

Utvikling av en kontekstuell Android applikasjon «Kontekst»



K

Oppgave 54E

Olav Alexander Mjelde
Bacheloroppgave ved NTNU 2016

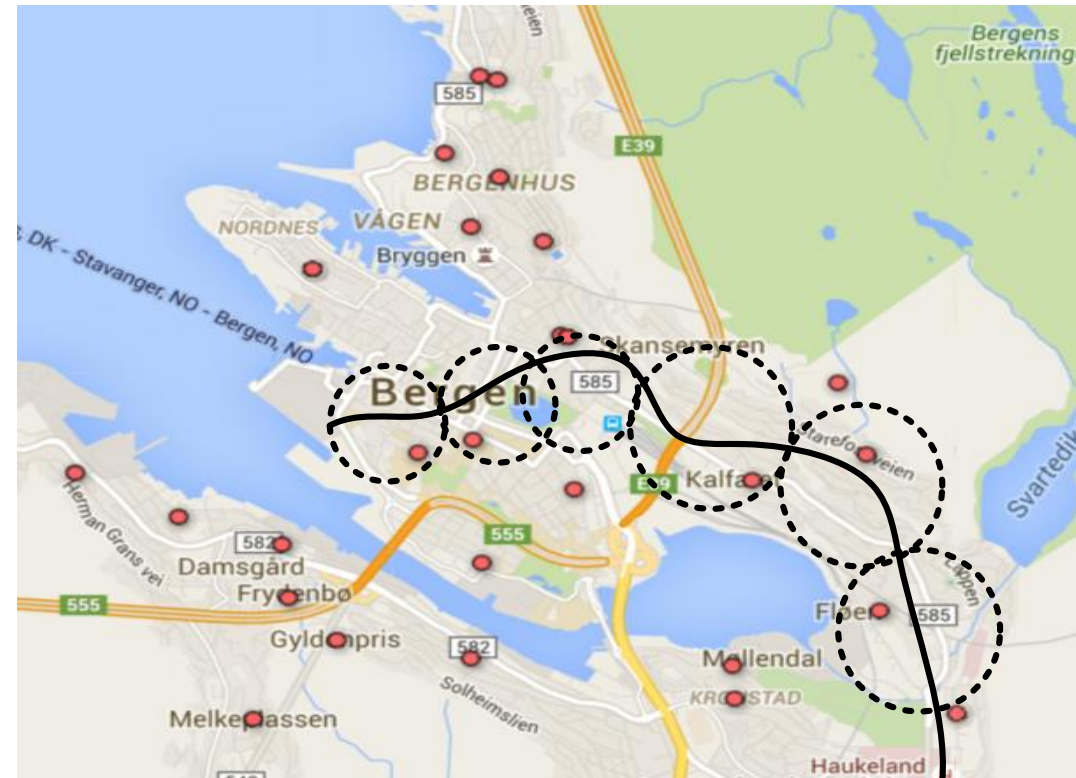
Oppgavestiller

- Oppgaven er en egendefinert oppgave gjennom Mjelde IKT ENK.
- Mjelde IKT eies og drives av Olav Alexander Mjelde. Foretaket tar primært oppdrag innen webutvikling og drift / vedlikehold av nettsteder.



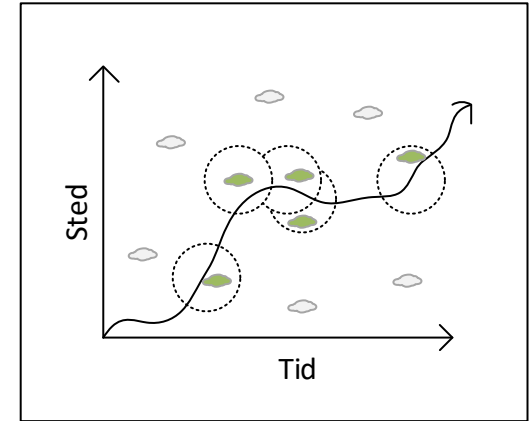
Problemstilling

- Hvordan kan man gjennom brukerens kontekst vise data som er relevant for brukeren, ved hjelp av en mobilapplikasjon?



Dagens løsninger

- «Kontekst» er en prototype for å vise kontekstuelle data, men det er nærliggende å sammenligne med applikasjoner for luftkvalitet og vær ved at det er datakildene som skal fungere i prototypen.
- Værapplikasjoner i Play Store viser ikke luftkvalitet for Norge.
- Luftkvalitetsapplikasjoner i Play Store viser ikke værdata og bruker ikke konteksten til brukeren for å finne data som er relevant.



Hvorfor oppgaven ble valgt

- «Kontekst» gir muligheten for å skaffe erfaring på en kontekstavhengig applikasjon som kan utvides med flere tjenester.
- Samfunnsnyttien av å kombinere luftkvalitet med værdata kan sies å være høy ved at luftkvalitet i stor grad glemmes når det ikke er vinter.
- Prototypen kan brukes til andre kontekstavhengige datakilder som kan knyttes til lokasjon og tid, dette gjør at oppgaven kan utvides med flere tjenester av samfunnsnyttig art.
- Oppgaven gir en stor bredde som betyr at man får realisert mange av fagene gjennomført ved HiST (nå NTNU).



Hvordan ble oppgaven løst

- Første steg i prosessen var å definere oppgaven, før problemdomenet ble analysert gjennom forstudierapporten.
- Forstudierapporten analyserte markedssituasjonen og behovene, samt andre faktorer som i kravdokumentet materialiserte seg som konkrete krav.
- Prosjektets aktiviteter ble laget i Gant diagram i Microsoft Project.
- Jevnlige nettmøter med faglærer, med referater.
- Timeføring ble gjort i Google Docs
- Ukentlige statusrapporter ble utarbeidet



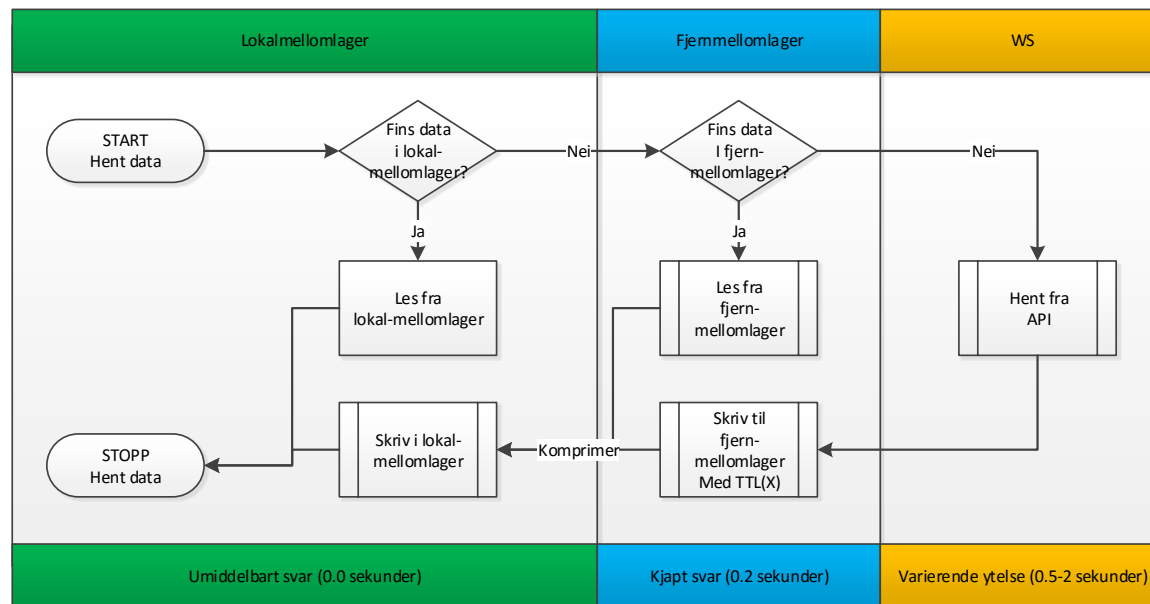
Hvordan ble oppgaven løst (forts.)

- Utviklingen ble gjort i iterative prosesser basert på milepælene.
- Android-utvikling har blitt gjort i Android Studio, testing har blitt gjort i emulator, samt på 3 mobiltelefoner og 1 nettbrett.
- Tjenestebussen i PHP ble utviklet i VIM i Linux konsollet
- Databasen (MySQL med MyISAM motor) ble utviklet gjennom MySQL WorkBench og MySQL konsollet i terminalen.
- Tjenestebussen ble satt opp hos Digital Ocean, som en VPS med LEMP stack (Linux, Nginx, MySQL og PHP) med støtte for HTTP/2.
- Landingside utviklet i Photoshop og brackets med JQuery og Bootstrap rammeverk. Google Maps mot Google Fusion Tables.



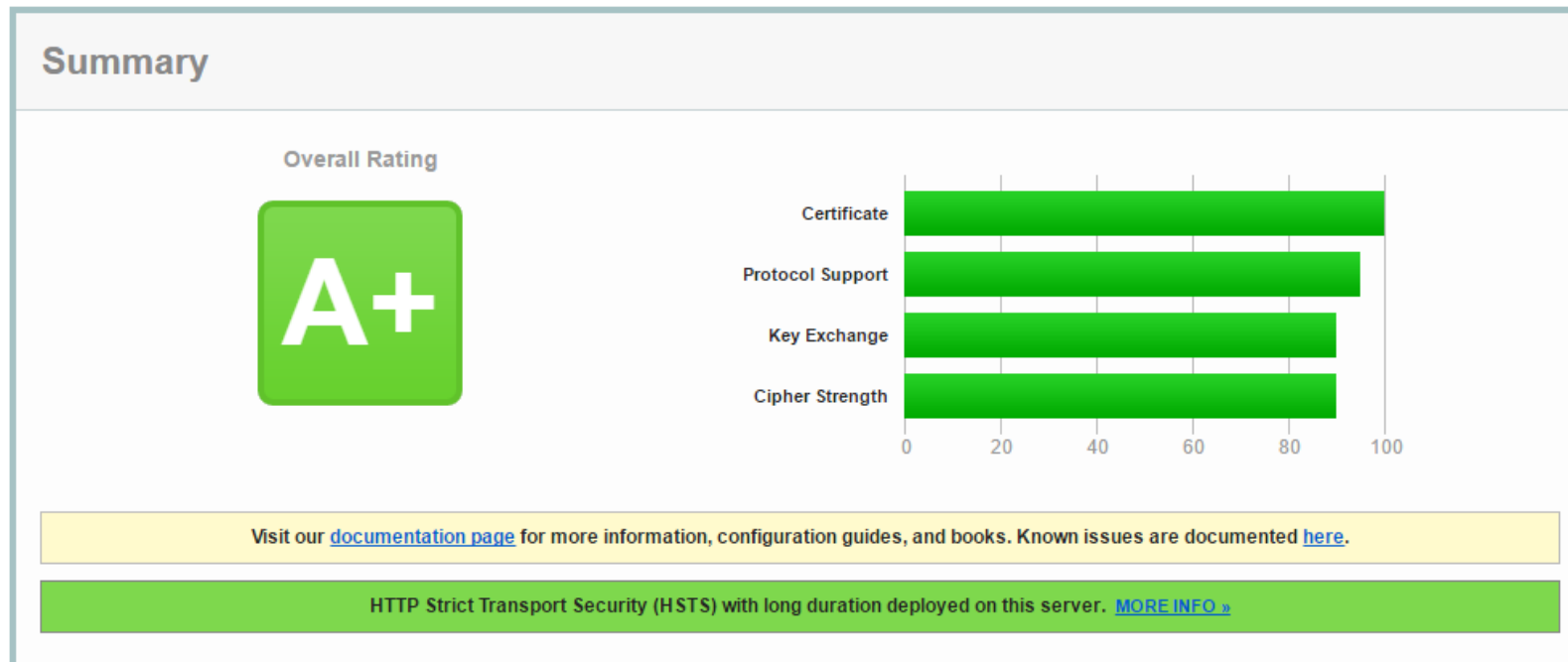
Hvordan ble oppgaven løst (forts.)

- Mellomlagringsprinsippet øker ytelsen vesentlig og understøtter kravene fra YR om mellomlagring av data i 20 minutter.
 - Redis benyttes for effektiv minnebasert mellomlagring.



Hvordan ble oppgaven løst (forts.)

- Kryptert kommunikasjon er løst gjennom letsencrypt.com sertifikater og er optimalisert opp til A+ nivå.



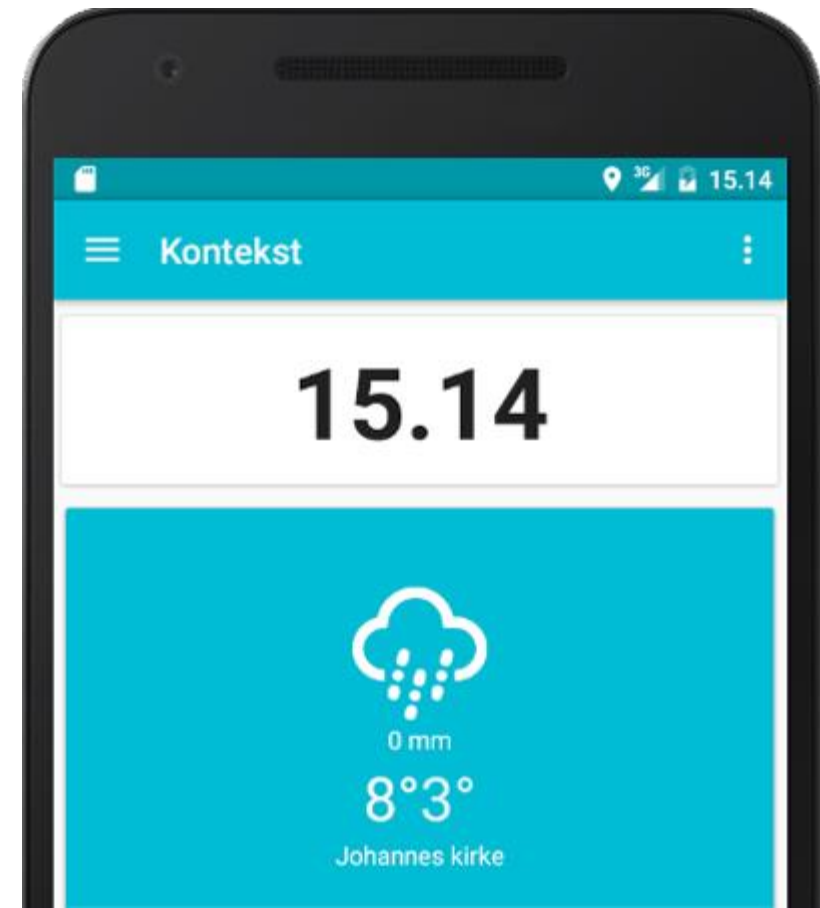
Hvordan ble oppgaven løst (forts.)

- Kompetanseheving i prosjektet ble gjennomført ved å lese dokumentasjon på Google sine utviklersider. Deltok også på en konferanse som omhandlet mellomlagring med Redis.
- Kompetanse tilegnet under studier hos HiST (nå NTNU) var veldig nyttig, særlig fag som: PHP, MySQL, Linux Systemdrift, Datamaskinteknikk, Drift av lokalnettverk, Objektorientert programmering og systemarbeid. Faget ITIL V3 var også greit å ha gjennomført. Oppgaven tar for seg mange lag i OSI-modellen og tar i bruk relasjonsdatabaser, geolokasjon og utvikling i PHP, samt Android (Java). Androidutvikling var nytt for meg, men tok java i 2001-2002.
- Filutveksling på Google Disk med faglærer.



Resultat

- Android Applikasjon «Kontekst»
- Material design grensesnitt som tilpasser seg tilgjengelig bredde føles moderne og lett.
- Asynkrone datakall gjør at applikasjonen ikke «låser seg»
- Mellomlagring av data i tjenestebussen gir god ytelse og etterlever YR sine krav.



Resultat (forts.)

- Grensesnittet bruker «RecyclerView» for god ytelse og mulighet for «uendelig» med datakilder.
 - Navnet kommer av at den kan gjenbruke grensesnitt for ny data.
- Man kan trekke ned listen for å laste ny data
- Brukere kan melde inn
 - Opplevd vær
 - Opplevd luftkvalitet
- Applikasjonens landingside nås fra menyen og har kart over datakilder.



Resultat (forts.)

- Applikasjonens landingside
 - Utviklet med responsivt grensesnitt.
 - Har kart over datakilder
 - Inneholder informasjon om applikasjonen på en lettlest måte.
 - Nås fra menyen i applikasjonen.
 - Kan også nås gjennom nettleser.



Kontekst

Din kunnskap er viktig for oss- derfor lar vi deg dele den

Det er mange faktorer som kan påvirke hvilke datakilder som er mest viktige for deg og din kontekst.

Kanskje har du lokale forhold som fjell, høyde over havet eller annet som gjør at nærmeste kilde ikke er den som er best for deg å se på.

Når flere brukere melder inn den opplevde situasjonen kan vi bruke dette til å filtrere ut de mest optimale stasjonene i fremtiden. Derfor er din kunnskap viktig!



Kontekst - endelig en APP som skjanner din kontekst



Med ~48.888 målepunkter i verden, har vi nesten full kontroll på hva som skjer rundt deg.

Vi bruker høykvalitets kilder som YR og NILU. Kontekst henter sine data gjennom en egen tjenestebuss som finner den riktige dataen ut fra brukerens kontekst.

Tilbakemeldingsfunksjonen er også knyttet til tjenestebussen, den sender ingen data til YR eller NILU.

Kontekst er laget av oss, for deg med lidenskap (og ~388 timers arbeid).

Kontekst virker på 99% av Android enheter - også nettbrett!

Vi har segmentert oss etter Android 2.3 og høyere, som betyr at kontekst støtter 99% av aktive Android telefoner.

Grensesnittene er dynamiske av natur og vil derfor til en viss grad skalere seg opp også på nettbrett.





Applikasjonen støtter også flere språk: Engelsk, Norsk (bokmål), Norsk (nynorsk), Dansk og Svensk.

- Språk velges automatisk basert på valgt språk i den mobile enheten.



Resultat (forts.)

- Språk velges automatisk basert på enhetens valgte språk.
 - Norsk (Bokmål og nynorsk), Dansk, Svensk og Engelsk.
 - 35 ord og setninger med 140 oversettelinger gir stort nedslagsfelt

Key	Default Value	Untr...	 Danish (da)	 Norwegian Bokmål (nb)	 Norwegian Nynorsk (nn)	 Northern Sami (se)
action_settings	Settings	<input type="checkbox"/>	Optioner	Instillinger	Instillinger	Val
app_name	Kontekst	<input type="checkbox"/>	Kontekst	Kontekst	Kontekst	Kontekst
aq0	Little	<input type="checkbox"/>	Lidt	Lite	Lite	Liten
aq1	Moderate	<input type="checkbox"/>	Moderat	Moderat	Moderat	Måttlig
aq2	High	<input type="checkbox"/>	Høj	Høyt	Høyt	Högt
aq3	Very high	<input type="checkbox"/>	Meget høj	Svært høyt	Svært høyt	Mycket hög
aq_0	Not sure	<input type="checkbox"/>	Ikke sikker	Usikker	Usikker	Osäkert
aq_1	Good air quality	<input type="checkbox"/>	God luftkvalitet	God luftkvalitet	God luftkvalitet	God luftkvalitet
aq_2	OK air quality	<input type="checkbox"/>	Ok luftkvalitet	Ok luftkvalitet	Ok luftkvalitet	OK luftkvalitet
aq_3	Bad air quality	<input type="checkbox"/>	Dårlig luftkvalitet	Dårlig luftkvalitet	Dårlig luftkvalitet	Illu luftkvalitet
aq_4	Terrible air quality	<input type="checkbox"/>	Frygtelige luftkvalitet	Forferdelig luftkvalitet	Forferdeleg luftkvalitet	Avskryärda luftkvalitet
aqnull	Unknown	<input type="checkbox"/>	Ukendt	Ukjent	Ukjent	Okänd
btnOk	Ok	<input type="checkbox"/>	Ok	Ok	Ok	Ok
cancel	Cancel	<input type="checkbox"/>	Ophæve	Avbryt	Avbryt	Annulera
chooseone	Choose one	<input type="checkbox"/>	Vælg en	Velg en	Velg ein	Välj en

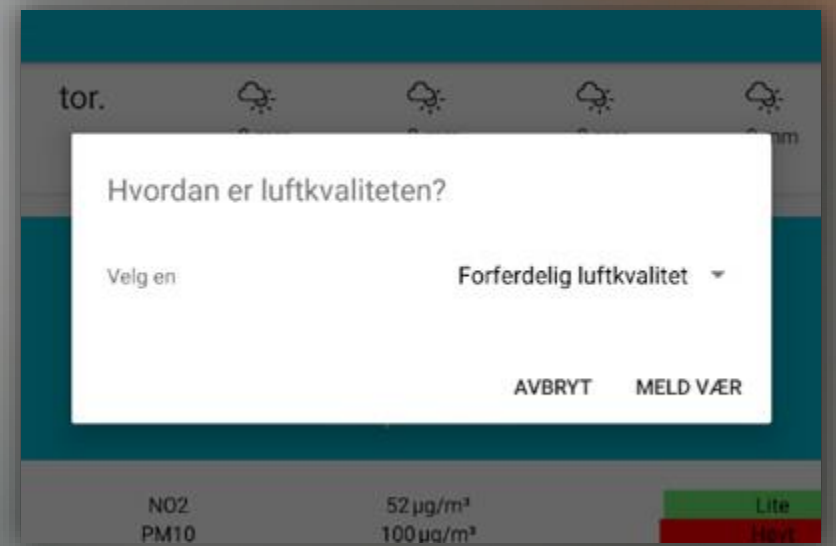


Resultat (forts.)

- Ytelsestesting av Redis mellomlagring i tjenestebuss viser ekstrem ytelse
 - PING_INLINE: 108695.65 requests per second
 - PING_BULK: 112359.55 requests per second
 - SET: 109890.11 requests per second
 - GET: 111111.11 requests per second
 - INCR: 90090.09 requests per second
 - LPUSH: 128205.12 requests per second
 - LPOP: 129870.13 requests per second
 - SADD: 67114.09 requests per second
 - SPOP: 61728.39 requests per second
 - LPUSH (needed to benchmark LRANGE): 60606.06 requests per second
 - LRANGE_100 (first 100 elements): 23752.97 requests per second
 - LRANGE_300 (first 300 elements): 9149.13 requests per second
 - LRANGE_500 (first 450 elements): 8361.20 requests per second
 - LRANGE_600 (first 600 elements): 5896.23 requests per second
 - MSET (10 keys): 56497.18 requests per second







Videre arbeid

- Utvidelser som kan bygge videre på konseptet til Kontekst
 - Flere datakilder
 - Flo/fjære, tur og løypedata og andre kilder som er hensiktsmessig å knytte mot lokasjon
 - Brukere kan legge til egne områder
 - Hytten, ferien og andre steder.
 - Sosiale interaksjoner
 - Sjekk-inn, visning av innrapporter vær og luftkvalitet, mulighet for å «dele» mot sosiale medier.
 - Skreddersøm
 - La brukerne velge fargetema i app
 - Statistikk/analyse
 - Innrapporterte værphenomener
 - Innrapporterte værphenomener opp mot varslet vær / luftkvalitet
 - Vekting av prioriterte stasjoner basert på innrapportert vær i forhold til meldt vær.
 - Sanntidsvisning av innrapportering og sjekk av vær på landingssiden (server sent events)



Videre arbeid (forts.)

- Utvidelser som kan bygge videre på konseptet til Kontekst (forts.)
 - Android Wear (smartklokke)
 - Widgets
 - Teste hvordan grensesnittet virker på Android TV
 - Teste hvordan grensesnittet virker på Android Auto (bilstereo)
 - Lage mulighet for å ta bilder i app og merke med en valgt datakilde
 - Lage bilder med luftkvalitet, vær, flo/fjære og andre ting som er relevant i brukers kontekst.
 - Lage heatmaps på landingssiden som viser hvor det er mest bruk av applikasjonen.
 - Lage eller finne relevante ikoner til luftkvalitet for økt estetisk stimulans.

