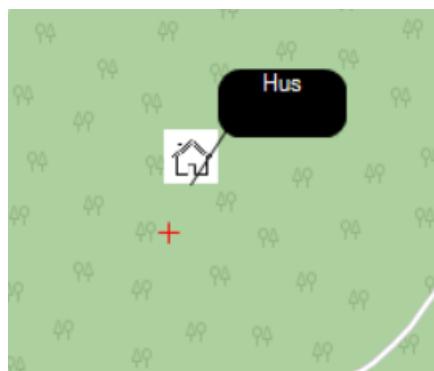
Prosjektet fra Visual Studio kan også åpnes og kjøres via Visual Studio hvis du har installert tilleggspakkene for å lage .NET og Windows Forms applikasjoner. Dersom du gjør dette eller har kopiert kode fra prosjektet til ditt eget prosjekt er det mulig du må laste ned bibliotekene GMAP (Windows Forms) og Dapper via «nuget package manager».

### C.2 Ikoner

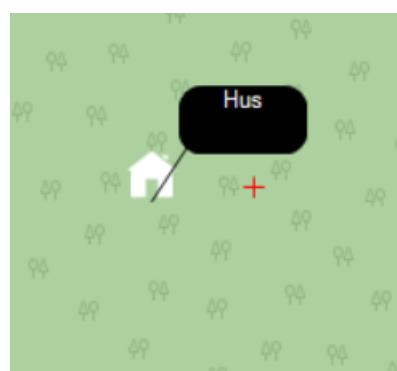
Alle ikonene som er blitt brukt i programmet er hentet fra: «Pichon - Free Icons». Det er en app i Windows Store [53], men er også en nettside man kan hente ikoner ifra [54]. På nettsiden har man flere muligheter rundt fargevalg enn i appen. Appen/nettsiden har et stort utvalg av ikoner.

Vi anbefalere denne appen/nettsiden for å finne passende ikoner til kategorier/objektsymbol som skal vise på kartet. Nettopp fordi disse ikoner kommer med usynlig bakgrunn, slik at appen virker mer gjennomført, se figur 37 og 38. Hvis man ønsker å lage egne, burde man fjerne bakgrunnene. Dette kan være tidkrevende.

Anbefalt størrelse på ikoner som blir brukt i programmet er: 25\*25 Pixel



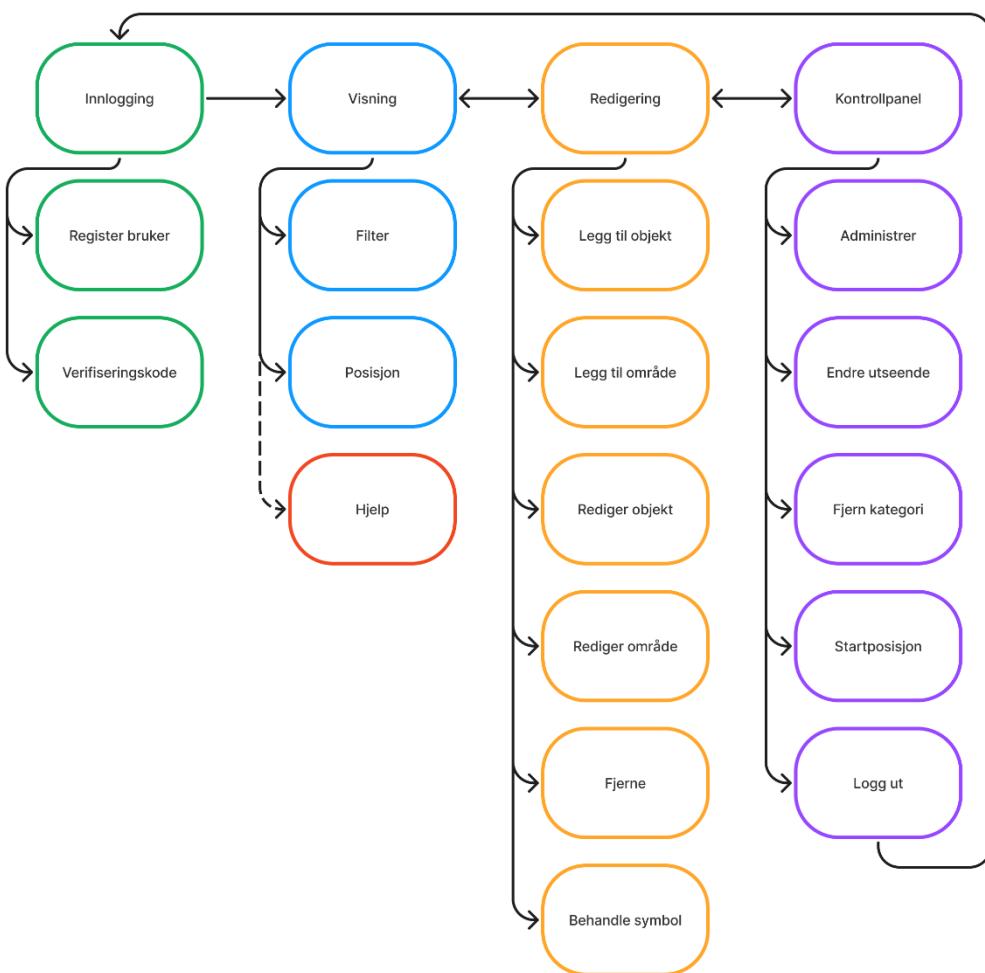
Figur 37 - Husikon med hvit bakgrunn.



Figur 38 - Husikon fra Pichon.

### C.3 Brukerdokumentasjon

Programmet består av 4 hovedseksjoner: Innlogging, visning, redigering og kontrollpanel. Hver av disse sidene benytter undersider som vises i deler av programbildet (REF figur i 5.4.1.3). For en mer teknisk oversikt over programmet se kapittel 5.4. Videre går ikke detaljert igjennom hver funksjon programmet har, men viser en kort beskrivelse av alternativene til fanene. Ønsker du å se en mer detaljert brukermanual, kan man se i vedlegg: «Brukermanual Gmap-demo». Her går vi mer nøyde igjennom hvordan man skal bruke de forskjellige funksjonene.



*Figur 39 - Hovedseksjonene i vårt program.*

Her følger en kort forklaring til funksjonsområdet til de forskjellige delene av vårt program.

- **Innlogging**  
I innlogging kan du logge inn med epost og passord.

- **Registrer bruker**

I registrer bruker kan du opprette en ny bruker ved å fylle inn nødvendig informasjon.

- **Verifiseringskode**

I verifiseringskode skriver du inn en kode mottatt på e-post for å verifisere at du eier e-posten du forsøker å opprette bruker med.

- **Visning**

Visning fungerer som hovedsiden for vårt program. Her vises kartet med objekter og områder. Visning starter med filter som standard underseksjon vist.

- **Filter**

Filter er underseksjonen hvor du kan filtrere bort objekter og områder. Her får du også opp all informasjon om objekter og områder.

- **Posisjon**

I posisjon kan du flytte kartet til et bestemt sted ved å søke på land, by eller adresse. Du kan også få opp en rute mellom to steder.

- **Hjelp**

Hjelp er en side som er laget for å forklare andre programmerere, på en visuell måte, hvordan programmet er satt sammen.

- **Redigering**

Redigering er hovedseksjonen for å endre objekter og områder.

- **Legg til objekt**

I legg til objekt kan du legge inn nye objekter i programmet.

- **Legg til område**

I legg til område kan du legge inn nye områder i programmet.

- **Rediger objekt**

I rediger objekt kan du endre aspekter ved objekter som allerede eksisterer i programmet.

- **Rediger område**

I rediger område kan du endre aspekter ved områder som allerede eksisterer i programmet.

- **Fjerne**

Her kan du fjerne objekter og områder som du ikke lenger ønsker å ha i programmet.

- **Behandle symbol**

Her kan du legge til, endre eller fjerne objektsymbol for en kategori av objekter i programmet.

- **Kontrollpanel**

Kontrollpanel er hovedsekksjonen for andre oppgaver du kan utføre i programmet. Flere av oppgavene er knyttet til endringer i måten programmet virker/framstår.

- **Administrer**

I administrer kan du godkjenne eller avslå nye brukere, og du kan oppgradere og nedgradere sikkerhetsklareringen til brukere.

- **Endre utseende**

I endre utseende kan du forandre fargetema for programmet.

- **Fjern kategori**

Fjern kategori lar deg fjerne kategorier for objekter, dersom de ikke lenger er i bruk av noen objekter.

- **Startposisjon**

I startposisjon kan du endre hvor på kartet du ser når programmet starter.

- **Logg ut**

Logg ut lar deg logge ut av programmet. Programmet avsluttes og startes på nytt på innloggingsskjermen.

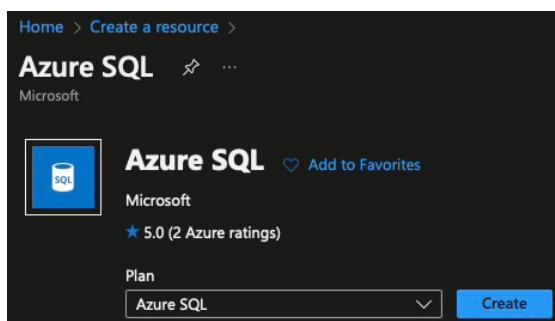
I tillegg har vi en hjelpestside som forklarer strukturen av brukergrensesnittet visuelt i programmet.

## C.4 Drifts- og vedlikeholds dokumentasjon

## C.5 Oppretting av Azure Database

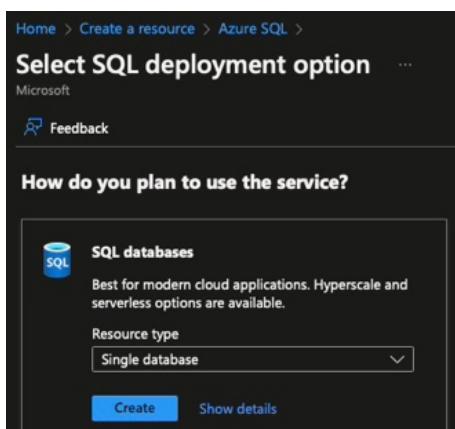
For å opprette en database i Azure så trenger du en konto i Microsoft sitt system. Dersom du er student/ansatt hos Høyskolen på Vestlandet, så har du en konto knyttet mot din elev/ansatt epost. Innlogging eller oppretting av konto kan gjøres via [www.portal.azure.com](http://www.portal.azure.com).

Fra forsiden kan du starte oppretting av databasen ved å velge «Create a resource», og deretter velge «Azure SQL» og «Create».



Figur 40 - Opprette «Azure SQL» database: Ressurstype.

Velg deretter «Single Database» og «Create».



Figur 41 - Opprette «Azure SQL» database: Databasetype.

I neste vindu så starter vi oppsettet av databasen med «Basics»:

Home > Create a resource > Azure SQL > Select SQL deployment option >

## Create SQL Database

Microsoft

**Basics** Networking Security Additional settings Tags Review + create

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

**Did you know** that new users in Azure can create a free Azure SQL Database and use it for 12 months using Azure free account? [Learn more](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Resource group \* ⓘ  [Create new](#)

**Database details**

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \*

Server \* ⓘ  [Create new](#)

Want to use SQL elastic pool? \* ⓘ  Yes  No

Compute + storage \* ⓘ

- Basic**
- 2 GB storage
- [Configure database](#)

**Backup storage redundancy**

Choose how your PITR and LTR backups are replicated. Geo restore or ability to recover from regional outage is only available when geo-redundant storage is selected.

Backup storage redundancy ⓘ

- Locally-redundant backup storage
- Zone-redundant backup storage
- Geo-redundant backup storage

**⚠️** Selected value for backup storage redundancy is Geo-redundant backup storage. Database backups will be geo-replicated which might impact your data residency requirements. [Learn more](#)

[Review + create](#) [Next : Networking >](#)

Figur 42 - Opprette «Azure SQL» database: «Basics».

Velg en «Subscription» å knytte databasen til. Opprett en ny eller velg f.eks. «Azure for Students».

Velg en «Resource Group» å knytte databasen til. Opprett en ny hvis du ikke allerede har en passende.

Gi databasen et passende navn og velg en server du ønsker å kjøre databasen på. Dersom du ikke

[Home](#) > [Create a resource](#) > [Azure SQL](#) > [Select SQL deployment option](#) > [Create SQL Database](#) >

## Create SQL Database Server

Microsoft

**Server details**

Enter required settings for this server, including providing a name and location. This server will be created in the same subscription and resource group as your database.

Server name \*  .database.windows.net

Location \*

**Authentication**

Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Azure AD authentication [Learn more](#) using an existing Azure AD user, group, or application as Azure AD admin [Learn more](#), or select both SQL and Azure AD authentication.

Authentication method  Use SQL authentication  Use only Azure Active Directory (Azure AD) authentication  Use both SQL and Azure AD authentication

Set Azure AD admin **583662@stud.hvl.no**  
Admin Object/App ID: b6fddc34-b1a5-4746-a06e-bd74b812925c  
[Set admin](#)

Server admin login \*

Password \*

Confirm password \*

Figur 43 - Opprette «Azure SQL» database: Opprette en server.

allerede har en, kan du opprette en ny.

Gi i så fall serveren et passende navn, velg stedet du ønsker databasen skal kjøres fra og velg autentiseringsmetode. For autentiseringsmetode kan det være greit å ha begge deler da du kan bruke SQL autentisering i programmet du lager, og Azure autentisering når du logger på databasen i nettoportalen.

I dette prosjektet trenger vi ikke å benytte oss av SQL elastic pool. Den enkleste løsningen holder med god margin.

I kategorien «Compute + Storage» velg «Configure database».

I kategorien «Service Tier», velg «DTU-Based Purchasing Model -> Basic (For less demanding workloads)».

«Data max size» utgjør ikke noen forskjell i pris så du kan like greit gå for max (2 GB).

Estimert pris på databasen utgjør da i skrivende stund 5.49 USD per måned. Dersom du har Azure for Students credits vil dette bli trukke av disse.

Når dine credits utløper vil du motta en epost om du ønsker å gå over til å betale for å beholde tjenestene eller om du ønsker å avslutte abonnementet. [14]

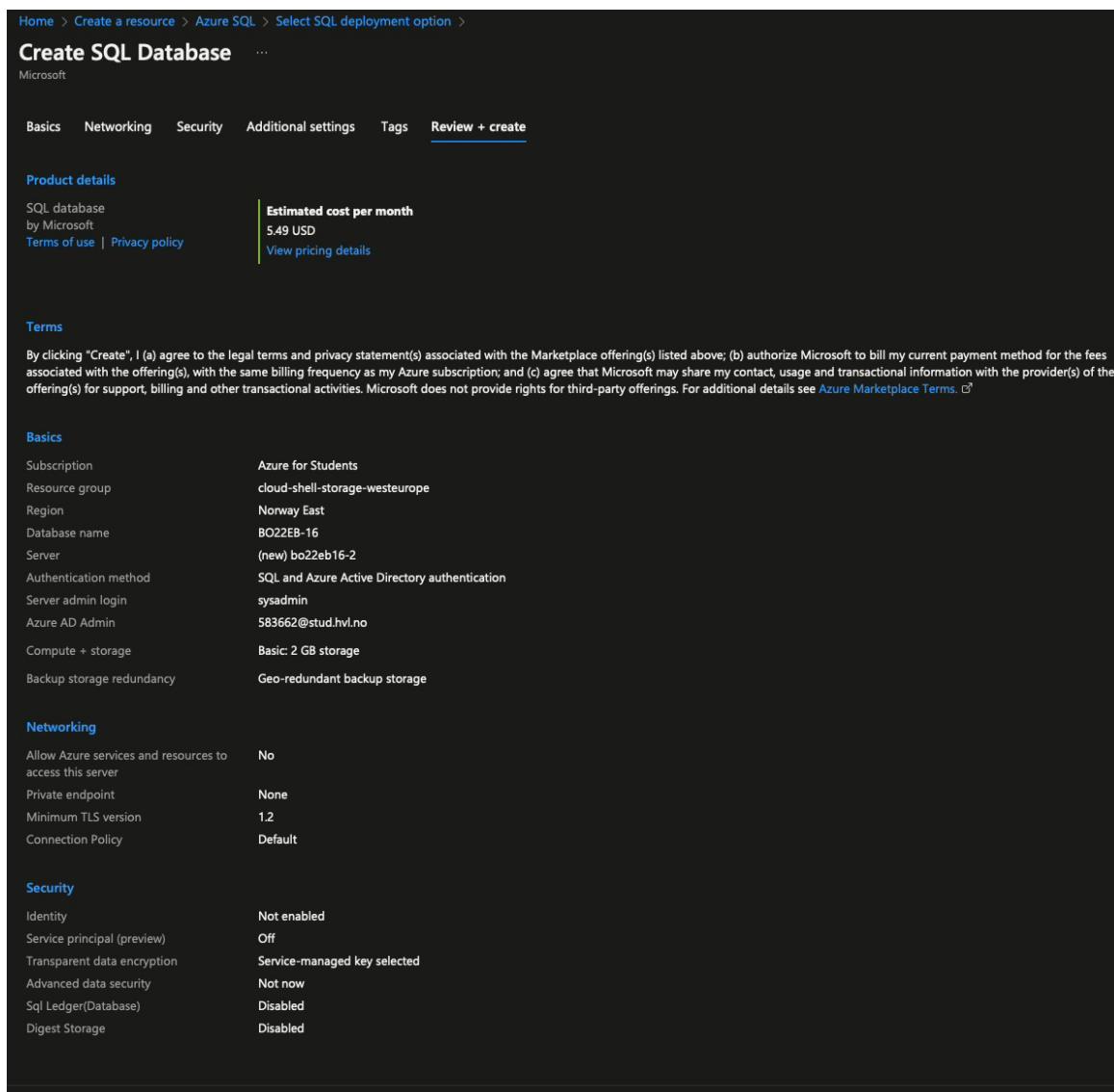
Trykk «Apply» nederst for å bekrefte konfigurasjonen og gå tilbake til «Basics».

For «Backup storage redundancy», la den stå på forhåndsvalet innstilling eller gjør endring dersom du har spesielle behov.

Trykk på «Next: Networking >» for å gå videre til neste side for oppretting av database.

For de neste kategoriene «Networking», «Security», «Additional Settings» og «Tags» fungerer dette prosjektet fint med standardinnstillingene. Dersom du ønsker spesielle innstillinger setter du de der du ønsker og går videre igjennom kategoriene.

Til slutt kan du se over innstillingene du har valgt i «Review + Create».



Figur 44 - Opprette «Azure SQL» database: Oversikt og bekreft oppretting.

Når du er fornøyd trykker du på «Create» for å starte opprettingen av database (og evt serveren). Dette kan ta noen minutter.

Serveren og databasen dukker opp i listen med dine ressurser.

The screenshot shows the Azure services dashboard with various icons for different services like SQL databases, Azure SQL, Cost Management, Education, Storage accounts, Resource groups, Power Platform, and Azure Active Directory. Below this, the 'Resources' section is shown with tabs for 'Recent' and 'Favorite'. Under 'Recent', there are two entries:

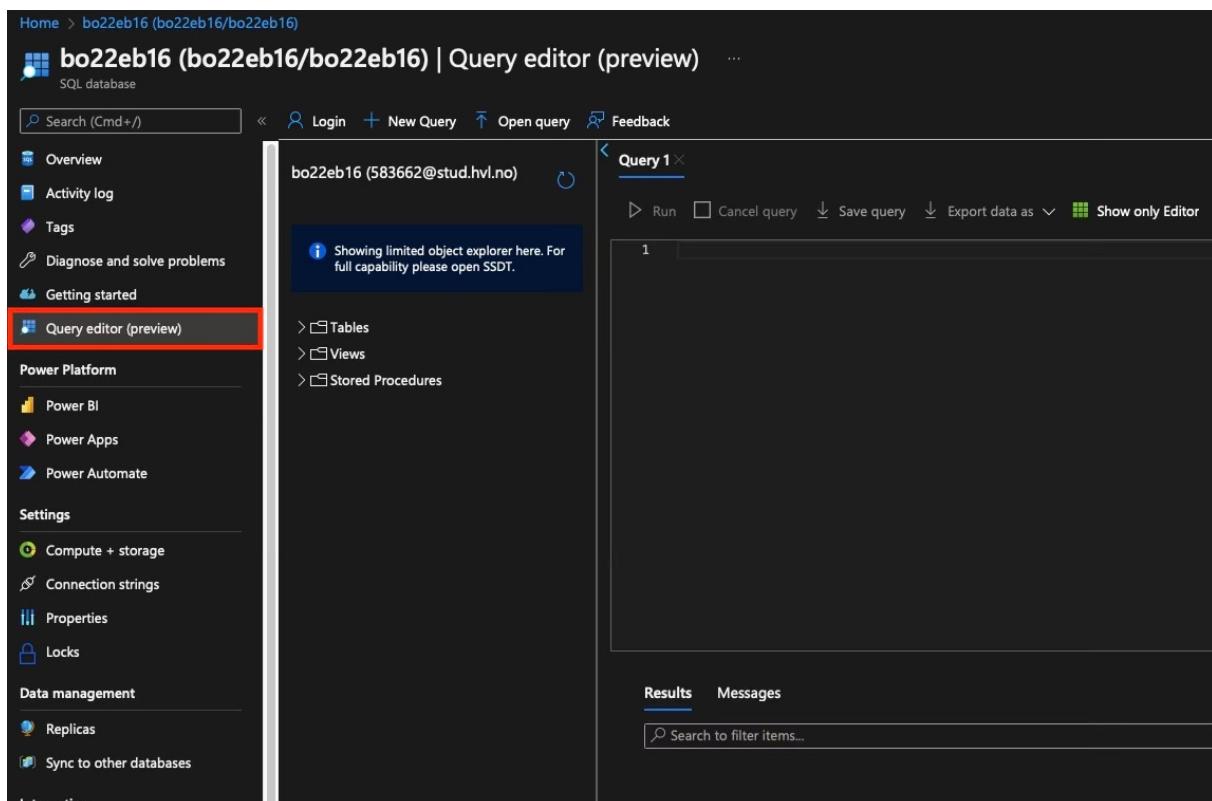
Name	Type	Last Viewed
bo22eb16 (bo22eb16/bo22eb16)	SQL database	a few seconds ago
bo22eb16	SQL server	a few seconds ago

Figur 45 - Azure: Dine ressurser.

Du kan nå velge å koble til databasen med forskjellige programmer. Bl.a.:

- Azure Data Studio [55]
- SQL Server Management Studio [56]
- Visual Studio [57]

Ved å velge databasen får du opp en god del muligheter. Her kan du også velge å utføre spørninger i SQL kode og jobbe i databasen på den måten.

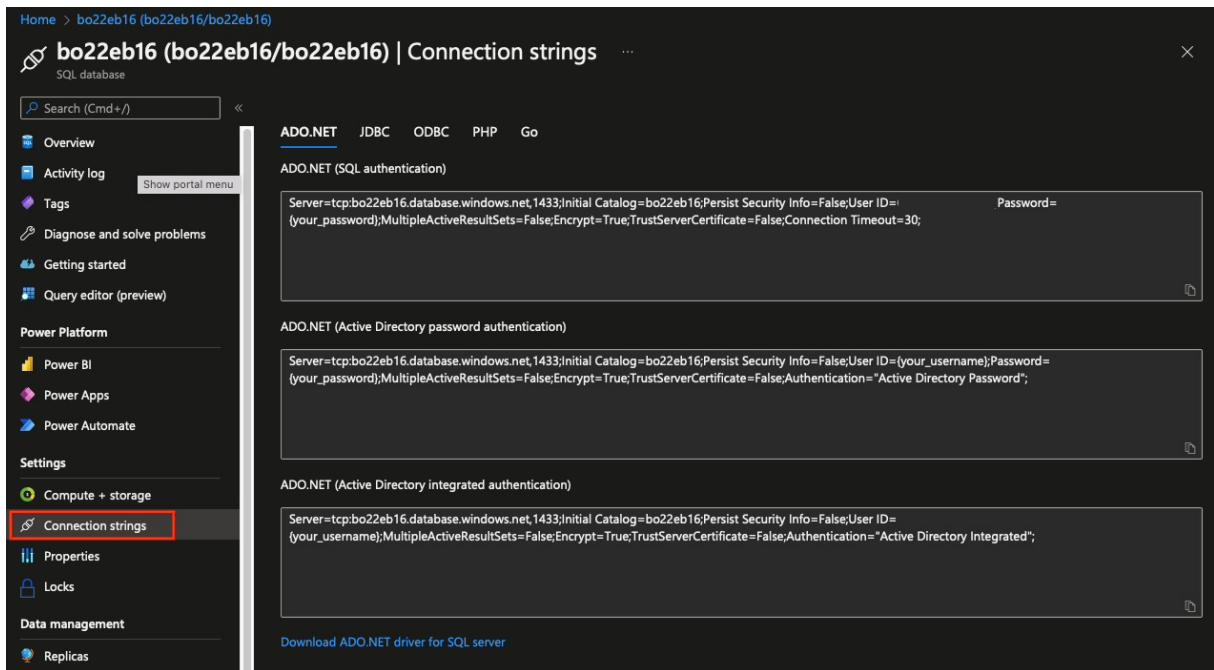


Figur 46 - Utfør spøringer.

Dersom du ønsker å sette opp databasen slik som vi har gjort det i vårt prosjekt kan du kopiere SQL scriptet som du finner i vedlegg og kjøre dette i «Query Editor (preview)».

## C.6 Koble Visual Studio prosjektet ditt mot Azure Databasen

På siden til databasen, i Azure portalen, kan du velge «Connection Strings» for å få opp forskjellige tilkoblingsparametere.



Figur 47 – Tilkoblingsstrenger.

Du kan bruke disse når du skal koble Visual Studio Prosjektet ditt opp mot databasen.

Åpne «System.Properties.Settings.Settings» i prosjektet ditt. Her kan du legge inn «ConnectionString» for «SQL Authentication» som vist under. Gi den et passende navn.

Tabellen fra Visual Studio er gjengitt under for bedre leseighet.



Figur 48 - Tilkoblingssteng i Visual Studio prosjektet

Name	Type	Scope	Value
Ønsket navn	(Connection string)	Application	Connection string du hentet

Tabell 3 - Tabellen fra "Settings" som vist i figur 48

I prosjektet vårt har vi opprettet en variabel som refererer til strengen og en metode som henter ut tilkoblingsparametriene etter navnet.

```

17     public static string bo22eb16DatabasePathUrlLocation = "GMAP_Demo.Properties.Settings.bo22eb16ConnectionString";
18
19
20     public static string CnnVal(string name)
21     {
22         return ConfigurationManager.ConnectionStrings[name].ConnectionString;
23     }

```

Figur 49 - Eksempekkode for å hente ut tilkoblingssteng

Deretter bruker vi Dapper [31] til å koble opp mot databasen og utføre spørninger. Se eksempel under.

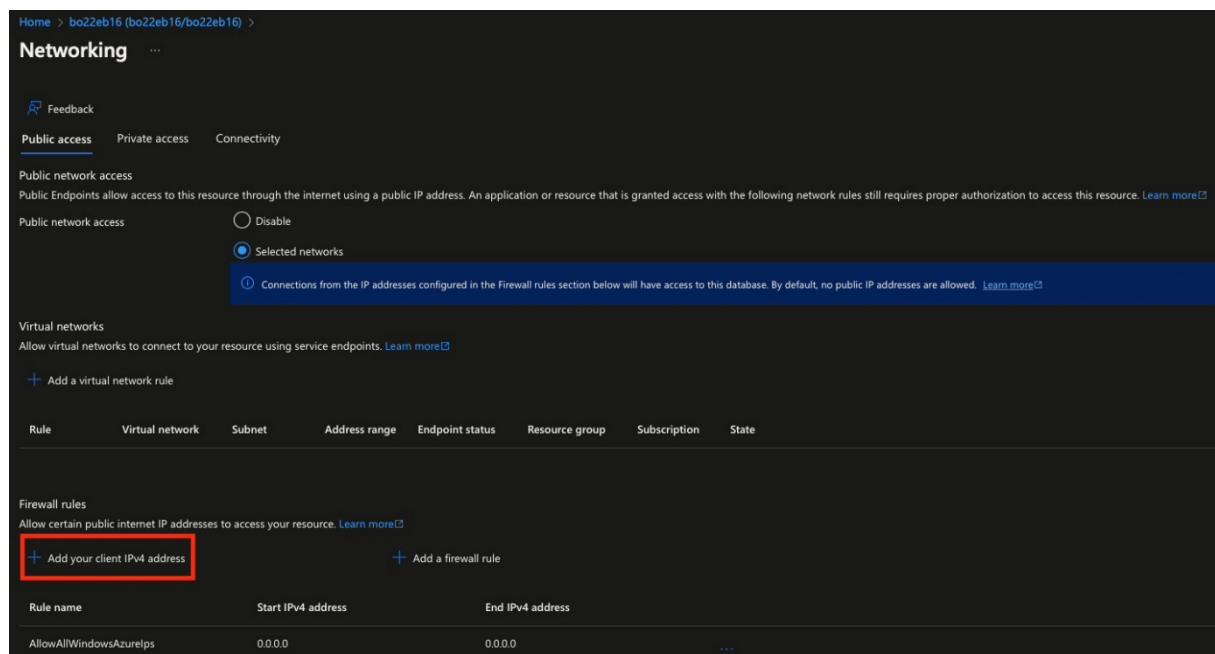
```

13  public static List<Bruker> CheckLoginAgainstDb(string Username, string Password)
14  {
15      try
16      {
17          using ( IDbConnection connection = new System.Data.SqlClient.SqlConnection(DatabaseCommunication.CnnVal(DatabaseCommunication.bo22eb16DatabasePathUrlLocation)))
18          {
19              var output = connection.Query<Bruker>($"SELECT* FROM [dbo].[Bruker] WHERE ( Epost = '{Username}' ) AND ( Passord = '{Password}' )").ToList();
20          }
21      }
22      catch (Exception exception)
23      {
24          DatabaseCommunication.FeilmeldingFikkIkkeKontaktMedDatabasen(exception);
25          List<Bruker> list = new List<Bruker>();
26          return list;
27      }
28  }

```

*Figur 50 - Eksempelkode for å utføre en spørring.*

Det er mulig at nettverk kan tildele deg en annen IP-adresse enn det du har hatt tidligere, i så fall kan du få en feilmelding om at programmet ikke fikk kontakt med databasen. Da må du legge til IP-adressen til databasens «Whitelist».

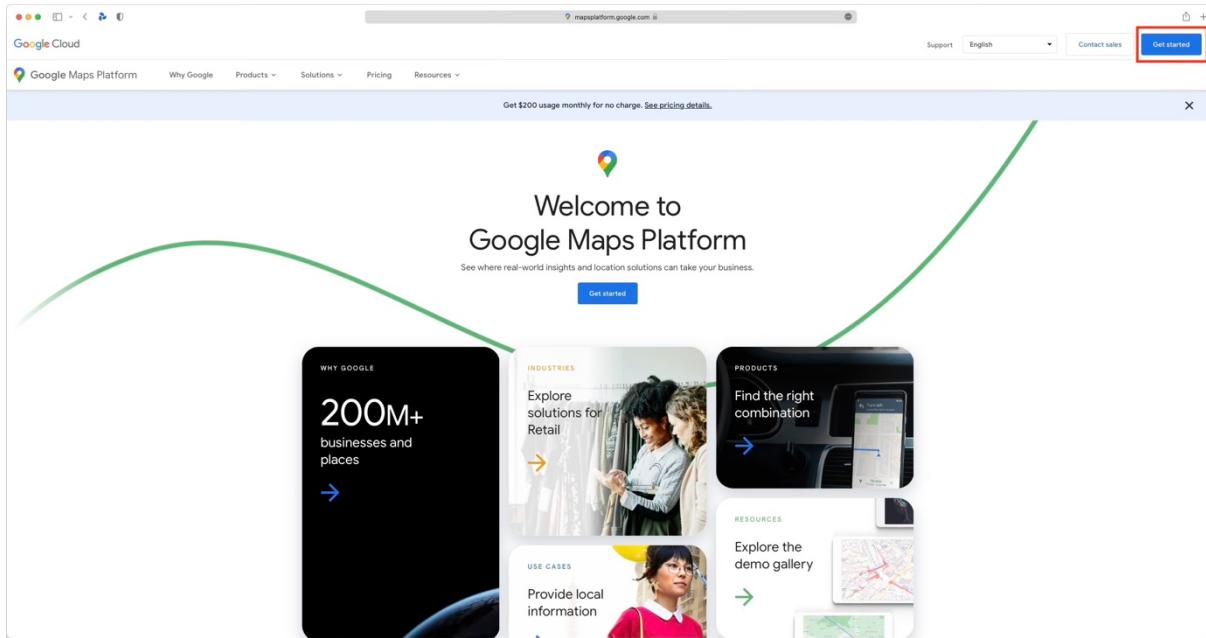


*Figur 51 - Legge inn din IP i «Azure SQL server» sin «Whitelist».*

## C.7 Opprette en «Google Cloud API key»

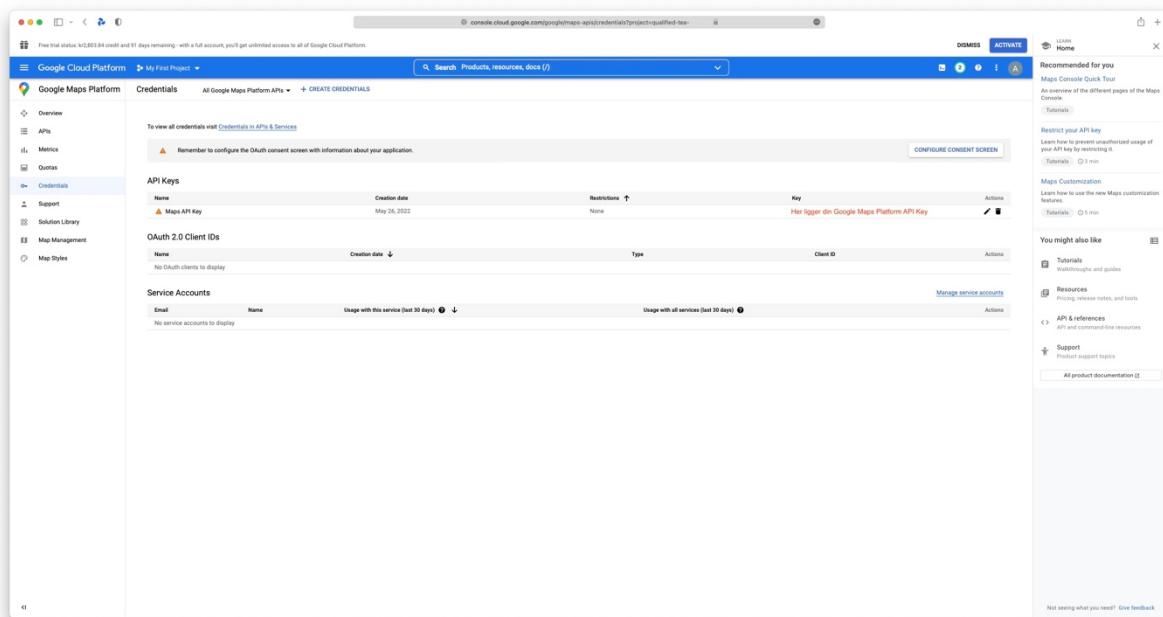
For å få tilgang til karttjenester som bruker «Google Maps Platform» via «Google Cloud», trenger du en «Google Cloud API key». I skrivende stund tilbyr Google \$300 kreditt som du kan bruke de første 90 dagene, for å teste ut «Google Maps Platform», og du blir ikke automatisk belastet etter at prøveperioden har gått ut. For øyeblikket er det også slik at du ikke blir fakturert for de første \$200 som du bruker hver måned. [58] [59]

For å få en Google cloud API key, gå til «Google Maps Platform» og velg «Get Started».



**Figur 52 - Google Maps Platform.**

Fyll inn nødvendig informasjon for å opprette kontoen og fortsett. Så snart kontoen er opprettet får du din «Google Cloud API key». [60]



**Figur 53 - Google Cloud API key.**

Dersom du bruker GMAP i ditt prosjekt kan du legge den inn ved hjelp av:

«`GMapProviders.GoogleMap.ApiKey = "Din Google Cloud API key";`»

I vårt prosjekt har vi lagt den inn i «System.Properties.Settings.Settings» og kalt den «GoogleMapApiKey». Da kan vi referere til den ved:

```
«GMapProviders.GoogleMap.ApiKey = Properties.Settings.Default.GoogleMapApiKey;»
```

## Appendiks D    Liste over vedlegg

- [Vi lager en .zip fil med alle vedlegg:](#)
  - Alle bilder og diagrammer fra rapporten som filer (Teams -> Wiseflow Vedlegg).
  - Programmet som .exe fil.
  - Program kode (hele prosjektet fra VS)
  - SQL Script
  - SQL feillogg Script
  - Oppgaven gitt til oss
  - Gantt
  - Brukermanual Gmap-demo

## Appendiks E    Creative Commons Attribution 4.0 International Public License

Det er vårt ønske at alt av koden og tilhørende rapport, som ikke omfattes av allerede eksisterende lisenser, skal være fritt tilgjengelig under «Creative Commons Attribution 4.0 International Public License». [61]

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licenser grants You such rights in consideration of benefits the Licenser receives from making the Licensed Material available under these terms and conditions.

### Section 1 – Definitions.

- a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licenser. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.
- b. **Adapter's License** means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.
- c. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section [2\(b\)\(1\)-\(2\)](#) are not Copyright and Similar Rights.
- d. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.
- e. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.
- f. **Licensed Material** means the artistic or literary work, database, or other material to which the Licenser applied this Public License.

- g. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.
- h. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.
- i. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.
- j. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.
- k. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

## Section 2 – Scope.

- a. **License grant.**
  - 0. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:
    - . reproduce and Share the Licensed Material, in whole or in part; and
    - A. produce, reproduce, and Share Adapted Material.
  - 1. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.
  - 2. Term. The term of this Public License is specified in Section [6\(a\)](#).
  - 3. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from

making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section [2\(a\)\(4\)](#) never produces Adapted Material.

4. **Downstream recipients.**

- . **Offer from the Licensor – Licensed Material.** Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.
  - A. **No downstream restrictions.** You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.
- 5. **No endorsement.** Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section [3\(a\)\(1\)\(A\)\(i\)](#).

b. **Other rights.**

- 0. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
- 1. Patent and trademark rights are not licensed under this Public License.
- 2. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

**Section 3 – License Conditions.**

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. **Attribution.**

0. If You Share the Licensed Material (including in modified form), You must:
  - . retain the following if it is supplied by the Licensor with the Licensed Material:
    - . identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
      - i. a copyright notice;
      - ii. a notice that refers to this Public License;
      - iii. a notice that refers to the disclaimer of warranties;
      - iv. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
  - A. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
  - B. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.
  1. You may satisfy the conditions in Section [3\(a\)\(1\)](#) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
  2. If requested by the Licensor, You must remove any of the information required by Section [3\(a\)\(1\)\(A\)](#) to the extent reasonably practicable.
  3. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

**Section 4 – Sui Generis Database Rights.**

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

- a. for the avoidance of doubt, Section [2\(a\)\(1\)](#) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

- b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and
- c. You must comply with the conditions in Section [3\(a\)](#) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

#### **Section 5 – Disclaimer of Warranties and Limitation of Liability.**

- a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.
- b. To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.
- a. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

#### **Section 6 – Term and Termination.**

- a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.
- b. Where Your right to use the Licensed Material has terminated under Section [6\(a\)](#), it reinstates:

0. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
1. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section [6\(b\)](#) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

- c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.
- d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

## **Section 7 – Other Terms and Conditions.**

- a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.
- b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

## **Section 8 – Interpretation.**

- a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.
- b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.
- c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.
- d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.