# IN2000

Solbrent



**Team 32**

Luka Momcilovic (lukamo)

Magnus Løkken Nilsen (magnusln)

Malin Saleh-Jan (malinsal)

Erik Anker Kilberg Skallevold (easkalle)

Sondre Andersen (sondan)

**Veileder**:

Sergey Jakobsen (sergeyj)

Synne Gåseidnes Thorsen (synnegth)

[**IN2000**](#_1zs5883umyx5) **1**

[1. Introduksjon](#_yimzyd89k9wm) 3

[**1.1 Presentasjon av case**](#_bkn9cgufmltx) **3**

[**Vitamin D**](#_bqqrnlo3z1jp) **4**

[**Hudkreft**](#_cqc8souityuy) **5**

[**1.2 Teamet vårt**](#_khda6i52fh0n) **5**

[**1.2.1 Teammedlemmene**](#_pzj8rcq1s9o6) **5**

[**1.3 Prosjektplan**](#_2enncbh0t0m8) **7**

[**2. Brukerdokumentasjon**](#_y5u8beeddqk8) **8**

[**2.1 Overordnet beskrivelse av løsning**](#_p67bmbrdnikf) **8**

[**2.1.1 Hvordan muliggjøres tjenesten**](#_bujd5bmnx19x) **8**

[**2.1.2 Hovedfunksjonalitet**](#_hg6ro57y8di2) **8**

[**2.2 Målgruppen**](#_tfqpqu80tygm) **9**

[**2.3 Plattformer**](#_8kdbhipo0ksd) **9**

[**2.4 Aksessering av applikasjon**](#_8nk2t9ecwrpn) **10**

[**2.5 Teknologier brukt**](#_ovh394p2kz9j) **11**

[**3. Kravspesifikasjon**](#_jev4836hwyah) **12**

[**3.1 Funksjonelle krav**](#_kk0s8hixe0ak) **12**

[**SKAL-krav**](#_f5dfh0jva6v1) **14**

[**BØR-kravene.**](#_raa9hloxurne) **15**

[**KAN-kravene.**](#_49efuh8w3ndq) **15**

[**3.2 Ikke-funksjonelle krav**](#_85zz1lpnzqh0) **15**

[**3.2.1 Produktkrav**](#_2anqt2px6bdd) **16**

[**Responstid**](#_88slha1c5gqz) **16**

[**Antall samtidige brukere**](#_iiuf7w5szunh) **16**

[**Lagringsplass**](#_hli2oeouej0l) **17**

[**Sikkerhet/personvern**](#_w8vaula2o2cm) **17**

[**3.2.2 Organisasjonskrav**](#_y806r6vpyi2a) **18**

[**3.2 Use case (og user-stories?)**](#_xmuq6km0ovo) **19**

[**3.3 Klassediagram**](#_xxx9n22byou0) **21**

[**3.4 Sekvensdiagram**](#_47i066bbue7a) **21**

[**3.5 Design pattern**](#_8xnzq1v8b2p7) **21**

[**4. Produktdokumentasjon**](#_3f9403twm3ll) **21**

[**4.1 Beskrivelse av løsning**](#_wf1u0kpfqe8m) **21**

[**4.2. Liste over app egenskaper**](#_vehlqs4sqqqc) **21**

[**4.3 API-nivå**](#_h4wnhm1fogf8) **22**

[**5. Testdokumentasjon**](#_enhs7fkpa46a) **22**

[**5.2 Testverktøy**](#_1szy6pq35lbj) **22**

[**5.3 Planlegging**](#_kwxu25ygoqb7) **23**

[**5.4 Dynamisk / Statisk testing (?)**](#_gsklh1kc68wp) **23**

[**5.4.1 Enhetstesting**](#_qa282srkatf3) **23**

[**6. Prosessdokumentasjon**](#_gph1yax2651f) **23**

[**6.1 Sprint 1**](#_rq3tvtivubzq) **23**

[**6.1.1 Planlegging og struktur**](#_ohuat4j10sl9) **23**

[**6.1.1.1 Snakke om første gruppemøte**](#_9i596q2lenil) **23**

[**6.1.1.2 Valg av case**](#_drq26d5o80r) **23**

[**6.1.2 Verktøy**](#_k8mo6dba7dg1) **23**

[**6.1.2.1 GitHub**](#_tu3npzs6za8i) **23**

[**6.1.2.2 Slack / Facebook**](#_bmk8r45ryht5) **23**

[**6.1.3 Valg av smidig utviklingsmetode**](#_qnrk0cco4f4i) **23**

[**6.1.4 Rollefordeling**](#_j32ajislfjg0) **23**

[**6.1.5 Brukerundersøkelser**](#_ayuaw28r3pr1) **23**

[**6.2 Sprint 2**](#_wf0oeahegopy) **23**

[**6.2.1 Kravspesifikasjon**](#_4c77d7ryp0wr) **23**

[**6.2.2 Datainnsamling**](#_hah5xmyte95q) **23**

[**6.2.2.1 Brukerundersøkelse**](#_za5595id9iz) **23**

[**6.2.2.2 Intervju**](#_ok631rb5h9cc) **23**

[**6.2.3 Dataanalyse**](#_udg3s2t82lz2) **23**

[**6.3 Sprint 3**](#_ha74zsbb77qn) **24**

[**6.3.1 Brukbarhetstesting**](#_ydko5wumnc9z) **24**

[**6.4 Sprint 4**](#_ophlkdcdrn0n) **24**

[**6.5 Sprint 5**](#_kpouaug8azi7) **24**

[**6.5.1 Implementering av funksjonalitet**](#_j3gn5y7x5zxq) **24**

[**6.6 Sprint 6**](#_lcap7y4f4sw8) **24**

[**6.6.1 Brukerevalusering av endelig applikasjon**](#_7j4un8gln5wm) **24**

[**7. Refleksjon**](#_7dnjz31jt9kp) **24**

[**7.1 Utfordringer**](#_kzx5z8md8n55) **24**

[**7.2 Lærdommeer**](#_ucbxl82l2lc8) **24**

[**8. Referanser**](#_5eclcsz2cmnf) **24**

[**9. Vedlegg (figuroversikt, intervju, sprint-loggføring, referater?)**](#_9fg97w7p2urg) **24**

## 1. Introduksjon

### 1.1 Presentasjon av case

I vårt prosjekt så har vi bestemt oss for å velge et åpent case der temaet skal omhandle og fokusere på UV-lys (ultrafiolett lys). Temaet UV-lys blir sett på i konteksten; hudkreft, vitamin-D beregning, solbrent tid og anbefaling av solkrem.

I oppstartsfasen ble forskjellige lukkede case vurdert, blant annet gjødsel-, sykkelstier- og fiskeoppdrett-casene, men vi ble raskt enige om at en app som fokuserer på UV-lys var mest interessant. Vi anså også denne ideen som nokså relevant på nasjonal-nivå da Norge befinner seg på tredjeplass i forekomst av hudkreft i verden.*[[1]](#footnote-0)*

Vi betraktet dette da også som et insentiv å lage en tjeneste som gir brukere muligheten å ta noenlunde informerte valg når de soler seg både i Norge og i utlandet.

Med dette som utgangspunkt ønsket vi å utvikle en app som gir brukere informasjon om hvor lenge de kan befinne seg i solen med gitt hudtype før de ble solbrent.   
I tillegg ønsket vi at brukere skal kunne selv se hvor mye D-vitamin de produserer per time under gitte UV-forhold, med gitt hudtype og ved valgt påkledning.   
Sist, men ikke minst, anså vi det også som nyttig at brukere skal kunne bli anbefalt en egnet solkrem tilpasset gitte værforhold og gitt hudtype en person innehar.

#### Vitamin D

Det er flere årsaker til at vi valgte dette caset. Under covid-pandemien var det mye diskusjon i nyhetene om Vitamin-D og dets innvirkning på immunsystemet og viktigheten av å få nok Vitamin-D gjennom kosten eller solen. Hvor mye sol hver person må utsette seg for - for å få daglig anbefalt dose D-vitamin - er allikevel en komplisert problemstilling som de fleste ikke vet svaret på.   
Det er mange faktorer som har innvirkning på D-vitamin syntese og dette gjør det derfor vanskelig for hverdagslige mennesker å finne frem til lettlest og nødvendig informasjon når de skal ta informerte valg omkring dette.

Mengden produsert D-vitamin i huden kompliseres av en rekke forskjellige faktorer. Det er mange kofaktorer som gjør tidsberegningen av D-vitamin syntese komplisert. Solvinkling, skyforhold, hudfarge, enzymer, UV-indeks, mengde UVB-stråling, mengden nødvendig D-vitamin substrat i huden og en rekke andre påvirkende faktorer.   
Vi vurderte derfor at det var et samfunnsbehov for en tjeneste som forenkler denne prosessen, og gjør det lettere for folk og faktisk finne frem til den nødvendige og pragmatisk-nyttige informasjonen. ([kilde](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4882679/))

Norge er dessuten også et land som befinner seg svært nordlig og har en lang og mørk vintertid, noe som gjør D-vitamin mangel til et viktig og sentralt tema for mange.   
Ved å gjøre denne informasjonen lettere tilgjengelig for hverdagslige mennesker så kan vi være med på å bidra til en reduksjon i Vitamin-D mangel i befolkningen og potensielt redusere andre mulige komorbiditeter av dette.

#### Hudkreft

En annen årsak til at vi valgte dette caset er at man ifølge WHO har funnet en sterk assosiasjon mellom hyppige solvaner/overdreven UV-eksponering og hudkreft.   
Man har fastslått at en av de vanligste årsakene til hudkreft i verden i dag er skade påført arvestoffet i hudcellene av UV-lys ([2](https://www.nhs.uk/conditions/melanoma-skin-cancer/causes/)).Norge er et land som befinner seg på verdenstoppen i forekomst av hudkreft blant befolkningen. Dette gjør det derfor særlig viktig at folk i norden er mer informert og har god tilgang på ressurser som gjør dem i stand til å ta gode valg rundt solevanene sine.

80-90% av Vitamin-D produksjonen hos mennesker skjer som følge av eksponering til UVB-stråling, men dette er også den typen UV-strålingen som er hovedårsaken til hudkreft. Det er derfor viktig at man finner en balanse mellom tilstrekkelig Vitamin-D syntese og risikoen for utvikling av hudkreft. ([kilde](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4882679/))

Vi mener derfor at ved å utvikle denne appen så kan vi bidra til å skape balanse i samfunnet omkring akkurat dette og forbedre folkehelsen.

### 1.2 Teamet vårt

Teamet vårt består av fem medlemmer: Luka Momcilovic, Malin Saleh-Jan, Magnus Løkken Nilsen, Erik Anker Kilberg Skallevold og Sondre Andersen.

#### 1.2.1 Teammedlemmene

**Gruppebilde**

**Sondre Andersen** går 4 semester på en Bachelor i Informatikk: Programmering og Systemarkitektur. Han har fra før en grad i økonomi/business og er interessert i skjæringspunktet mellom teknologi og økonomi. Fra før har han erfaring med webprogrammering, dette inkluderer bruk av rammeverket React for frontend samt Flask og Express for backend. I prosjektet har han bidratt med å kode for å implementere funksjonaliteten i appen, skrive rapport, skrive tester og deltatt i de mange møtene vi har hatt om design og UI. På fritiden liker han å være med venner, trene og har i det siste blitt litt hekta på padel tennis.

**Luka Momcilovic** tar en Bachelor i Programmering og Systemarkitektur ved Universitetet i Oslo og har fra tidligere en Bachelor i Bioingeinør ved Oslomet - Storbyuniversitet. Han er en sterk proponent for bruk av Javascript i alle kontekster og er veldig glad i objekt-orientert programmering. I prosjektet har han vært ansvarlig for utviklingen av API, backend-kode, skriving av tester, skriving av rapport og komme med konstruktiv kritikk til design. På fritiden liker han å klage på teknisk gjeld, fantasere om goldplating, tilbringe tid med venner, gå turer, spille piano og trene.

**Malin Saleh-Jan** går en fritt sammensatt bachelor med emner innen både design og programmering. Hun har erfaring med ulike programmeringsspråk, samt interaksjonsdesign, og har delansvar for datainnsamling, organisering av rapport og rapportskriving i prosjektet. Fritiden bruker hun oftest på å være med venner, lese bøker eller være kreativ, hvorav hun bl.a maler, syr klær, dj-er og tar bilder. Hun har også en enorm interesse for mote.

**Magnus Løkken Nilsen** tar en bachelor i programmering og systemarkitektur ved Universitetet i Oslo.

**Erik Anker Kilberg Skallevold** tar en bachelor på Informatikk: Design, bruk og interaksjon og har gjennom studiet fått mye erfaring med brukerorientert design. Erik er stor tilhenger av funksjonelt design og mener at brukeren hele tiden burde få se hva enhver artefakt gjør, uten noen gimmicker, noe som kanskje krasjer med moderne, flatt design. I prosjektet har han vært ansvarlig for å lage designet på applikasjonen i samarbeid med gruppa (som har mer fokus på moderne utforming).

### 1.3 Prosjektplan

| **Prosjektplan** | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mars** | | | | **April** | | | | **Mai** | |
| **1 - 2** | | **3 - 4** | | **5 - 9** | | | | **10 - 11** | |
| - Fullføre oblig 3  - Bestemme case  - Sette opp KanBan-board og sub-boards  - Sette opp kommunikasjonsverktøy og git/hub  - Finne relevante API-er  - Datainnsamling  - Lage kravspesifikasjon (funksjonell og ikke-funksjonell)  - Skrive use-case/use-stories  - Bestemme oss for smidig-metode, velge designpattern (MVVM) og skrive ned verktøy  - Finne målgruppe/demografi  - Første kladd av kravsspesifikasjon  - Lage en overordnet struktur for rapporten og lage innholdsfortegnelse  - Idemyldre | | **-** Samle inn data (intervju, spørreundersøkelse)  - Oppdatere kravspesifikasjon  - Finjustere rapport og prosjektplan basert på oppståtte krav/arbeidsoppgaver  - Bli bedre kjent med API/skrive tester for API  - Starte utvikling/design av app på overordnet nivå  - Videre utforming av wireframes i Procreate og Balsamiq  - Testing av wireframes med brukes  - Justere wireframes basert på brukerundersøkelse  - Fortløpende laging av diagrammet osv. basert på kode  - Ferdigstille design diagrammer  - Ha XML-ene-templatene ferdig | | **-** Utvikle app (frontend og backend)  - Oppdatere rapport, skrive rapport  - Fortløpende laging av diagrammet osv. basert på kode  - Ferdigstille kravspesifikasjon  - Oppdatere rapport  - Kjøre integrasjonstest og enhetstester | | | | **-** Ferdigstille app  - Ferdigstille rapport  - Retrospektiv studie  - Gjennomgang av rapport, finskriving, rettskriving, lese igjennom | |

## 2. Brukerdokumentasjon

### 2.1 Overordnet beskrivelse av løsning

Løsningen vi tilbyr på caset/problemstillingen kan beskrives som en spesialisert værapp.   
Ved hjelp av flere API-endepunkter, brukerdefinert informasjon og forskningsartikler tilbyr den brukere oversiktiglig og praktisk-nyttig informasjon relatert til problemstilling. Løsningen skal unngå svada og fokusere på nytte kontra tekniske detaljer.

#### 2.1.1 Hvordan muliggjøres tjenesten

Den overordnede beskrivelsen av løsningen vil primært bli muliggjort av API-et til Meteorologiske institutt, data fra forskningsartikler og et egenprodusert API-endepunkt. API-et til MET tilbyr daglig og timebasert værdata hvor det blant annet også gis informasjon omkring UV-forholdene ved en gitt lokasjon.

Hovedinformasjonen som ekstraheres fra API-et til MET er styrken på UV-strålingen på en gitt lokasjon. Denne styrken rapporteres som en ultrafiolett-indeks (UVI). Dette er en internasjonal og standardisert måleenhet for styrken på ultrafiolett lys og sier noe om hvor stort potensiale den har til å gjøre noen solbrent på et gitt tidspunkt på et gitt sted. Jo høyere UV-indeks, jo mindre tid tar det for at skade påarvestoffet, huden og øynene inntreffer, jo mindre UV-indeks, jo lenger tid tar det ([kilde](https://snl.no/ultrafiolett_str%C3%A5ling)).

I tillegg til API-et til MET (Meteorologisk Institutt) så vil tjenesten også ta i bruk ett egenprodusert API-er som bestemmer hudtype basert på et opplastet bilde av huden til brukeren. Dette er en AI-basert løsning. Beregninger av Vitamin-D produksjon og solkrem anbefaling vil basere seg på publiserte forskningsartikler og vil være en løsning innebygd i appen lokalt.

#### 2.1.2 Hovedfunksjonalitet

Løsningen på caset skal tilby følgende funksjonalitet:

* Appen finner mengde D-vitamin produsert per time for gitt Fitzpatrick-type basert på UV-indeks og grad av påkledning.
* Appen viser farlighetsgrad for en gitt UV-indeks
* Appen finner tiden det tar å bli solbrent med valgt Fitzpatrick-type og gitt UV-indeks
* Appen anbefaler solkrem basert på Fitzpatrick-type og UV-index
* Appen gir push-varslinger for når hudkrem bør påføres på nytt etter påføring tidligere
* Appen gir mulighet å søke på andre plasser i verden som ikke er der du er for å få samme informasjon
* Appen skal gjøre en grov bestemmelse av de ovennevnte funksjonaliteten der hovedfaktorene som tas i bruk i beregningene er hudfarge, UV-indeks og grad av påkledning.
* Appen skal ha et enkelt språk slik at hverdagslige mennesker kan dra så mye nytte av appen og informasjonen den tilbyr som mulig
* Appen skal tilby en hudbestemmelsestjeneste (i form av en API) for de menneskene som ikke er sikre på hvilken konkret hudtype de har.

### 2.2 Målgruppen

Målgruppen vi bestemte oss for å utvikle en applikasjon for var studenter i aldersgruppen 18-25 år. Dette er på bakgrunn av at det er en tilgjengelig gruppe, og fordi det var den gruppen som viste mest initiativ og interesse i vår spørreundersøkelse. Det å ha en definert og tilgjengelig målgruppe var ideelt for oss for å kunne tilegne nok data uten å bruke alt for mye tid på datainnsamlingen. Alt i alt så ble det enklere for oss å forstå hva slags krav vi skulle yte, implementere og videreføre, basert på data fra vår spesifikke målgruppe.

### 2.3 Plattformer

Vi fastslo relativt tidlig i prosessen at vi ønsket å utarbeide en applikasjon som var egnet for Android enheter, utviklet i Android Studio. Dette gjorde at vi fikk satt et hovedfokus på å yte den faktiske funksjonaliteten applikasjonen vår ville ha, som vi prioriterte i større grad enn at den ville støtte ulike plattformer. Det ble vurdert om vi skulle ta i bruk andre plattformer, men på bakgrunn av at ikke alle på gruppen hadde erfaring med å bruke andre relevante rammeverk, resulterte det i at vi kun forholdt oss til Android for å minimere tidsbruken. Dessuten, så ville det vært uforutsigbart for oss å anslå hvor tidkrevende en innføring av et multiplattform rammeverk ville vært, og om vanskelighetsgraden var verdt bryet. Derfor konkluderte vi med at implementasjon av et multiplattform rammeverk var unødvendig.

Valget vårt ble også avgjørende på bakgrunn av brukerstatistikk. På verdensbasis så benyttes Android operativsystemet av 88% av alle mobiltelefoner, i motsetning til 12% for iOS. Det er dog viktig å påpeke at denne skjevheten ikke eksisterer i Norge. I Norge så er brukerandelen på hver side (Android/iOS) omtrent 50/50[[2]](#footnote-1), noe som gjør valget vårt om å fokusere kun på Android mindre stadfast. Til tross for dette, så anser vi det også som nyttig å appellere til befolkningen utenfor Norge, da appen ikke er spesifikt for den norske befolkning[[3]](#footnote-2). Derfor velger vi å bare fokusere på Android.  
I fremtiden ville det dog vært rimelig å utvikle appen også for iOS-plattformer.

### 2.4 Aksessering av applikasjon

*\*Hvilke plattformer kjører applikasjonen på? Hvordan og hvor kan brukere aksessere applikasjonen?\**

*-*Github? Appstore (androids appstore?)?

I en gruppe med 5 personer kan det være vanskelig å finne møtetider hvor alle har muligheten til å delta. Flere i gruppen har jobb som krever mye tid og alle har forelesninger de også skal på. Gruppen kom til slutt fram til 2 møtetider som fungerte fint for alle. Dette består av et slags hovedmøte på tirsdager hvor vi alle møtes fysisk og jobber og diskuterer sammen. Dette møtet skal vare i maks 4 timer. I tillegg har vi et digitalt standup møte hver fredag på maks 30 minutter. Her forteller hvert gruppemedlem om hva de har gjort så langt denne uken og eventuelle problemer som har oppstått. Hvis det blir noe tid til overs brukes det til å videre diskutere prosjektet. Gruppen har også satt av 30-60 minutter annenhver tirsdag til et retrospektivt møte hvor vi alle diskuterer hvordan arbeidet har fungert så langt og hva vi kan gjøre bedre framover. Gruppen kommer til å skrive et møtereferat av hvert eneste møte om alt vi har gjort og diskutert. Denne rollen skal rulleres.

I dette prosjektet kommer gruppen til å bruke den smidige metodikken Scrumban. Gruppen bruker sprinter på 2 ukers lengde. I tillegg bruker gruppen et kanban-board gjennom nettsiden Trello. Vi kommer ikke til å bruke noen definerte roller i prosjektet. Ansvarsfordelingen vil stort sett være likt, med noen forskjeller her og der. Alle kommer til å ha like mye ansvar på rapporten. Vi er alle ansvarlige for å skrive på rapporten, se over vårt eget og andres arbeid på rapporten og hjelpe til med rettskriving. Alle i gruppen skal også kode, og vi er alle selv ansvarlig for å skrive enhetstester av koden vår. Andre i gruppen er derimot ansvarlig for å etterpå se over koden. Prosa studentene Luke, Sondre og Magnus er mer ansvarlige for at backend biten av programmet fungerer, mens de mer Design studentene Erik og Malin er mer ansvarlige for frontend, design og datainnsamling. Men alle i gruppen er ansvarlige for alle deler av programmet.

### 2.5 Teknologier brukt

## 3. Kravspesifikasjon

*Her vil vi beskrive de ulike kravene vi satt til applikasjonen vår, både de funksjonelle og de ikke-funksjonelle. I tillegg kommer brukerhistoriene som kravene er basert p˚a, og hensyn til universell utforming.*

TODO kravspek:

* (skal vi ha med brukerkrav og systemkrav også?)
* (domene krav?)
* WCAG (accessibility) krav?
  + contentDescription for hver element

### 3.1 Funksjonelle krav

(definisjon?)

*Funksjonelle krav er handler om hva systemet skal gjøre, dvs. hvilke funksjoner og/eller tjenester som skal være tilgjengelig for brukeren. De funksjonelle kravene skal også spesifisere hvordan applikasjonen svarer på bruker-input.*

I dette prosjektet er tid en knapp faktor så vi har valgt å prioritere kravene våre, slik at vi kan fokusere på å få på plass den viktigste funksjonaliteten først (SKAL-krav) før vi går videre på krav som vi gjøre appen bedre men som ikke er strengt nødvendig (BØR-krav). Og tilslutt, dersom vi har tid tar vi fatt på KAN-kravene som er funksjonalitet som vil gi en bra finish på produktet men som ikke er strengt nødvendig for at appen skal kunne tilfredsstille de behovene vi ønsker å fylle med appen.

(usikker på om dette bør stå under krav eller prosess?)

Når vi har prioritert har vi også lagt vekt på å fullføre såkalte lavthengende frukter først, slik at vi kommer raskt i gang og har noe å vise til i blant annet brukerundersøkelser og ikke minst for å få en følelse i teamet om at vi har bra progresjon. Siden dette også er et prosjekt der vi alle må lære mye nytt så betyr det også at det vi lærer ved å gjennomføre de lettere oppgavene også blir viktig når vi senere tar fatt på mer avanserte krav.

Under har vi listet opp de funksjonelle kravene for appen (notasjon: naturlig språk):

| SKAL | Systemet skal vite hvor brukeren befinner seg eller gi brukeren mulighet til å velge en lokasjon.  Systemet skal gi en basic værmelding (temp, nedbør, vær, lokasjon, solstyrke i dagligtale)  Systemet skal vise nivå av UV-stråling der brukeren befinner seg (evt. valgt lokasjon)  Systemet skal anbefale en solkrem basert på vær og hudtype  Systemet skal vise mengde D-vitamin produsert per time (basert på hudtype) eller hvor lang tid det tar å få dagsbehoved oppfylt?  Systemet skal anslå tiden det vil ta å bli solbrent  Systemet skal finne korrekt Fitzpatrick-type (AI)  Systemet bør huske på tidligere Fitzpatrick-type (via settings?)  Navigasjonslinje - navigere mellom activities. |
| --- | --- |
| BØR | System-UI bør endres basert på værforhold  Ha en graf som viser prognose for UV-resten av dagen. |
| KAN | Systemet kan vise UV-soner i kart  Gi en liste over stedene i Norge/Verden der UV-strålingen er sterkest??  Systemet kan gi notifikasjon om sterk UV der man befinner seg  Gi notifikasjon til bruker om at ny runde med solkrem er anbefalt |

#### SKAL-krav

**SKAL-kravene** handler i hovedsak om å bruke data fra MET-APIet til å gi brukeren informasjon om nivå av UV-stråling. Altså er UV-varsling det viktigste kravet vårt. Det har ført til at vi har måttet lære endel om domenet ved å sette oss inn i UV-stråling og hvordan ulike nivåer av stråling relaterer seg til solbrenthet og hudkreft risiko.

I tillegg til å gi brukeren informasjon om nivået av UV-stråling der brukeren befinner seg (evt. valgt lokasjon) så vil vi også gi brukeren supplerende informasjon som hvor langt tid det tar før man blir solbrent og hvor lang tid det tar før man har fått en dagsdose D-vitamin ved nåværende nivå av UV-stråling. Vi tenker denne supplerende informasjonen er nødvendig da brukerne av appen vår i hovedsak er studenter og andre personer uten eller med svært varierende kunnskap for hvordan UV-stråling påvirker kroppen.

Ettersom UV-stråling påvirker mennesker ulikt avhengig av hudfarge så ønsker vi å skreddersy informasjon om parameterne diskutert i avsnittet over basert på den hudtypen brukeren har.

For å skaffe informasjon om brukerens hudtype planlegger vi å tilby to ulike alternativer:

1. Som standard ønsker vi at brukeren tar et bilde av armen sin, og basert på det skal appen kunne anslå hudtype (mer om implementasjon i kapittelet om ikke-funksjonelle krav).
2. Dersom brukeren ikke ønsker å ta bilde av armen sin, eller appen vår skulle gi et rart svar, så kan brukeren også manuelt velge hudtype basert på 5 (6?) ulike fargetoner.

UV-stråling er kun interessant (for målgruppen) i en kontekst av å være utendørs (enten det gjelder soling, bading, tur i fjellet, påskeskitur eller noe annet). Av den grunn ønsker vi også å tilby brukerne en enkel “now-cast” av været i området brukeren befinner (evt. valgt lokasjon).

Systemet må også ha en navigasjonslinje som gjør at brukeren kan navigere mellom aktivitetene i appen (hovedsiden, innstilling, kart? og mer?). Denne navigasjonslinjen skal ligge nederst på skjermen slik at brukeren lett kan nå den.

#### BØR-kravene.

Vi ønsker at grafikken på hovedsiden skal være dynamisk og avhengig av den nåværende vær/UV-situasjonen.

#### KAN-kravene.

En tilleggsfunksjon vi ønsker å implementere dersom vi får tid er å kunne gi brukeren en oversikt over hvor det er mest/minst UV-stråling i landet. I første omgang tenkte vi at denne hadde vært fint å vise brukeren visuelt i et kart, der landet deles opp etter fylke eller kommune og hver del av landet får en farge basert på den nåværende UV-strålingen i området. Vi er imidlertid veldig usikre på hvor omfattende dette vil være å implementere, så vi har det også som en mulighet å vise en liste over de N stedene med mest/minst UV-stråling i landet.

Et vanlig problem på sommeren er at man glemmer å påføre mer solkrem etter en stund og blir solbrent som følge av dette. Derfor ønsker vi også å kunne sende brukeren en notification/varsel etter X antall minutter når det er tid for en ny runde med solkrem (dette bør antakelig være basert på solkrem/faktor brukt, hudtype og nåværende UV-nivå).

Å brukeren en notification når strålingen er over et nivå X i området brukeren befinner seg er også noe vi *kan* ha med i applikasjonen.

### 3.2 Ikke-funksjonelle krav

(definisjon?)

*Ikke-funksjonelle krav omhandler hvordan applikasjonen vår skal oppnå de funksjonelle kravene. Her vil vi blant annet stille krav til pålitelighet, brukervennlighet og hastighet. Egenskaper som størrelse og flyttbarhet er også elementer innen ikke-funksjonelle krav, men siden appen er såpass liten antar vi at størrelse ikke blir noe problem og siden appen kun utvikles for Android plattformen blir heller ikke flyttbarhet et sentralt element.*

#### 3.2.1 Produktkrav

(brukervennlighet, ytelse, lagringsplass, responstid, antall, samtidige brukere, sikkerhet osv.)

##### Responstid

Kravet vårt til responstid er at brukeren har informasjon om vær og UV-stråling innen 1 sekund etter at appen laster. I forhold til responstid så er det nok API-kallene appen gjør som vil være flaskehalsen, resten av appen forsøker vi å gjøre så “lightweight” som mulig både for å få god responstid og for å koden så ryddig og enkel som mulig for å tillate videreutvikling. Applikasjonen henter data fra MET-APIet sitt Nowcast endpoint. Dataene som hentes fra dette APIet er avgjørende for at appen skal gi brukeren den informasjonen hun ønsker. Dermed blir appens responstid med tanke på å vise frem data antakelig i hovedsak avgjort av hvor raskt dette APIet svarer. I tillegg bør vi selv implementere MVVM arkitekturen på en så god måte som mulig for å hindre at vi selv øker responstiden.

##### Antall samtidige brukere

Når det gjelder maksimalt antall samtidige brukere er det igjen MET-APIet som blir flaskehalsen. Appen vår kjører lokalt på en Android telefon og helt uavhengig av de andre enhetene som bruker appen. Slik vi ser det er det altså bare hvor mange brukere som kan hente ut værdata samtidig fra MET-APIet som er en begrensende faktor. Siden målgruppen vår er studenter (?) i Norge, så vil nok aldri dette kunne bli et reelt problem. MET-APIet ligger bak store tjenester som YR.no som håndterer langt flere brukeren enn vår målgruppe. Det vi imidlertid skal gjøre for å kunne håndtere mange brukere er å cache data, slik at brukere ikke kan overbelaste MET-APIet ved å få appen til å gjøre unødvendige kall til APIet. Dette gjør også at vi reduserer sannsynligheten for at MET-APIet “throttler” brukerne våre.[[4]](#footnote-3)

##### Lagringsplass

Applikasjonen skal ikke ta betydelig stor plass på brukerens telefon. Vi har dette som mål, da de funksjonelle kravene til appen ikke skulle tilsi at appen behøver mye plass. Funksjonaliteten er relativt simpel og applikasjonen er ikke avhengig av store mengder data for å fungere, derfor bør den heller ikke ta opp unødvendig plass. Vi kommer bare til å lagre et fåtall parametere om brukeren og cacher (?) noe værdata (her er det snakk om lagringsplass i bytes). I tillegg har appen lite “tungt” media innhold. På den annen side tenker vi ikke å legge særlig vekt under utviklingen på at appen skal ta lite plass, for uansett om vi ikke gjør alle mulige optimaliseringer med tanke på lagringsplass så vil antakelig ikke dette bli et problem uansett.

##### Sikkerhet/personvern

Det er viktig for oss at applikasjonen ivaretar brukerens personvern. Appens funksjonelle krav tilsier at vi ikke behøver å vite mye om brukeren for å kunne tilby ønsket funksjonalitet. Derfor skal vi heller ikke be om/samle inn mye informasjon om brukeren. Vi trenger følgende to personvern sensitive opplysninger for å oppfylle appens krav:

* Lokasjon
* Hudtype

For å få tak i lokasjon kan brukeren velge å dele sin lokasjon med applikasjonen, men vi tilbyr også en løsning for at brukeren manuelt kan velge lokasjon (via en søkeside). Informasjon om lokasjon skal **kun** brukes ved API-kall til MET-APIet for å få værmelding/UV-varsel for den aktuelle lokasjonen. Lokasjon vil også caches (?) lokalt i appen, men aldri sendes til andre servere.

Mye av det samme som gjelder for lokasjon er også sant for informasjon om brukerens hudtype. Vi tilbyr en tjeneste der brukeren kan ta bilde av armen sin slik at vi kan anslå hudtype. Men vi har også en mulighet det brukeren manuelt kan velge mellom de 6 Fitzpatrick hudtypene. Dersom brukeren velger å benytte bilde-funksjonen vil bildet sendes til et eksternt API ([lukavic1996@gmail.com](mailto:lukavic1996@gmail.com) må si mer om dette). Etter at vi har fått svar fra APIet vil kun hudtypen lagres lokalt i applikasjonen som et tall (1-6), bildet vil ikke lagres.

#### 3.2.2 Organisasjonskrav

(kostnader, ressurser, leveransetidspunkt, prosessmodeller, utviklingsmodeller, programmeringsspråk og rammeverk standarder og regler i bedriften)

* Skrive om lav kobling og høy kohesjon (?) <- står i rapportkrav

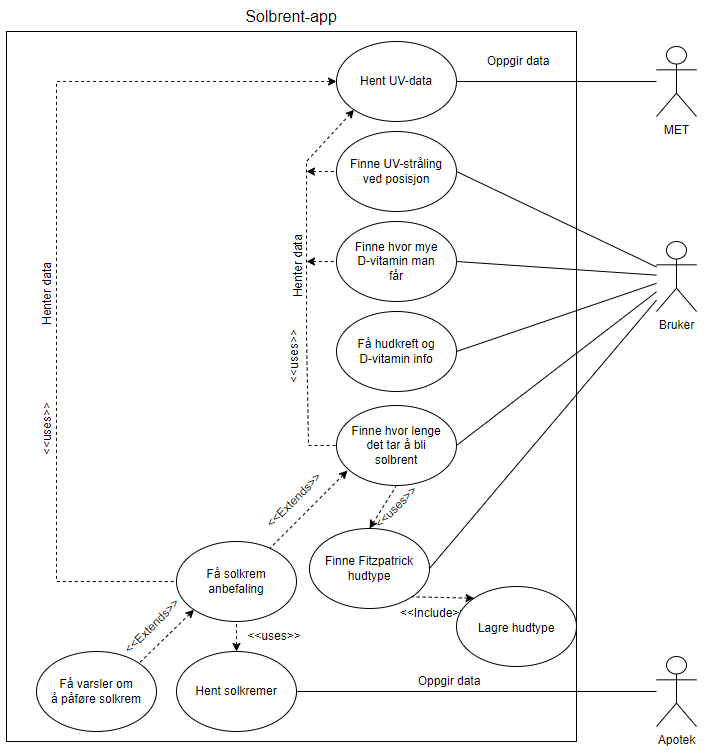
**MVVM.** Vi benytter en MVVM arkitektur der vi deler opp ansvaret for å hente, formatere og vise data i appen. Vi skal ha minst et DataSource objekt som har ansvar for DataSource, Model, ViewModel…

**Utviklingsmodeller.**

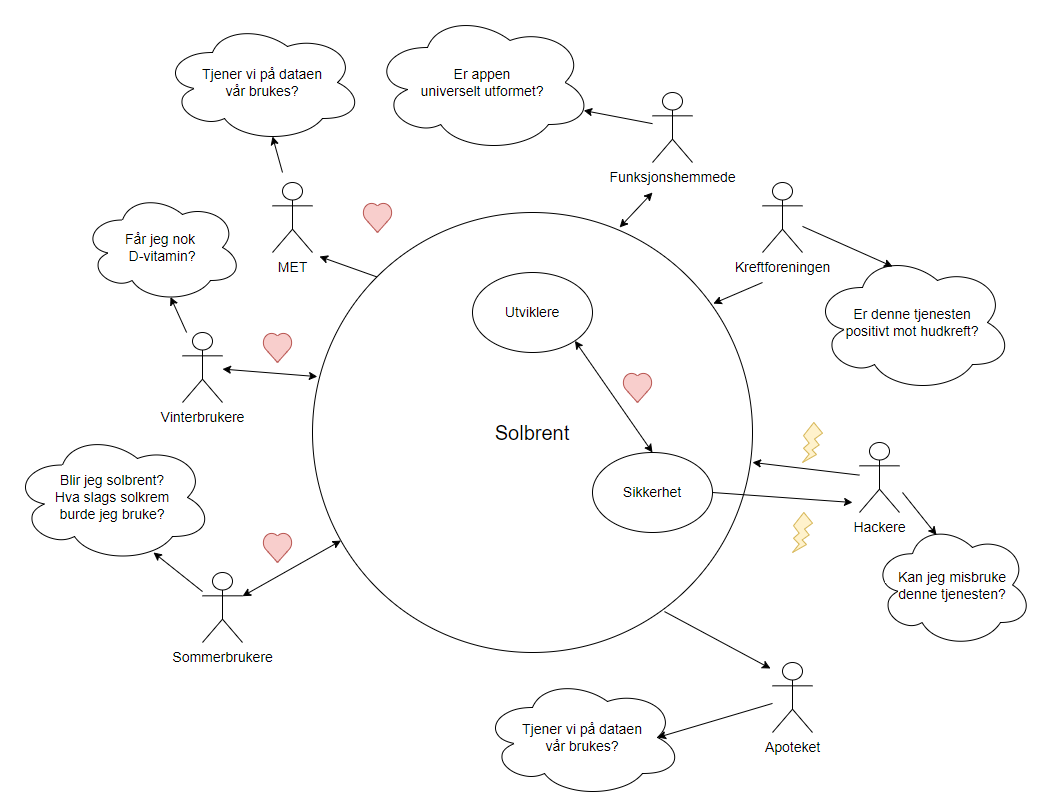
**Programmeringsspråk.** Kotlin, XML (kanskje noe Java)....

**Leveransetidspunkt.** Vi skal ha implementert alle SKAL-krav innen 20(??) mai 2022.

### 3.2 Use case (og user-stories?)



(et utkast til et use-case ihvertfall. kommentarer til forbedring er ønsket)



(legger til forsøk på rikt bilde her også, vet ikke hvor den plasseres)

User-stories

- Som bruker ønsker jeg anbefalinger for solkrem

- Som bruker ønsker jeg å vite hvor mye Sol huden min tåler før jeg blir solbrent

- Som bruker ønsker jeg å få vite når jeg bør ta på solkrem igjen

- Som bruker ønsker jeg å finne min fitzpatrick-hudtype

- Som bruker ønsker jeg info om hudkreft og D-vitamin

- Som bruker ønsker jeg å se hvor mye D-vitamin jeg får

- Som bruker ønsker jeg at appen er lett å bruke

- Som funksjonshemmet ønsker jeg at appen er universelt utformet

- Som bruker ønsker jeg info om været

- Som bruker ønsker jeg å vite hvor sterk UV-stråling det er ved min posisjon

- Som bruker ønsker jeg at appen husker min hudtype

- Som utvikler ønsker jeg å få UV-data fra MET

- Som utvikler ønsker jeg å få oppgitt brukers Fitzpatrick-hudtype

- Som utvikler ønsker jeg å få info om solkremer fra apoteket

### 3.3 Klassediagram

### 3.4 Sekvensdiagram

### 3.5 Design pattern

## 4. Produktdokumentasjon

### 4.1 Beskrivelse av løsning

*\*Beskriv løsningen beregnet på lesere som skal jobbe med drift, vedlikehold og videreutvikling av løsningen. Beskriv hvilke teknologier og arkitektur som brukes i løsningen (både front-end og back-end).\**

### 4.2. Liste over app egenskaper

*\*Beskriv de viktigste kvalitetsegenskapene ved appen (for eksempel funksjonalitet, brukskvalitet, pålitelighet (fravær av feil) og hvordan disse er evaluert (jfr. forelesning 07.03).\**

### 4.3 API-nivå

Det ble tidlig besluttet at vi for denne applikasjonen skulle anvende API-nivå 30 (R). Vi valgte dette API-nivået på bakgrunn av at enkelte benyttet MacBook med M1-chip, hvor API-nivå 30 er det laveste som kan bli anvendt.

Til fordel gir det å anvende et høyere API-nivå muligheten til å implementere flere funksjonaliteter, og generell frihet til å kunne utvikle nye konsepter. Til ulempe er den ikke like tilgjengelig for alle enheter, da hovedsakelig brukere som benytter eldre enheter naturligvis også går på eldre API.

(<https://www.androidpolice.com/googles-latest-android-version-distribution-numbers-show-11-in-dead-heat-with-10/>)

## 5. Testdokumentasjon

*• Beskriv planlegging og gjennomføring av testing av løsningen underveis i prosjektet,  
Beskrivelsen skal inneholde hva målet med testingen var, og hvilke testverktøy som er brukt.\**

5.1 Formål med testing

Testing er en elementær del av den helhetlige prosessen innen apputvikling og burde iverksettes så tidlig så mulig, men ikke minst gjentas kontinuerlig under hele prosessen. Dette er en viktig del av utviklingen for å kunne forhindre feil som er uoppdaget, bugs( i guess det er feil på norsk?), samt redusere risikoen for problemer som oppstår under eksekvering.

Vi skiller mellom testing som blir (skrive om at det er testing av bugs/kode/feil i systemet, men også testing av det grafiske (brukertesting), wcag krav, er appen forståelig og brukervennlig)

Det er ikke mulig å teste alt, men ved å innføre

### 5.2 Testverktøy

-> android studio, JUnit?

### 5.3 Planlegging

### 5.4 Dynamisk / Statisk testing (?)

#### 5.4.1 Enhetstesting

## 6. Prosessdokumentasjon

### 6.1 Sprint 1

#### 6.1.1 Planlegging og struktur

##### 6.1.1.1 Snakke om første gruppemøte

##### 6.1.1.2 Valg av case

#### 6.1.2 Verktøy

##### 6.1.2.1 GitHub

##### 6.1.2.2 Slack / Facebook

#### 6.1.3 Valg av smidig utviklingsmetode

#### 6.1.4 Rollefordeling

#### 6.1.5 Brukerundersøkelser

### 6.2 Sprint 2

#### 6.2.1 Kravspesifikasjon

#### 6.2.2 Datainnsamling

##### 6.2.2.1 Brukerundersøkelse

##### 6.2.2.2 Intervju

#### 6.2.3 Dataanalyse

### 6.3 Sprint 3

#### 6.3.1 Brukbarhetstesting

### 6.4 Sprint 4

### 6.5 Sprint 5

#### 6.5.1 Implementering av funksjonalitet

### 6.6 Sprint 6

#### 6.6.1 Brukerevalusering av endelig applikasjon

## 7. Refleksjon

### 7.1 Utfordringer

### 7.2 Lærdommeer

## 8. Referanser

## 9. Vedlegg (figuroversikt, intervju, sprint-loggføring, referater?)

1. <https://www.wcrf.org/dietandcancer/skin-cancer-statistics/> 10.03 [↑](#footnote-ref-0)
2. <https://gs.statcounter.com/os-market-share/mobile/norway> [↑](#footnote-ref-1)
3. <https://www.europarl.europa.eu/RegData/etudes/BRIE/2018/621894/EPRS_BRI(2018)621894_EN.pdf> 16.03 [↑](#footnote-ref-2)
4. https://api.met.no/doc/TermsOfService [↑](#footnote-ref-3)