



Gruppe prosjekt - **Software engineering** - HIOF

Medlemmer:

Walter Eikrem, Kristoffer Snopestad Søderkvist, Kristoffer Eriksen Beck.

Frontend (Vue & webutvikling)

Vi bruker VUE til fremvisning av programmet/appen vår i en webform. det vil si at vi bruker en del generell webutvikling. dette er noe som jeg(walter) liker godt å jobbe med. så jeg tok på meg denne oppgaven alene.

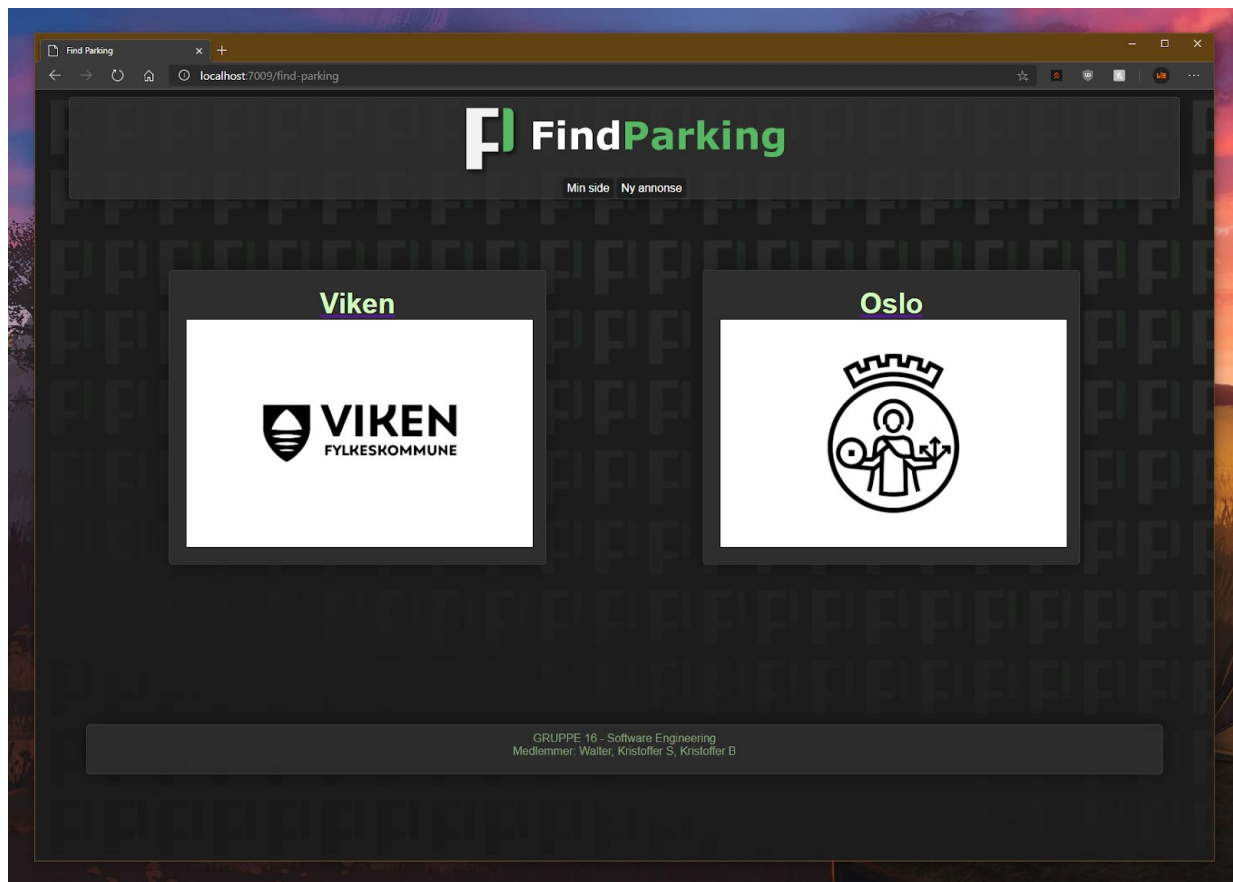
Det var ikke stort press på utforming i oppgavens krav, så vi holdt ting relativt simpelt. Vi ble enige om at vi måtte ha dark-theme, men ellers var det frie tøyler.

jeg lagde raskt en logo og et navn for prosjektet i photoshop. Så startet jeg å lage en skisse av hvordan ting ville sett ut ved å lage en standard nettside med html og css.

det gjorde at å vise frem skissen ble veldig lettvinnt. deretter når de andre på gruppen var klare for å starte på backend delen, startet vi å gjøre det om til vue filer. med en "layout.html" som base.

ting som skulle være på alle sider, typ header og footer, ble satt til å være i html filen. slik at vi ikke ved en feil endrer på noe i en vue fil.

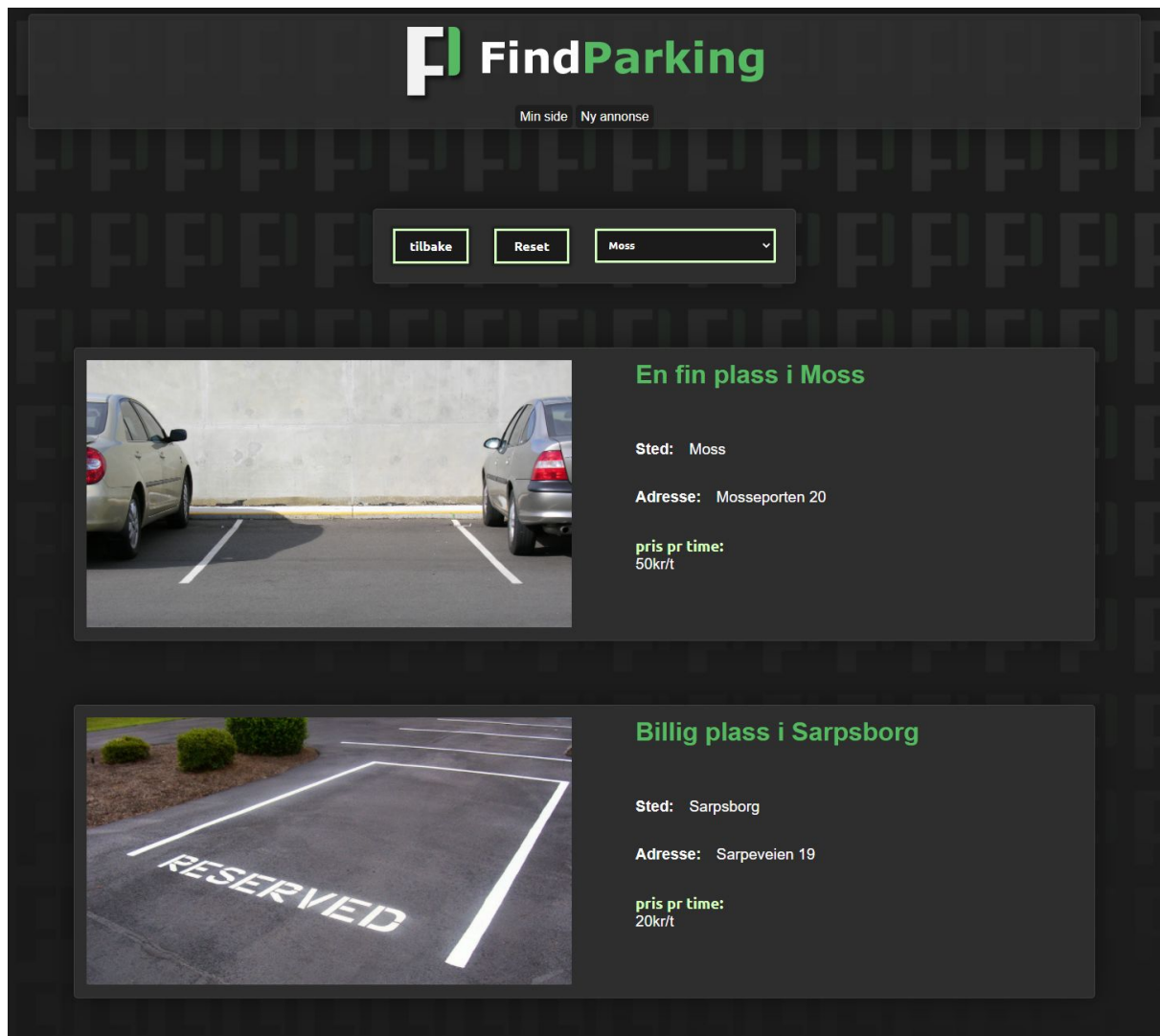
i vue filene fant vi at det holdt med å ha css i én av filen. så vi valgte forsiden til å holde på css.



designet vi gikk for var ment å være minimalistisk men moderne. jeg gikk for veldig simplistiske seksjoner som flyter for seg selv. Jeg valgte å gi seksjonene skygger for å få de til å poppe litt mot bakgrunnen. fokuset skulle være på innholdet siden viste frem. så bildene vi bruker har god kontrast mot siden.

Etter å ha valgt fylke ville vi ha litt stil ala det finn.no har for deres annonser. Nemlig seksjoner listet vertikalt nedover, hvor man trykker på et annonse-element for å gå inn på den annonsen. så det gjorde vi. det er litt css for å tydelig vise at dette er funksjonen. ved at annonsen blir highlighted ved "mouse hover". og bildet til annonsen zoomer inn ved "hover" av musepeker også.

Siden det visuelle ikke var så stort fokus i oppgaven, gjorde vi ikke noe med universell utforming eller SEO i denne prototypen.



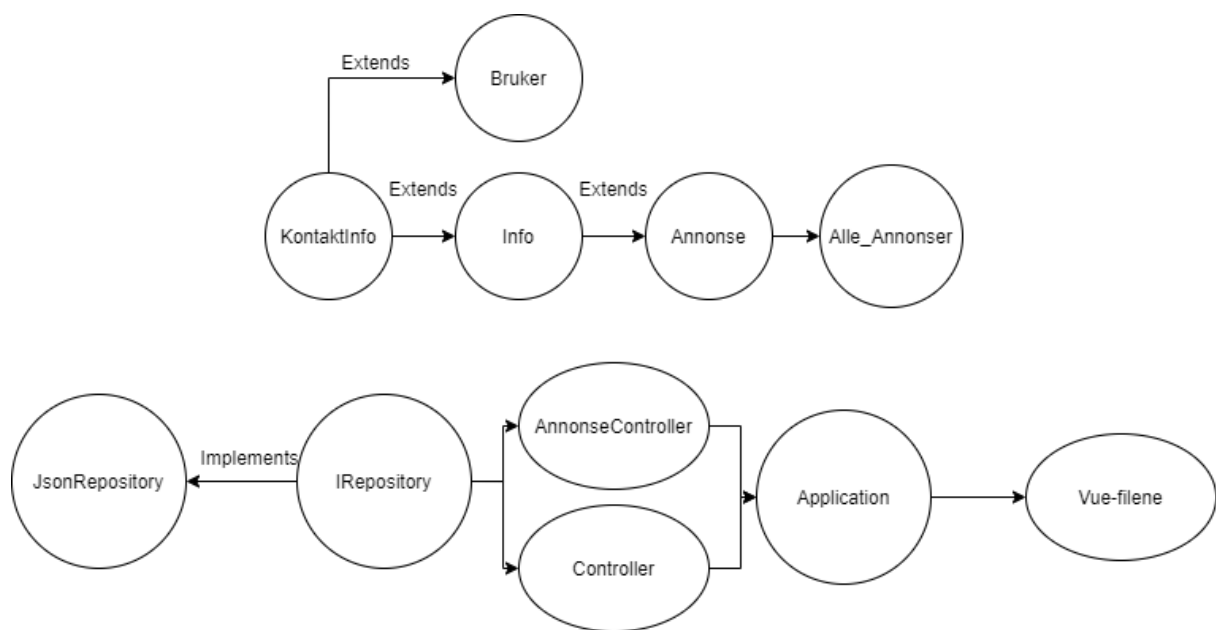
Som det blir mer utdypet lenger ned under “utfordringer”, hadde vi en del trøbbel med bruken av VUE ettersom vi overvurderte hva kunne fra før om vue som rammeverk. men til slutt tok nettsiden form. og var fungerende.

Backend

I det vi kaller for backend her bruker vi Java, gradle og Javalin. Og som nevnt bruker vi vue til frontend/UI.

Java jobber her med funksjonen/logikken bak siden. Gradle og Javalin kobler denne logikken sammen med frontend. Dette valgte vi fordi vi har brukt akkurat denne “workflow’en” i tidligere oppgaver på skolen. Og vi følte at java kunnskapen på gruppa var relativt solid nå.

Vi diskuterte også å lage en terminal løsning med python, eller en nettside som jobbet med javascript og vanlig html. hvor det ville jobbet mot json filer for å simulere en database. men det endte med java, vue og javalin.



`KontaktInfo` leverer adresse, telefonnummer og epost til `Bruker` klassen. `KontaktInfo` leverer adresse pluss sted til `Info`, for at beskrivelse blir levert med adresse og sted til `Annonse` klassen. `Alle_Annonser` klassen har en metode som heter `addAnnonse`, slik at en `List` med `Annonser` blir lagt under `Alle_Annonser` objektet.

`JsonRepository.java` holder alt av informasjon og metoder. Den blir knyttet til resten av prosjektet med at den implementerer `IRepository` som er et interface for `JsonRepository`. Da kan metodene som blir nevnt i `IRepository` bli brukt i `AnnonseController` og `Controller` klassene, det er

disse to klassene som blir kalt på i Application klassen. Metodene som blir laget i Controller klassene blir kalt på i API delen som er i Application. Denne informasjonen blir hentet i vue filene, siden i vue filene leser API adresser.

Testing

Tester toString til Annonse klassen, Alle_Annonse og Bruker klassen

I disse testene blir det sjekket at hva som blir laget i objektet i disse klassene er det samme som er “expected” i deres toString, da får man også sjekket at alt av info som blir fylt inn blir lagt riktig.

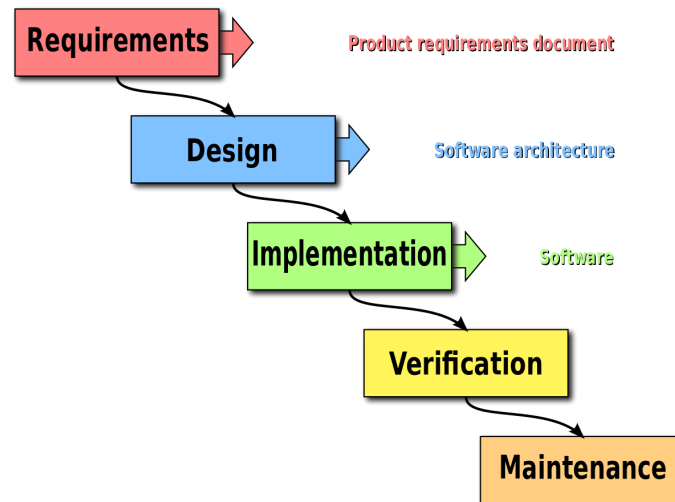
Tester toString til Annonse klassen, Alle_Annonse og Bruker klassen hvis det ikke blir fylt inn riktig info i konstruktørene.

Hvis disse testene får inn noe “unexpected” input som ikke er det samme som det som er forventet. Så vil testen bli true.

Tester create annonse metoden i JsonRepository.java av riktig input.

Denne skriver også til Annonser.json filen. Så kan man virkelig se at den virker da man starter nettsiden. Men er for å teste at metoden i JsonRepository vil virke som den skal. Denne testen kan også bli brukt for en alternativ måte å fylle inn annonser til nettsiden.

Arbeidsmetode - Gruppen



Vi prøvde å gå ut fra Agile modellen for hvordan vi skulle jobbe. Men til dels i starten ble det faktisk en slags waterfall, uten at vi egentlig planla det. fordi det ble vanskelig å gjøre noe i backend før frontend hadde noe skjelett. derfor ble det mer at arbeidet utviklet seg fallende nedover ettersom steppet over ble gjort klar.

men når det var på plass fikk vi jobbet mer i stil av Agile modellen.

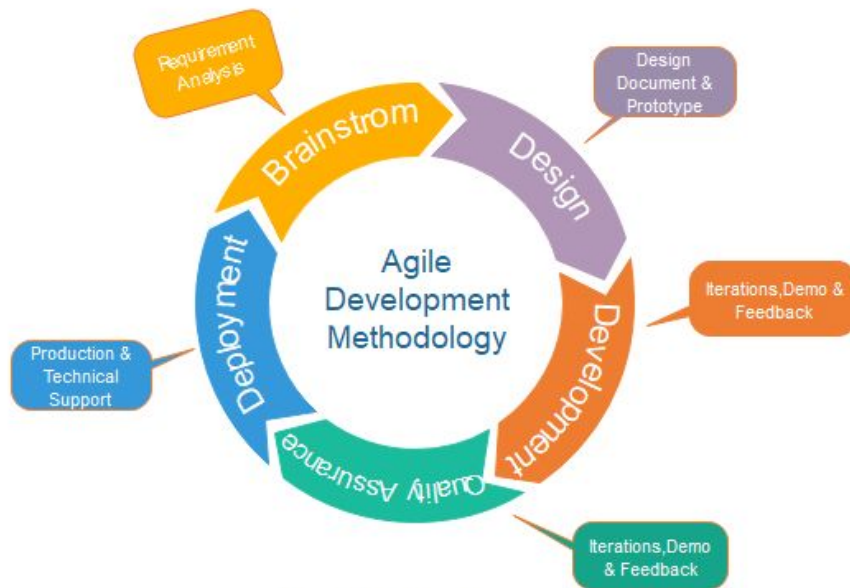


Fig. Agile Model

vi fikk utviklet mer og mer funksjonalitet, og mens det kom på plass utviklet vi UI'et, og slik gikk det dynamisk i sirkler.

Likevel endte vi ubevisst med å gå mer inn i waterfall stilen mot slutten igjen. det var mest fordi vi fikk et problem med en av kjernefunksjonene. det gjorde at alt stoppet opp.

det ble ikke bedre av at skolen ønsket at elever ikke oppholdt seg unødvendig på campus. noe som gjorde at all teamwork ble remote. dette ville ikke være noe problem for erfarne utviklere. men vi hadde aldri jobbet som dette før.

Derfor ble teamwork veldig vanskelig. vi satt på discord og livestreamet skrivebordet slik at alle kunne se hva man gjorde om man skulle trenge hjelp eller råd. det endte heller med at vi hadde et medlem som kodet av gangen, da med delt skjerm. og at vi kodet sammen på en maskin av gangen. hvor vi redigerte og sendte kodesnutter via discord i stedet. så

vi slapp å bruke git for hver minste lille detalj som ble endret. dette var jo veldig unikt for dette tilfelle.

med utviklere som er uvant med teamwork, git, og Covid som slo ned på alt. vi følte at tross alt, så gikk det ok med å ha en som skrev av gangen mens de andre hjalp til via delt skrivebord.

Utfordringer

Våres største utfordring var å finne ut hvor vi skulle starte og hvordan vi skulle dele arbeidet mellom oss på gruppa. Og komplikasjoner mot slutten av arbeidsperioden på grunn av Covid. vi hadde møter nesten hver fredag for å snakke om oppgaven, men et unikt problem kom av det faktum at et gruppemedlem meldte seg helt ut av gruppen. Vi mistet helt kontakt med denne personen. Han sa selv etter en stund, at han hadde svært lite motivasjon å jobbe. ettersom vi ikke kunne møtes i samme rom, og jobbe som ett team face-to-face.

Et av våre startproblemer var som sagt når vi skulle starte opp, og hvordan vi skulle dele opp arbeidet. Vi overvurderte vår egne kunnskaper innenfor Java og rammeverket Vue.js for frontend og JSON til lagring av data. i en ekte fungerende app, ville nok JSON blitt erstattet med en database.

Vi valgte dette som sagt lenger opp, fordi vi har jobbet med dette tidligere. og var kjent med mye av funksjonaliteten fra før.

men oppdelingen av arbeidet ble vanskelig ettersom flere deler ikke kunne testes ordentlig uten å ha mye av vue filene ferdig og klare fra før av, i dette tilfellet endte Kristoffer Snopestad Søderkvist med å gjøre mesteparten av start arbeidet. hvor han laget midlertidige vue filer for å vise frem veldig basic informasjonen. slik at man kunne feilsøke java-koden.

Som nevnt hadde også vi et gruppemedlem som forlot gruppen nesten 3 uker før fristen for innlevering. Det gjorde at vi hadde arbeid som måtte fordeles på nytt, som igjen kuttet ned på tid vi hadde til andre oppgaver. det. det skapte også masse unødvendig stress for jobb med backend delen og testing. *(Nevner dette for å forklare hvordan det var å jobbe, og ikke for å “syte”. vi tok tak i problemet og fikk ordnet det vi trengte).*

Vi var i snakk med faglærer som hjalp oss med noen beroligende råd, og en improvisert veiledning via discord. det var enkelte feil som var kommet av “crunch” og stress-koding. rett og slett slurvefeil. når vi så fikk fikset opp i det gikk ting lettere. Det var unødvendig stress, men sånn går det, og det var ikke noe å gjøre med det annet enn å jobbe litt ekstra.

Vi satt også til dels oppe til kl 3 på natten og kodet for å ta igjen arbeidet som studenten som droppet ut hadde ansvar for. derfor ble noe funksjonalitet kuttet ned på. men vi er fremdeles fornøyd med resultatet.

Konklusjon

Som konklusjon kan vi si at vi har lært hvordan det er å jobbe som et lag med koding. Dette var også vår første gang der vi jobbet med en satt arbeidsmetode som agile, selv om det ble komplikasjoner med det og.

Vi lærte at vi burde hatt backup planer i tilfellet en eller flere av medlemmene forsvinner. Ettersom det kan skje av mange årsaker. Særlig nå under Covid.

Det er klart at vi burde ha gjort mer research på det å bruke Vue. og kanskje vurdert om ikke det var andre rammeverk og språk som hadde passet bedre for oss.

Arbeidsfordeling fant vi ut er mer eller mindre 50% av et prosjekt som dette. Det at riktig person får riktig oppgave, men også at alle får en mestringsfølelse. Vi burde helt klart ha hatt bedre fordeling i begynnelsen. men det ble en lærepenge vi alle kommer til å ta til oss, og vi er ganske klare på at det er bra å lære graden teamwork og lederskap spiller i gruppearbeid som dette nå, heller en siden.

Hadde vi hatt mer tid kunne vi implementert en sorteringsmetode på byer. det var litt surt å ende med for lite tid til dette. Da ville vi nemlig også hatt muligheten til å implementere en boolean i objektene som kunne bestemme om en plass var ledig eller ikke. og deretter hatt en drop-down liste med innhold basert på boolean'n sin verdi. typ 1 for ledig, og 0 for reservert. det er klart at dette ville være enkelt nok å legge til i fremtiden, og for den saks skyld laget tester mot også.