

TEKNISK BRUKERMANUAL

For å kjøre programmet vårt, gjøres dette lettest ved hjelp av Eclipse med ADT-plugin og SDK-verktøy, ADT Bundle anbefales. Da kan man laste applikasjonen over til et nettbrett eller en telefon som støtter Android, eller man kan kjøre det i simulator. Ettersom vi har valgt å konsentrere oss om å utvikle systemet med fokus på én nettbrettmodell(Samsung Galaxy Tab 2), ser vi helst at testing foregår på denne. Ettersom programmet ikke er tilgjengelig på Google Play må det gjøres på denne måten.

For å laste inn koden i Eclipse, gjør følgende:

1)

Gå til
file > import > android > existing android code into workspace

Velg mappen kalt "Find_my_sheep" fra zip-filen, og trykk så "finish".

Programmet skal nå være lastet i Eclipse. For å kjøre programmet må man koble en Android-enhet til datamaskinen og skru på USB-feilsøking i enhetens innstillinger. Man kan også opprette en emulator i Eclipse med Android Virtual Device Manager under *window*. Emulatoren burde ha en størrelse på 1204 x 600. Etter å ha satt opp en enhet er det bare å trykke på run. Ønsker man å få opp Andorids debug output-vindu gjør man følgende:

2)

Gå til
window > show view > other > android > log cat

For å kunne se kartet når man kjører programmet må man laste ned Google Play Services fra Android SDK manager(under *window*-menyen). Etter at denne er lastet ned må man:

3)

Gå til
file > import > android >existing android code into workspace

Importer mappen:

SDK>extras > google > google_play_services > libproject > google-play -services_lib

Når denne er importert, høyreklikker man på prosjektmappa

4)

Høyreklikk på prosjektmappen, og velg
Properties(eller preferences) > android >add > google-play-services_lib

Dette er kun nødvendig dersom denne ikke allerede er lagt til. Til slutt må man legge til debug.keystore for å få riktig Google Maps API Key.

Dette gjøres slik:

- 5)

Gå til
window > preferences > android > build

Velg *browse* på *custom debug keystore* og legg til filen:
"debug.keystore copy"- som ligger vedlagt i zip-filen.

MANUAL FOR OPPSETT AV SERVER

Denne manualen vil kun se på hvordan man setter opp serveren på Linux, men det skal i teorien være mulig å gjøre det på andre operativsystemer.

For å sette opp serveren til applikasjonen trenger man en server med rottillgang. Hvis man har installert PostgreSQL og Apache, må Apache settes opp slik at at CGI-skript mappen er i:

/var/www/cgi-bin/

Videre må Python 3.0 installeres. All koden som skal kjøres ligger vedlagt. Kopier cgi-bin mappen fra vedlegget til

/var/www/

Du skal nå ha filer med denne strukturen:

/var/www/cgi-bin/adduser.py

Når dette er gjort må databasen settes opp. Det første du må gjøre er å lage en database med navnet "findmysheep" i PostgreSQL og gi brukeren postgres-tilgang på denne med passordet '*kohxotoo*'.

Vedlagt ligger også en sql-fil. Denne inneholder all informasjonen om strukturen på databasen. Denne kan du mate til databasen og alle tabellene vil bli opprettet på rett måte. Du må også rute alle data fra port 80 (standard for HTTP) til serveren din fra ruteren om du ikke allerede har gjort dette.



«Sheepolini» Brukermanual



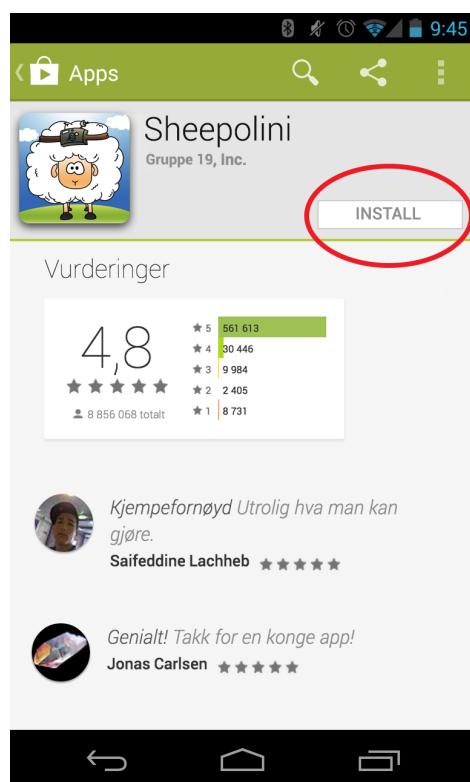
Innhold

1. Installasjon	2
2. Muligheter	3
3. Kom i gang	4
4. Eksisterende brukere	5
5. Førstegangs brukere	10
6. Andre valg	15
7. Kart	18
8. Varsling	19
9. Hjelp	20

1. Installasjon

Søk opp **Sheepolini** i **GooglePlayStore** og trykk [Installer]

The screenshot shows the Google Play Store interface. In the search bar at the top, 'Sheepolini' is typed. Below the search bar, the app's page for 'Sheepolini' is displayed. The app icon features a cartoon sheep wearing a camera. The title 'Sheepolini' is followed by the developer information 'Gruppe 19, Inc.' and the release date '7. november 2013'. A green 'Installer' button is highlighted with a red circle. Below the button, there are three screenshots of the app's interface: a splash screen with the title 'SHEEPOLINI', a login screen with fields for 'NAVN' and 'PASSORD', and a main menu with options like 'NAME', 'ALDER', 'VØRT', 'HØST', 'STASJON', 'INDIKER', 'POSITION', 'VELDOS', and 'SLÅ AV/PÅ'. At the bottom of the page, there is a section titled 'Beskrivelse' with a long placeholder text block, and another section titled 'Vurderinger' with several user reviews and their ratings.

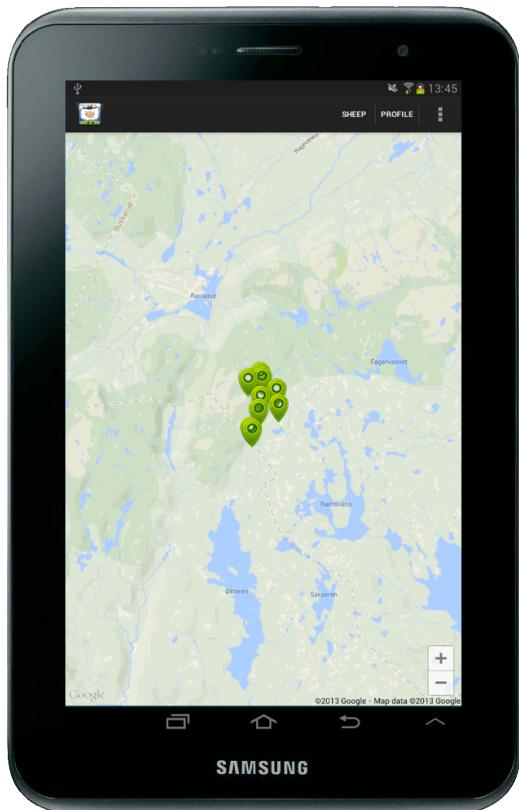


2. Muligheter

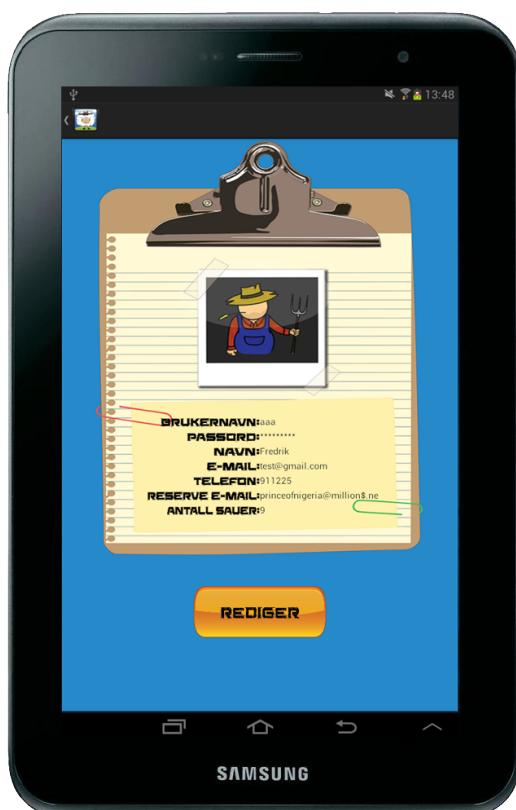
- Registrer bruker
- Registrer sauер
- Administrer sauene dine
- Logg hendelser
- Se sauene dine på et kart
- Få varsling

3. Kom i gang

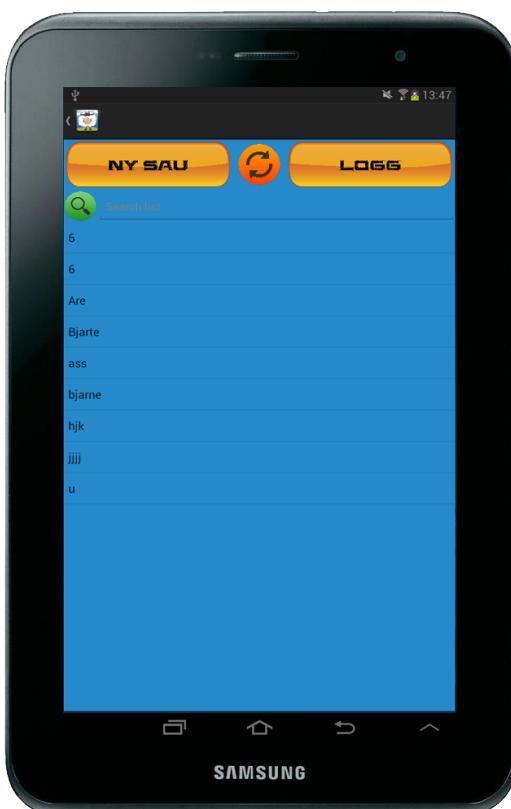
Applikasjonen har 4 hoved skjermer.



Kart



Brukerprofil



Liste over sauер



Logg

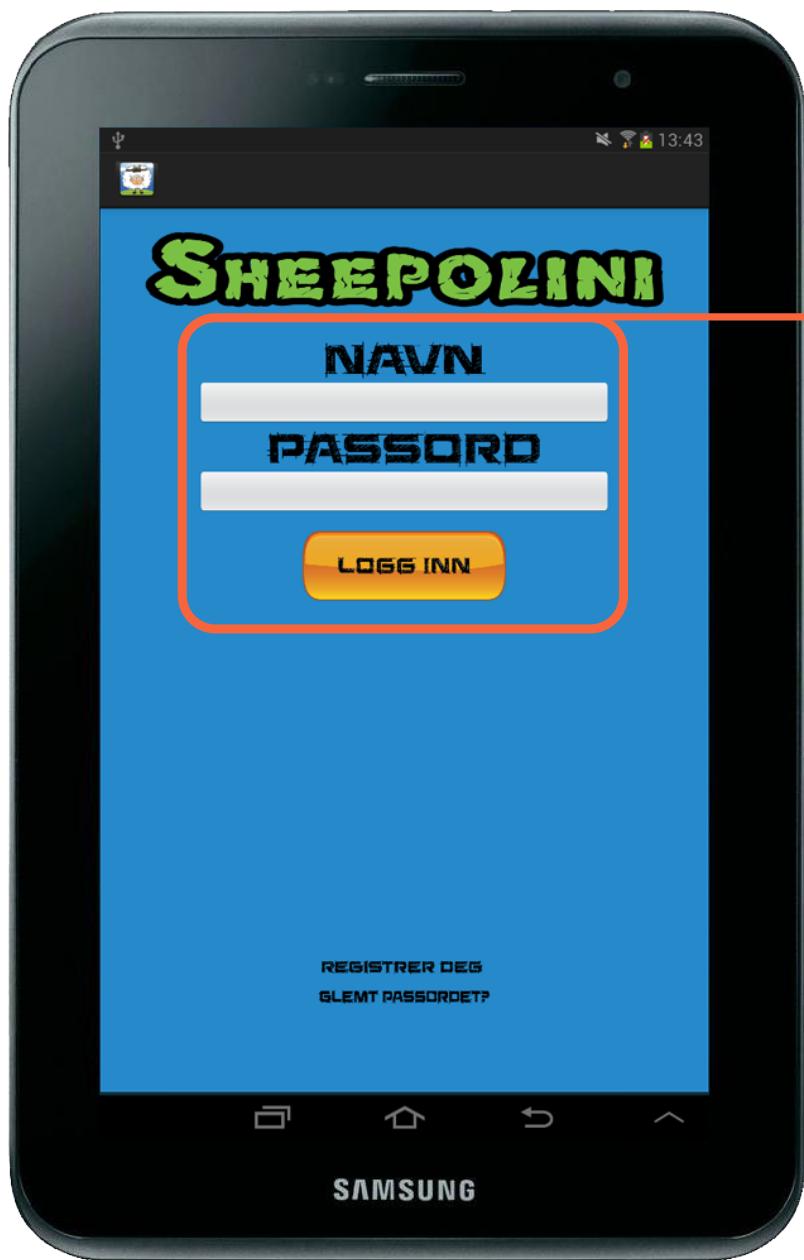
4. Eksisterende brukere

Hvis du har en registrert bruker i Sheepolini,
har du disse mulighetene.

Ønsker du å opprette en bruker gå til side 10

Eksisterende brukere - Logg-inn

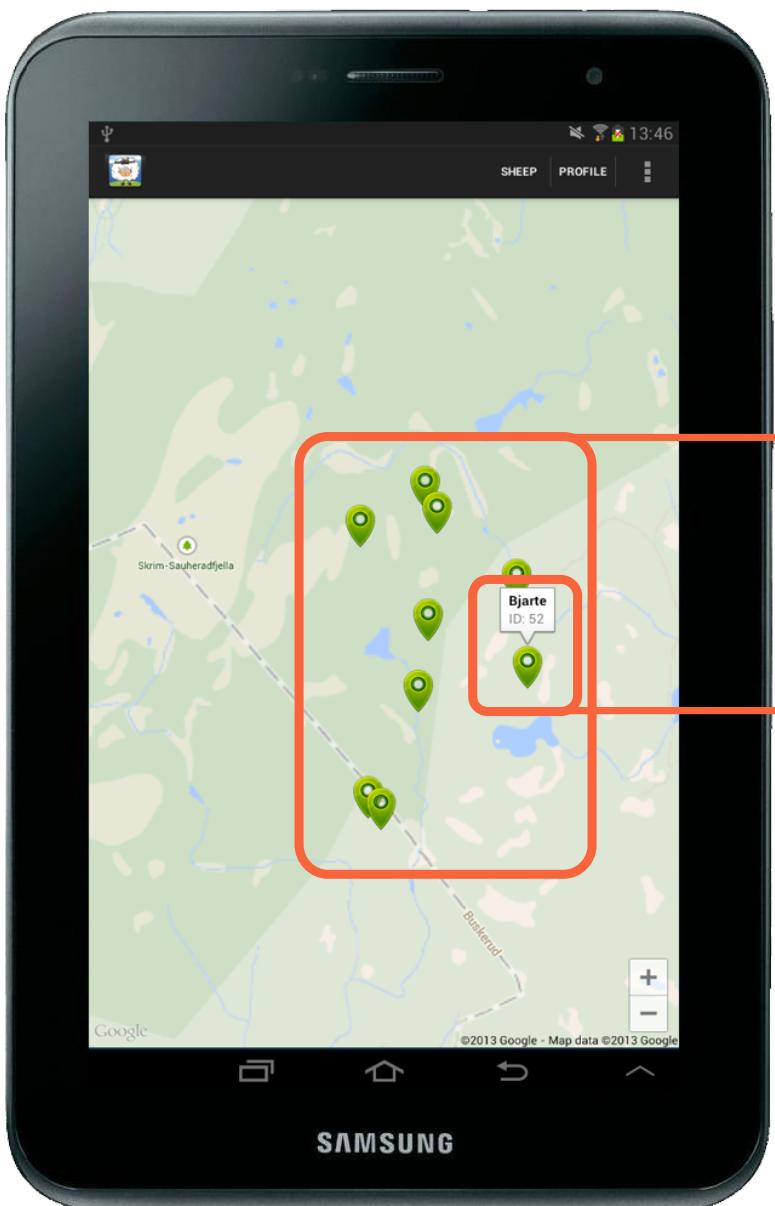
Først må man logge inn med brukernavn og passord.



Skriv inn **brukernavn** og
passord og trykk **[Logg inn]**

Eksisterende brukere - Kart

Kartet viser hvor dine registrerte sauene befinner seg.



De grønne markører viser registrerte sauene brukeren har lagt til

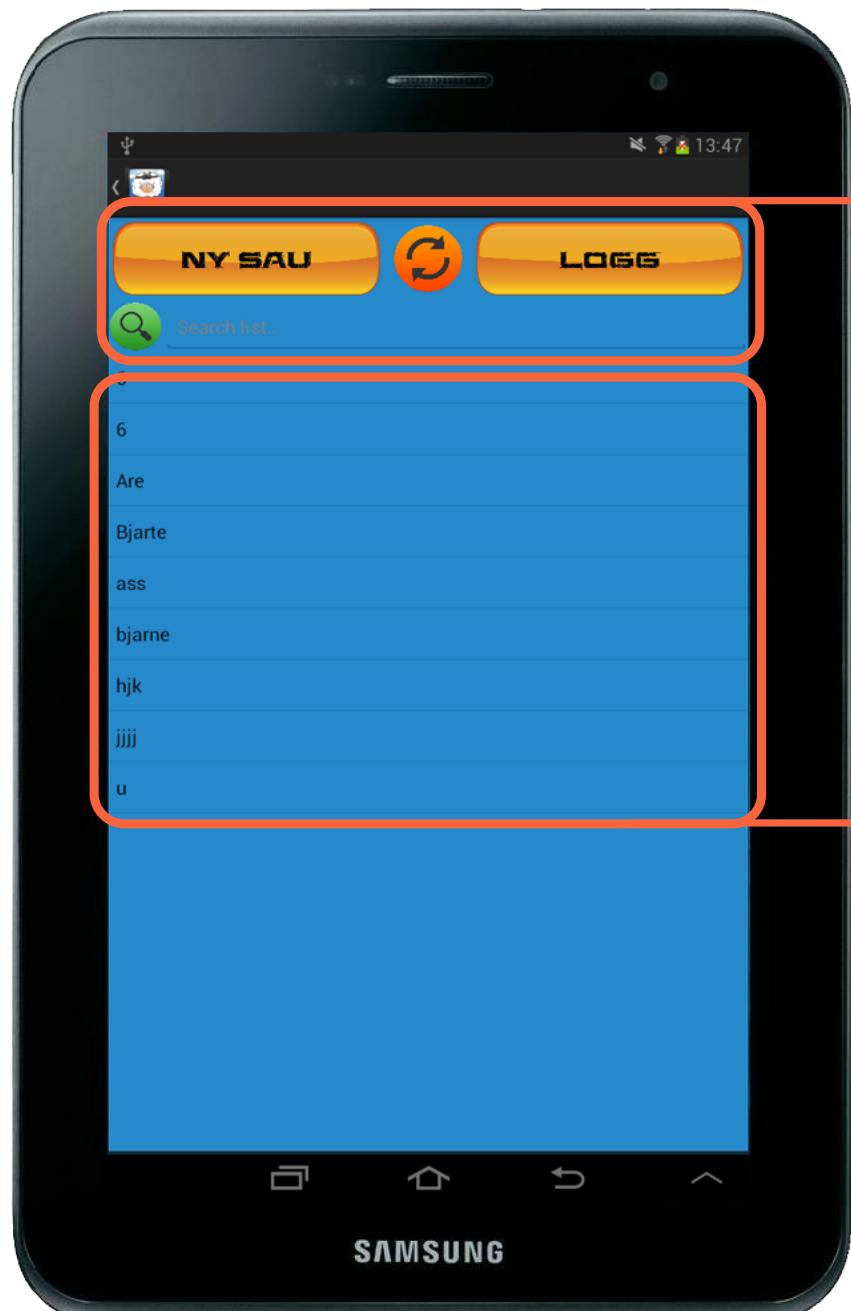
Ved å trykke på en markør poppes det opp et lite informasjons vindu med **[Navn]** og **ID** til sauens

Trykk på informasjons vindu for å se sauaprofil

Eksisterende brukere

Liste over sauene

Her ser man liste over alle registrerte sauene

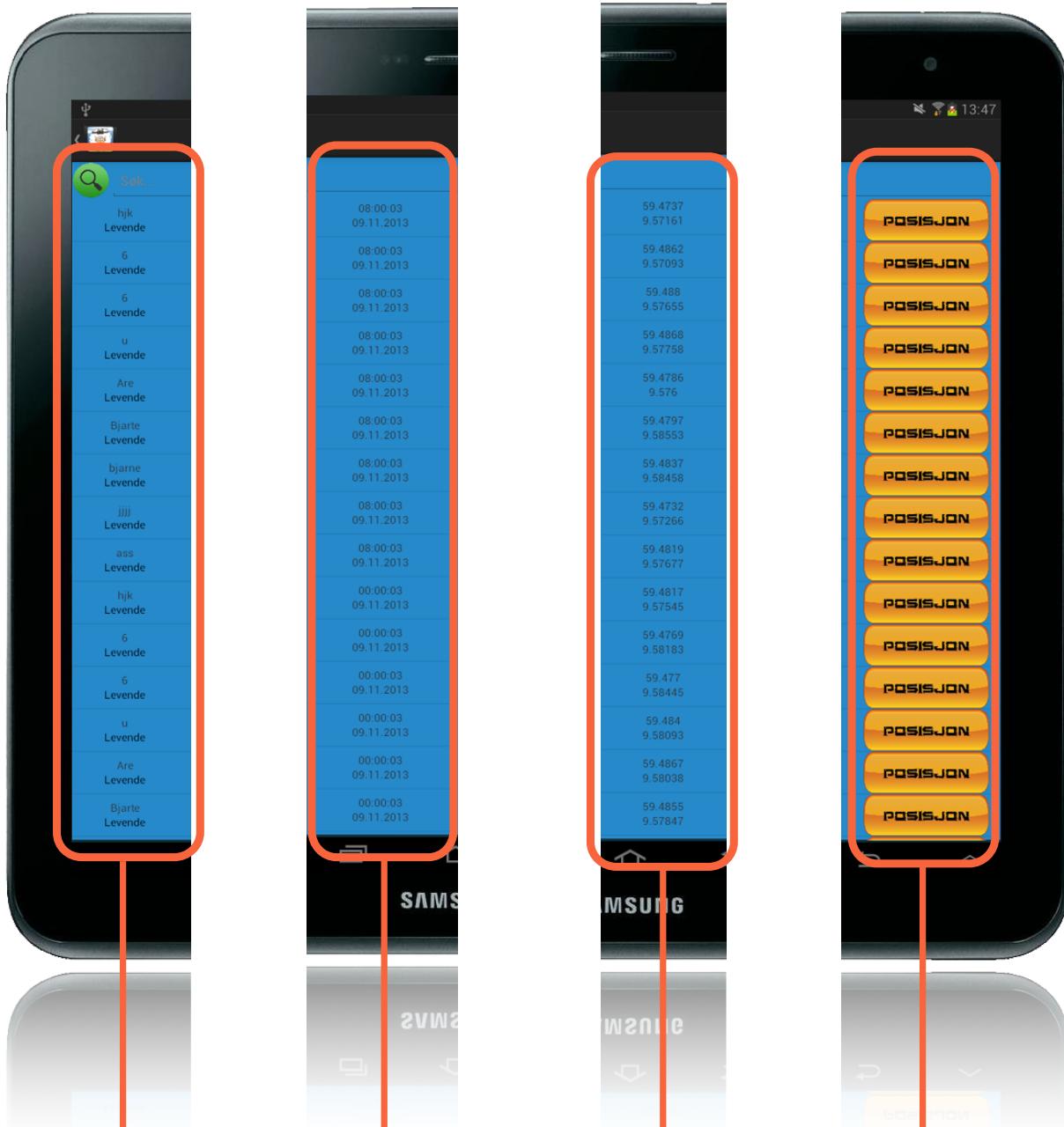


Knapper med muligheter
for å legge til [Ny Sau], se
[Logg], [Søke] etter navn
og [Oppdatere] listen

Fullstendig liste over
registrerte sauene

Eksisterende brukere - Logg

I loggen kan man se posisjon og status for alle sauene.
Status oppdateres 3 ganger i døgnet.



Oversikt over
navn til sauene
og status

Tid da
oppdateringen
ble gjort

Sauens
koordinater

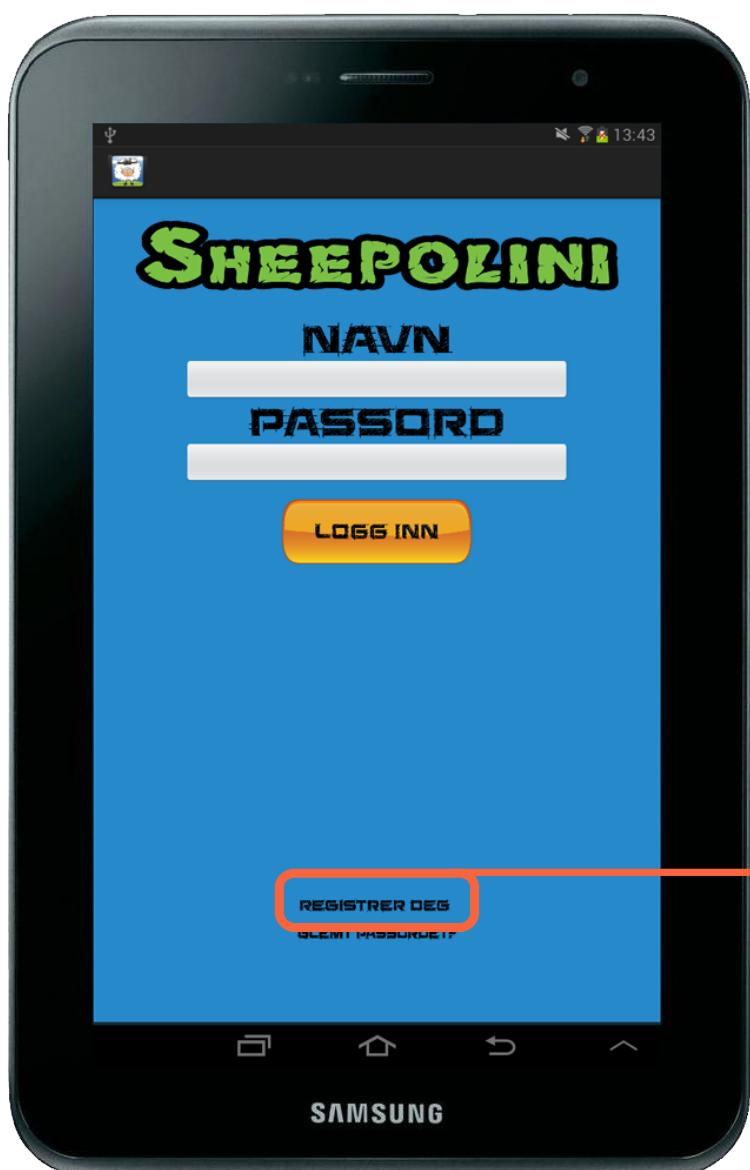
Ved trykk på
[Posisjon] får man
opp sauens
posisjon under
oppdaterings tid

5. Førstegangsbrukere

For brukere som ikke har konto eller ønsker å opprette en ny konto.

Førstegangsbrukere Registrering (1 av 2)

Første gang man bruker Sheeplini må man opprette en bruker.

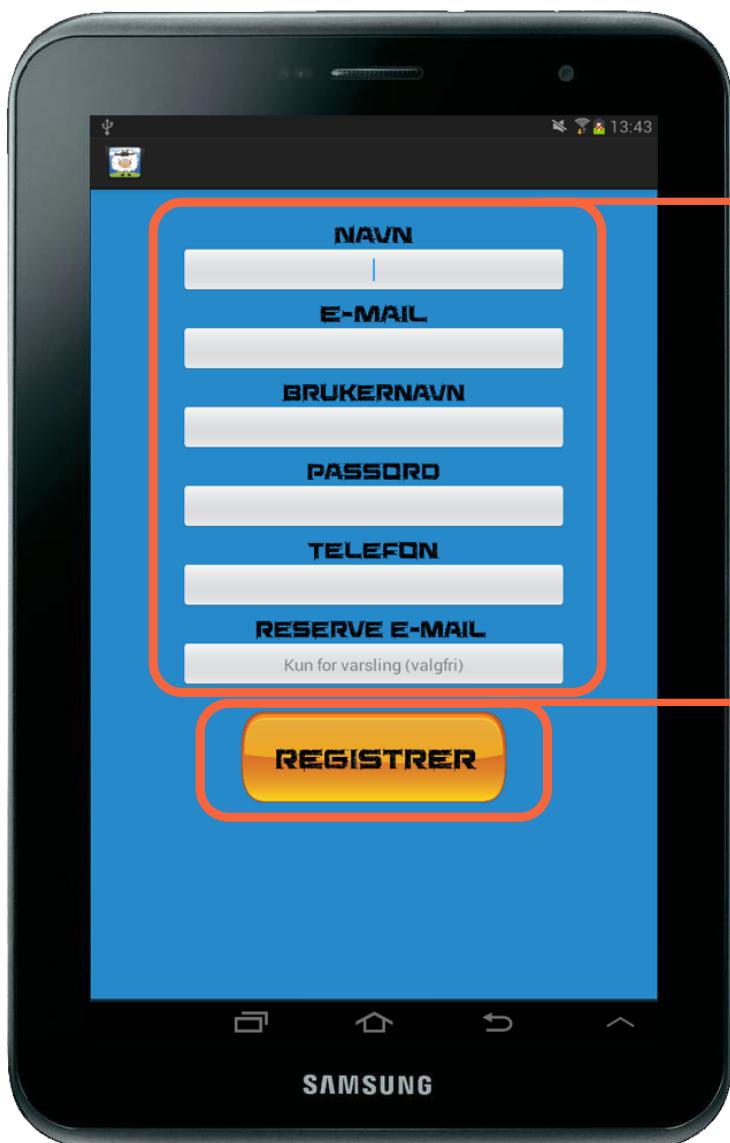


Trykk **[Registrer deg]** dersom du ønsker å lage ny bruker.

Trykk **[Glemt Passordet]** dersom du ønsker å gjenopprette passordet

Førstegangsbrukere Registrering (2 av 2)

Her skriver vi inn den informasjonen vi ønsker å registrere oss med



Skriv inn **[Navn]**, **[E-mail]** og **[Telefon]**

Velg **[Brukernavn]** og **[Passord]**

[Reserve E-mail] fylles ut valgfritt. Dersom den blir tomt skal varsling-mail sendes til **[E-mail]**

Når alle feltene er utfylt, trykk **[Registrer]** for registrere deg og gå videre



Førstegangsbrukere Liste over sau

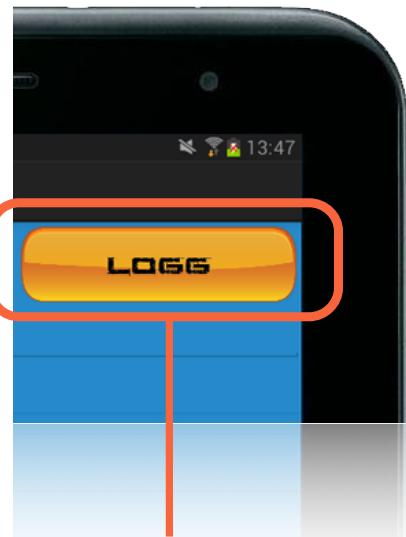
Her får brukeren full oversikt over alle sauene som ble registrert



Trykk **[Ny Sau]** for å legge til ny sau



[Oppdater] knappen, oppdaterer status og lokasjon til sau

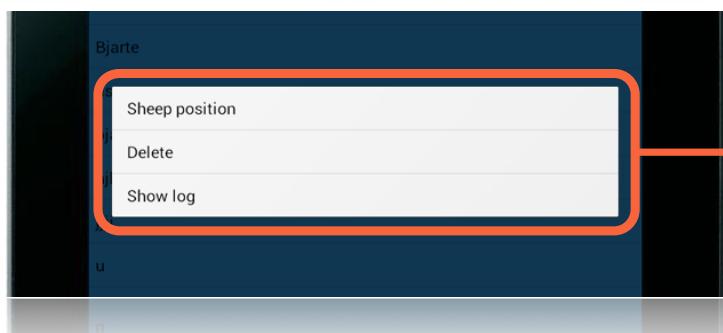


Trykk **[Logg]** for å se full logg over alle sauene



[Søkevindu]

Skriv inn navn på sau du vil se på

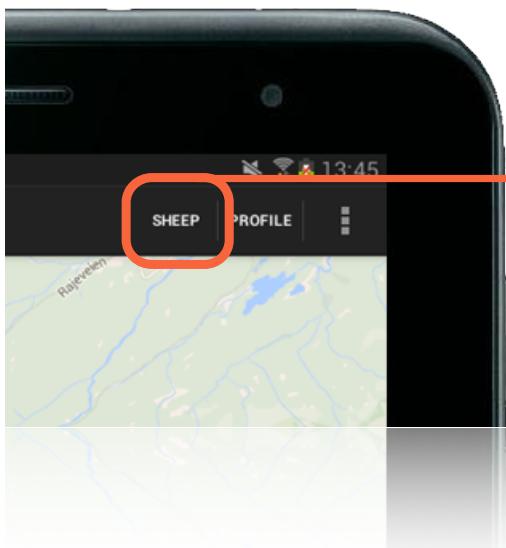


Ved å holde fingeren på en sau popper opp meny med snarveier til funksjonene

Førstegangsbrukere

Registrering av sau

Når brukeren er logget på med ny konto må vi legge til sauer.



Ved å trykke på [**Sheep**] får man opp liste over alle sauene som er registrert



Trykk [**Ny Sau**] for å registrere ny sau



Her fyller brukeren [**Navn**], [**Alder**], [**Vekt**] og [**Helse**] i hver sin felt

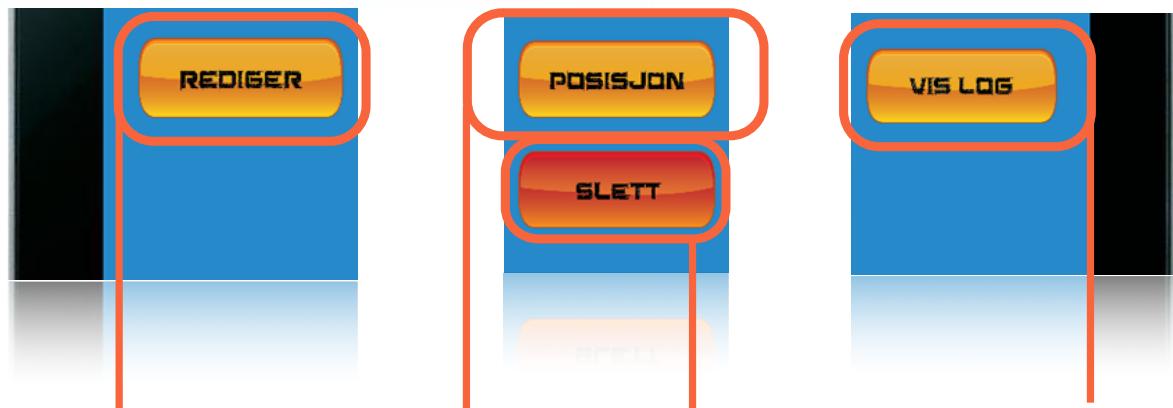
Når alle feltene er utfylt, trykk [**Legg Til**]

Andre valg Saueprofil

Her kan brukeren se på informasjon om en enkelt sau.



Her får brukeren full oversikt over sauens informasjon



Dersom du ønsker å endre informasjon, trykk **[Rediger]**

Viser sauens posisjonen

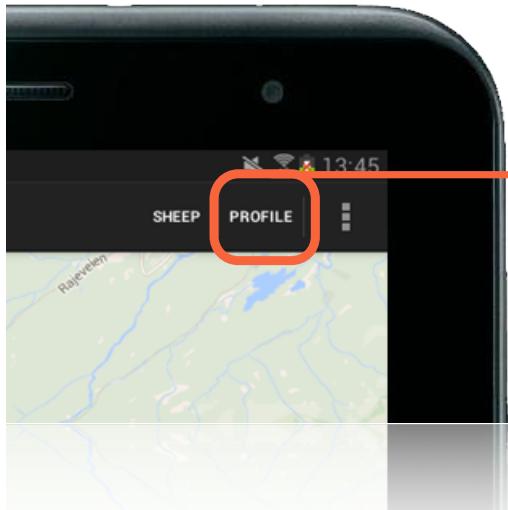
Ved å trykke på **[Slett]** knappen slettes sauven fra lisen

Trykk **[Vis Logg]** for å se sauens logg

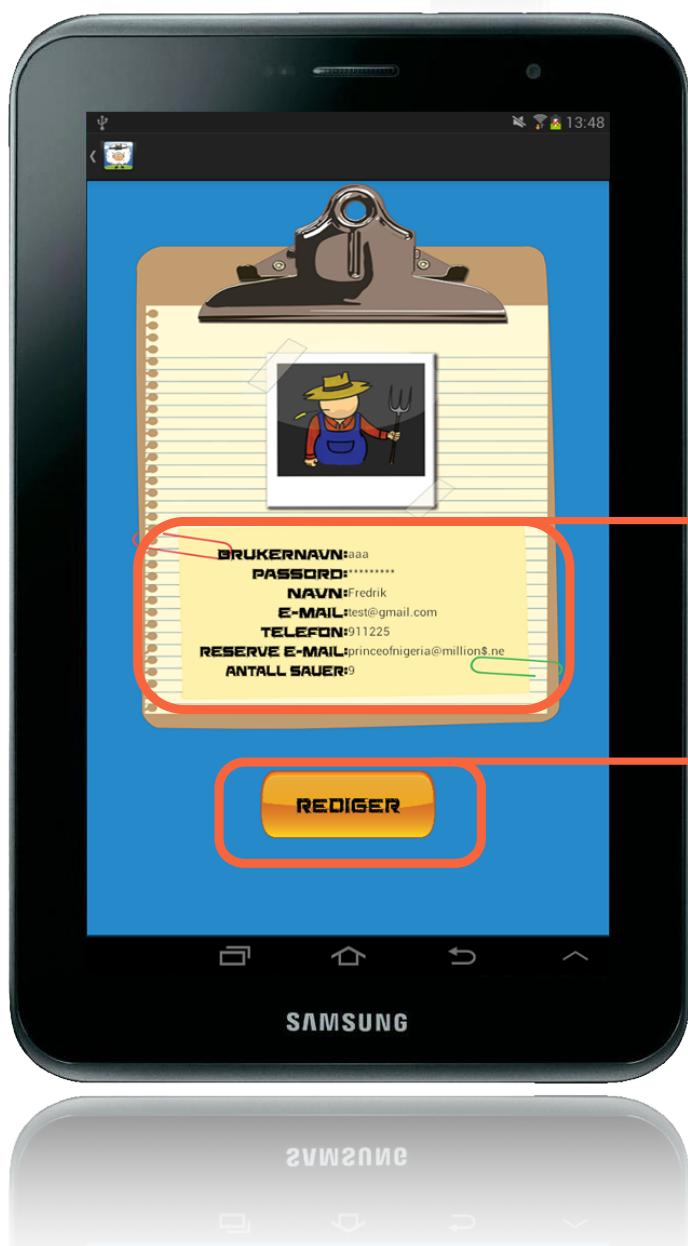
Andre valg

Brukertilgang

Her kan brukeren se på sine registrerte opplysninger



Ved å trykke på **[Profile]**
får man opp
brukerinformasjonen



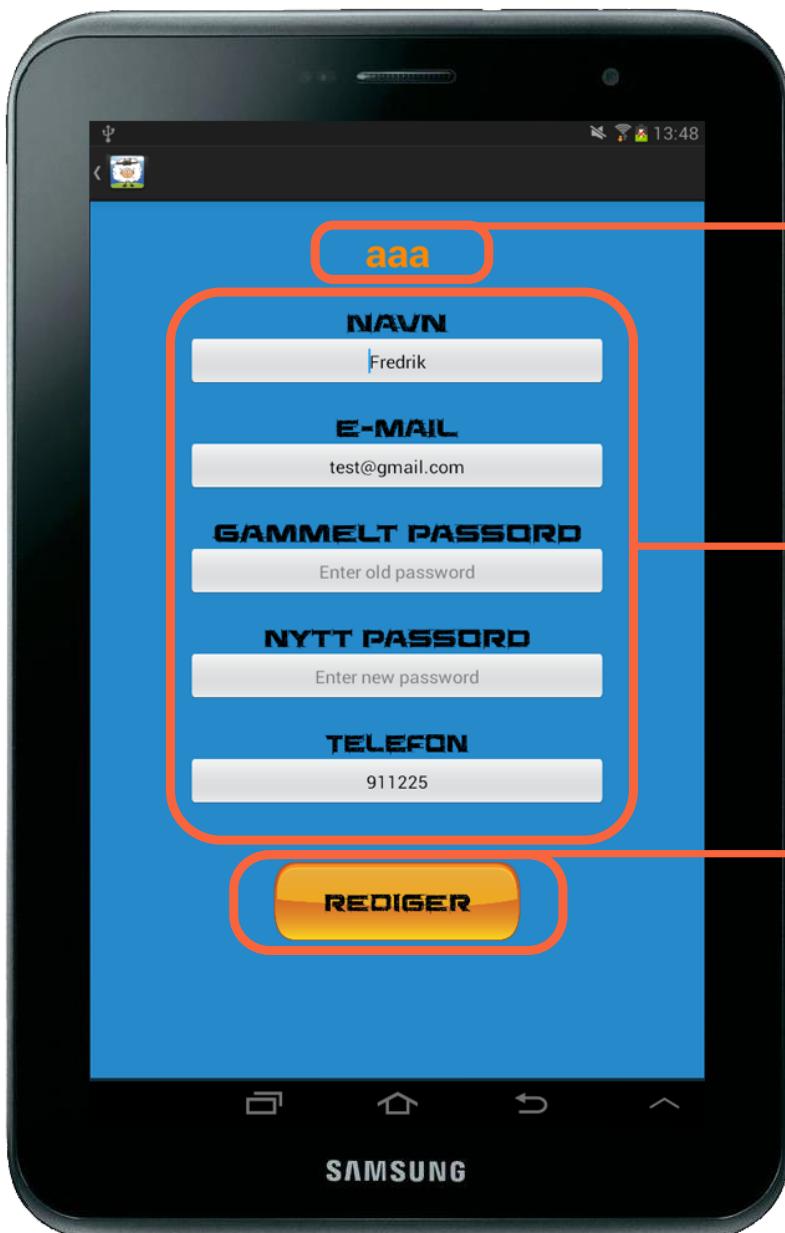
Full oversikt over bruker
informasjonen

Dersom et er ønske å
redigere
brukerinformasjonen,
trykk **[Rediger]**

Andre valg

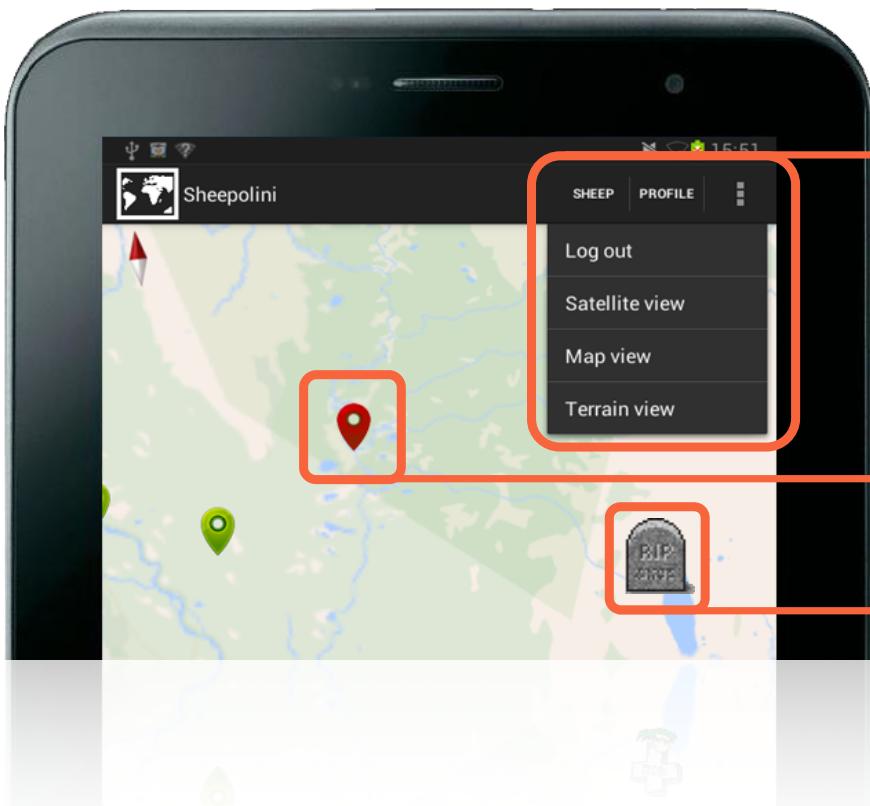
Rediger Brukerprofil

Her kan brukeren redigere sine opplysninger



Kart

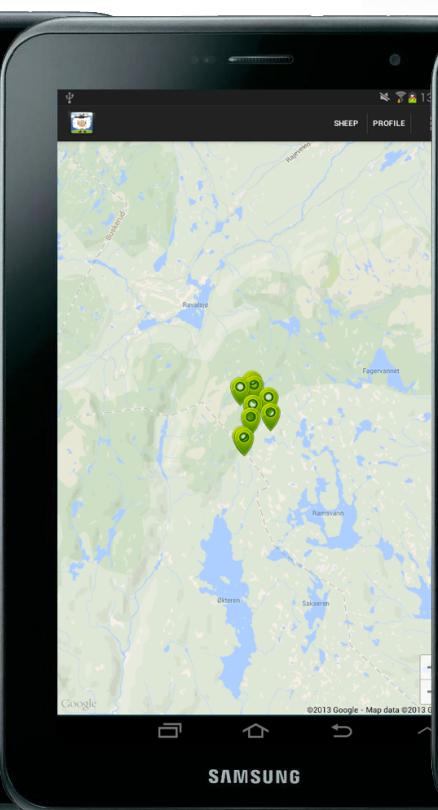
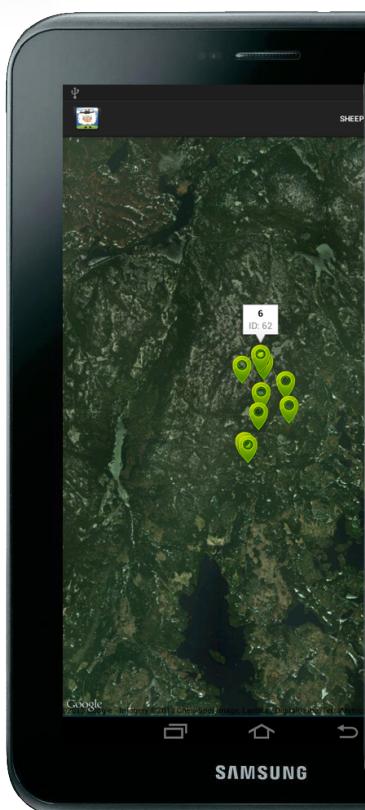
Her kan brukeren se sauene sine



Menu med valg av
kartvisning og **[Log Out]**
knapp for å logge ut

Rød markør som viser at
alarmen har gått

Gravstein som
viser dø sau



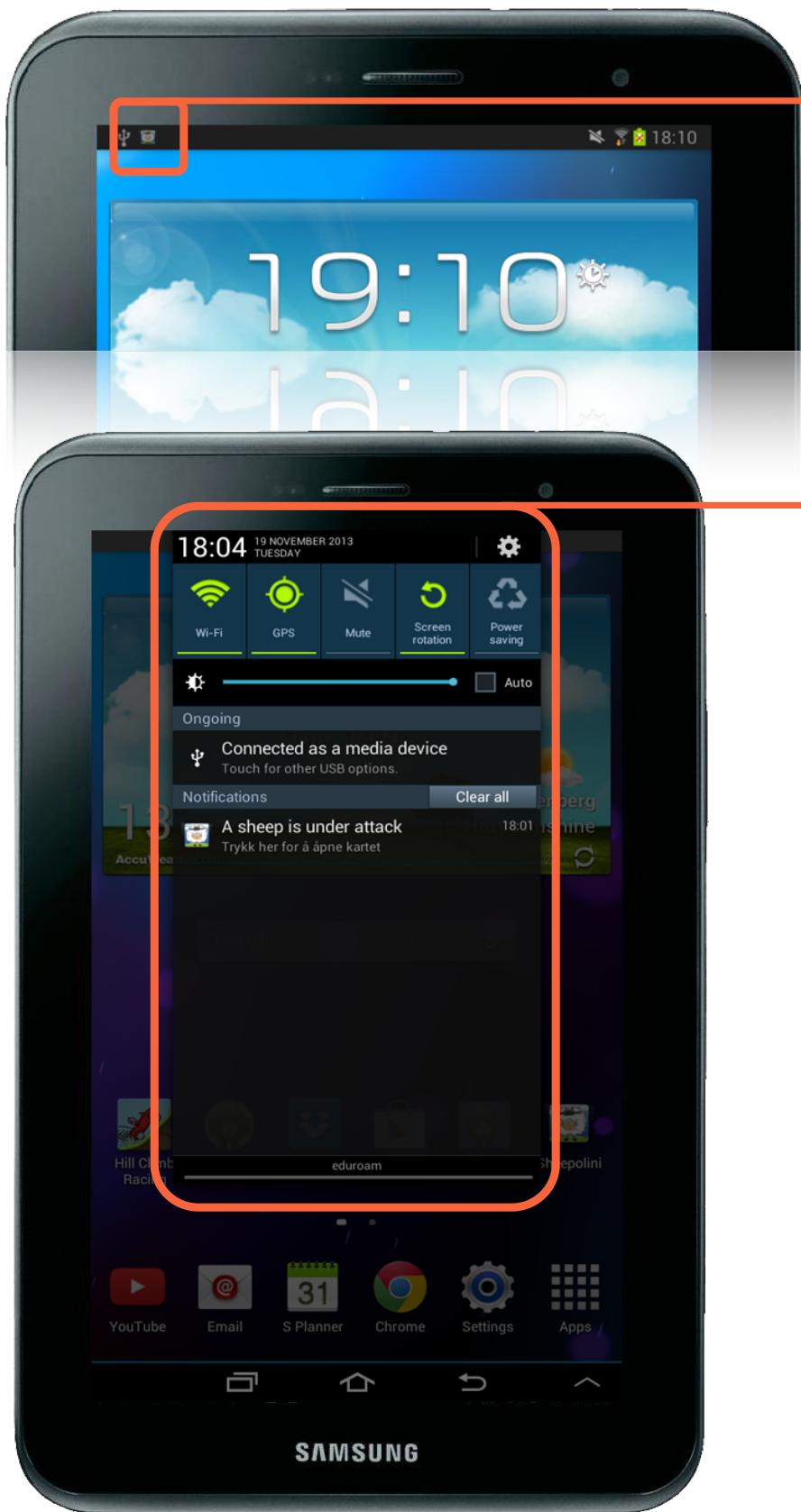
Satellitt

Kart

Terreng

Varsling

Brukeren får varsel når sauens blir angrepet



Ved alarmen popper opp applikasjons ikonet i actionbar på enheten

Ved å rulle ned actionbar på enheten får brukeren oversikt over alarmer og varslinger

Hjelp

Trenger du hjelp

Send mail til findmyherd@gmail.com

Sprint Backlogg 1 - 5

Innhold

Sprint backlog sprint 1.....	1
Sprint retrospective.....	1
Sprint backlog Sprint 2.....	2
Sprint retrospective.....	2
Sprint backlog Sprint 3.....	3
Sprint retrospective.....	3
Sprint backlog Sprint 4.....	4
Sprint retrospective.....	4
Sprint backlog Sprint 5.....	5
Sprint retrospective.....	5

Sprint backlog sprint 1

Torsdag 05.09.13 til torsdag 19.09.13

Som en bonde ønsker jeg å kunne logge inn på en profil med brukernavn og passord

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer
Sette opp database	Edgar	100 %	20 t	19 t
Use-case diagram	HM	100 %	1 t	1 t
Tilstands- eller aktivitetsdiagram	HM	90 %	2 t	5 t
UI design/Lage ikon	Said	100 %	2 t	5 t
Kode loggin «klassen»	Fredrik	100 %	10 t	12 t
Lage klassediagram	Alle	60 %	3 t	3 t
Lage risikotabell	Alle	100 %	1 t	2 t
Møte med kunden	Alle	100 %	1 t	30 min
Testing	Fredrik	100 %	5 t	4 t
ER-diagram	Edgar	100 %	1 t	20 min
Totalt antall jobbtimer			41	Ca 52 t

Sprint retrospective

Bra: Gruppemedlemmene som har vært til stede har jobba bra sammen, og vi har fått gjort mye.

Bra at vi traff målet. Fått litt oversikt over hvordan systemet skal være.

Dårlig: At ikke alle møter opp og at folk kommer for sent.

Bedre neste gang: Forbedre oppmøte.

Sprint backlog Sprint 2

Torsdag 19.09.13 til torsdag 3.10.13

Som en bonde ønsker jeg å kunne registrere opplysninger om mine sauene med id(navn, alder), vekt, helseinformasjon og geografisk lokasjon om en enkelt sau

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer	Kommentar
Kode brukerregistrering	Fredrik	100 %	8 t	6 t	Aktivitet som tar seg av brukerregistrering
Sauedatabase	Edgar	100 %	2 t	5 t	Tabell i databasen for sauene
Kode registrering av sauene	HM	70 %	10 t	7 t	Aktivitet som tar seg for saueregistrering
Møte med kunden	Alle	100 %	1 t	30 min	
Kode testklasse	Said, Edgar	39 %	15 t	22 t	Klasse som tester applikasjonens funksjonaliteter
Hovedrapport	Jim	25 %	5 t	4 t	Skrive introduksjon
Endre WBS	Said, HM	100 %	30 min	30 min	
Endre klassediagram	Said, HM	40 %	1 t	30 min	
Kartforbedring	Fredrik	100 %	10 t	11 t	
Lagret login	Fredrik	100 %	5 t	5 t	
Ikon toppmeny	Hanne Marie	100 %	5 t	7 t	
Totalt antall jobbtimer			62.5 t	69.5 t	

Sprint retrospective

Bra: Mye ble fort ferdig, vi har fått gjort mye, folk har vært flinke til å møte opp

Dårlig: Ikke flinke nok til å bruke trello, mange som kommer for sent

Bedre neste gang: Komme tidsnok, hjelpe de som trenger det for å komme i mål, bli flinkere til å pushe til git, bli bedre til å bruke trello

Sprint backlog Sprint 3

Torsdag 04.10.13 til torsdag 17.10.13

Som en bonde ønsker jeg å kunne slå opp i systemet og redigere informasjon om en sau og om meg selv

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer	Kommentar
Kode «redigere bruker»	Fredrik	100 %	3 t	3 t	Lage aktivitet til å redigere brukerprofil
Kode «vis brukerprofil»	Fredrik	100 %	2 t	2 t	Lage aktivitet som viser informasjon om bruker
Kode «redigere sau»	Hanne Marie	100 %	7 t	10 t	Vise liste med sauer og lage aktivitet til å redigere en sau
Validere inputfelt	Said	80 %	3 t	5 t	Skrive kode så programmet sjekker om bruker har skrevet inn riktig type i inputfelt Flyttes til neste sprint
Test til eksisterende funksjoner	Edgar	100 %	5 t	5 t	
Møte med kunde	Alle	100 %	1 t	1 t	
Skrive om prosjektorganisasjon i hovedrapport	Jim	100 %	3 t	3 t	
Skrive introduksjon til hovedrapport	Jim	100 %	5 t	3 t	
Lage sau- og bruker-ikon	Said	100 %	5 t	5 t	Ikoner til sauemeny og brukermeny
Endre klassediagram	Alle	100 %	1 t	2 t	Bytte ut ubrukte klasser med faktiske klasser
Totalt antall jobbtimer			35 t	39 t	

Sprint retrospective

Bra: Midtveispresentasjon gikk bra! Oppgavene til denne sprinten har stort sett blitt ferdig. Gruppemedlemmene har blitt flinkere til å komme tidsnok

Dårlig: Dårlig kommunikasjon i forbindelse med møteinnkalling. Status på starten av møter har vært mangefull

Bedre neste gang: Statusoppdatering må bli bedre igjen, få bedre struktur på møtene

Sprint backlog Sprint 4

Torsdag 17.10.13 til torsdag 31.10.13

Som en bonde ønsker jeg å se på et kart hvor en sau sist var registrert. Som en bonde ønsker jeg å kunne se på tidligere logger

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer	Kommentar
Sauesimulator	Edgar	100 %	5 t	6 t	Simulator som styrer saueposisjon og om sauer blir angrepet
Legge sauer i kartet	Fredrik	100 %	5 t	3 t	Få til visning av sauer i kart
Lage metode for å hente koordinater fra lokal database	Hanne Marie	100 %	1 t	30 min	
Vise, søke og filtrere logg	Hanne Marie	80 %	12 t	11 t	
Splashscreen	Said	100 %	5 t	7 t	Viser bilde når applikasjonen startes
Activitylayout	Said	100 %	9 t	10 t	Utseende på de forskjellige aktivitetslayoutene
Logg serverside	Edgar	100 %	2 t	2 t	
Lage burndownchart	Jim	100 %	2 t	2 t	
Undersøke om det er MVC vi bruker	Jim	100 %	5 t	1 t	
Lage oppdateringsknapp	Jim	100 %	4 t	3 t	Knapp som oppdaterer sauenes posisjon på kartet
Se på gårndløsninger	Fredrik	100 %	5 t	5 t	Sjekke om det er mulig å implementere gårndløsninger
Lage ny tabell i databasen som inneholder gård og lage delete-script	Edgar	100 %	1 t	1 t	
Totalt antall jobbtimer			56 t	51 t 30min	

Sprint retrospective

Bra: Bedre til å komme tidsnok

Dårlig: Mindre effektivt fordi det blir mer avansert, oppdatere burndownchart.

Bedre neste gang: Flinkere på kommentering av kode og alle må skrive dokumentasjon

Sprint backlog Sprint 5

Torsdag 31.10.13 til torsdag 14.11.13

Som en bonde ønsker jeg å få beskjed/alarm når en sau er under angrep

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer	Kommentar
Alarm	Edgar	100 %	5 t	20 t	Bonde skal kunne få alarm når sau er under angrep
Sette opp så man får automatisk gård	Edgar	100 %	1 t	1 t	Bonden skulle få en automatisk gård når man oppretter bruker
Skrive ferdig arkitekturforklaring	Jim	100 %	5 t	6 t	Skrive i dokumentasjonen
Debugge søkefunksjon	Hanne Marie/Fredrik	100 %	4 t	5 t	
Legge inn fødselsdato	Hanne Marie	100 %	3 t	3 t	Fødselsdato i stedet for alder på sau
Lage ikoner til aktivitet	Said	100 %	4 t	2 t	Lage ikoner som passer til aktiviteten man er i
Legge til antall sauer på brukerprofil	Fredrik	100 %	2 t	30 min	
Logg med oversikt over alle sauer	Hanne Marie	100 %	4 t	2 t	
Oppdatere diagrammer	Said	100 %	4 t	3 t	
Rapportarbeid	Alle	100 %	25 t	25 t	Skrive det som gjenstår i rapporten
Totalt antall timer			57 t	67 t 30min	

Sprint retrospective

Bra: Programvaren ble ferdig til deadline. Alle har jobbet mye.

Dårlig: For lange arbeidsdager

Bedre neste gang: Ingen neste gang

Sprint Backlogg 1 - 5

Innhold

Sprint backlog sprint 1.....	1
Sprint retrospective.....	1
Sprint backlog Sprint 2.....	2
Sprint retrospective.....	2
Sprint backlog Sprint 3.....	3
Sprint retrospective.....	3
Sprint backlog Sprint 4.....	4
Sprint retrospective.....	4
Sprint backlog Sprint 5.....	5
Sprint retrospective.....	5

Sprint backlog sprint 1

Torsdag 05.09.13 til torsdag 19.09.13

Som en bonde ønsker jeg å kunne logge inn på en profil med brukernavn og passord

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer
Sette opp database	Edgar	100 %	20 t	19 t
Use-case diagram	HM	100 %	1 t	1 t
Tilstands- eller aktivitetsdiagram	HM	90 %	2 t	5 t
UI design/Lage ikon	Said	100 %	2 t	5 t
Kode loggin «klassen»	Fredrik	100 %	10 t	12 t
Lage klassediagram	Alle	60 %	3 t	3 t
Lage risikotabell	Alle	100 %	1 t	2 t
Møte med kunden	Alle	100 %	1 t	30 min
Testing	Fredrik	100 %	5 t	4 t
ER-diagram	Edgar	100 %	1 t	20 min
Totalt antall jobbtimer			41	Ca 52 t

Sprint retrospective

Bra: Gruppemedlemmene som har vært til stede har jobba bra sammen, og vi har fått gjort mye.

Bra at vi traff målet. Fått litt oversikt over hvordan systemet skal være.

Dårlig: At ikke alle møter opp og at folk kommer for sent.

Bedre neste gang: Forbedre oppmøte.

Sprint backlog Sprint 2

Torsdag 19.09.13 til torsdag 3.10.13

Som en bonde ønsker jeg å kunne registrere opplysninger om mine sauene med id(navn, alder), vekt, helseinformasjon og geografisk lokasjon om en enkelt sau

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer	Kommentar
Kode brukerregistrering	Fredrik	100 %	8 t	6 t	Aktivitet som tar seg av brukerregistrering
Sauedatabase	Edgar	100 %	2 t	5 t	Tabell i databasen for sauene
Kode registrering av sauene	HM	70 %	10 t	7 t	Aktivitet som tar seg for saueregistrering
Møte med kunden	Alle	100 %	1 t	30 min	
Kode testklasse	Said, Edgar	39 %	15 t	22 t	Klasse som tester applikasjonens funksjonaliteter
Hovedrapport	Jim	25 %	5 t	4 t	Skrive introduksjon
Endre WBS	Said, HM	100 %	30 min	30 min	
Endre klassediagram	Said, HM	40 %	1 t	30 min	
Kartforbedring	Fredrik	100 %	10 t	11 t	
Lagret login	Fredrik	100 %	5 t	5 t	
Ikon toppmeny	Hanne Marie	100 %	5 t	7 t	
Totalt antall jobbtimer			62.5 t	69.5 t	

Sprint retrospective

Bra: Mye ble fort ferdig, vi har fått gjort mye, folk har vært flinke til å møte opp

Dårlig: Ikke flinke nok til å bruke trello, mange som kommer for sent

Bedre neste gang: Komme tidsnok, hjelpe de som trenger det for å komme i mål, bli flinkere til å pushe til git, bli bedre til å bruke trello

Sprint backlog Sprint 3

Torsdag 04.10.13 til torsdag 17.10.13

Som en bonde ønsker jeg å kunne slå opp i systemet og redigere informasjon om en sau og om meg selv

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer	Kommentar
Kode «redigere bruker»	Fredrik	100 %	3 t	3 t	Lage aktivitet til å redigere brukerprofil
Kode «vis brukerprofil»	Fredrik	100 %	2 t	2 t	Lage aktivitet som viser informasjon om bruker
Kode «redigere sau»	Hanne Marie	100 %	7 t	10 t	Vise liste med sauer og lage aktivitet til å redigere en sau
Validere inputfelt	Said	80 %	3 t	5 t	Skrive kode så programmet sjekker om bruker har skrevet inn riktig type i inputfelt Flyttes til neste sprint
Test til eksisterende funksjoner	Edgar	100 %	5 t	5 t	
Møte med kunde	Alle	100 %	1 t	1 t	
Skrive om prosjektorganisasjon i hovedrapport	Jim	100 %	3 t	3 t	
Skrive introduksjon til hovedrapport	Jim	100 %	5 t	3 t	
Lage sau- og bruker-ikon	Said	100 %	5 t	5 t	Ikoner til sauemeny og brukermeny
Endre klassediagram	Alle	100 %	1 t	2 t	Bytte ut ubrukte klasser med faktiske klasser
Totalt antall jobbtimer			35 t	39 t	

Sprint retrospective

Bra: Midtveispresentasjon gikk bra! Oppgavene til denne sprinten har stort sett blitt ferdig. Gruppemedlemmene har blitt flinkere til å komme tidsnok

Dårlig: Dårlig kommunikasjon i forbindelse med møteinnkalling. Status på starten av møter har vært mangefull

Bedre neste gang: Statusoppdatering må bli bedre igjen, få bedre struktur på møtene

Sprint backlog Sprint 4

Torsdag 17.10.13 til torsdag 31.10.13

Som en bonde ønsker jeg å se på et kart hvor en sau sist var registrert. Som en bonde ønsker jeg å kunne se på tidligere logger

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer	Kommentar
Sauesimulator	Edgar	100 %	5 t	6 t	Simulator som styrer saueposisjon og om sauer blir angrepet
Legge sauer i kartet	Fredrik	100 %	5 t	3 t	Få til visning av sauer i kart
Lage metode for å hente koordinater fra lokal database	Hanne Marie	100 %	1 t	30 min	
Vise, søke og filtrere logg	Hanne Marie	80 %	12 t	11 t	
Splashscreen	Said	100 %	5 t	7 t	Viser bilde når applikasjonen startes
Activitylayout	Said	100 %	9 t	10 t	Utseende på de forskjellige aktivitetslayoutene
Logg serverside	Edgar	100 %	2 t	2 t	
Lage burndownchart	Jim	100 %	2 t	2 t	
Undersøke om det er MVC vi bruker	Jim	100 %	5 t	1 t	
Lage oppdateringsknapp	Jim	100 %	4 t	3 t	Knapp som oppdaterer sauenes posisjon på kartet
Se på gårndløsninger	Fredrik	100 %	5 t	5 t	Sjekke om det er mulig å implementere gårndløsninger
Lage ny tabell i databasen som inneholder gård og lage delete-script	Edgar	100 %	1 t	1 t	
Totalt antall jobbtimer			56 t	51 t 30min	

Sprint retrospective

Bra: Bedre til å komme tidsnok

Dårlig: Mindre effektivt fordi det blir mer avansert, oppdatere burndownchart.

Bedre neste gang: Flinkere på kommentering av kode og alle må skrive dokumentasjon

Sprint backlog Sprint 5

Torsdag 31.10.13 til torsdag 14.11.13

Som en bonde ønsker jeg å få beskjed/alarm når en sau er under angrep

Oppgave	Ansvarlig	Status % complete	Planlagte jobbtimer	Faktiske jobbtimer	Kommentar
Alarm	Edgar	100 %	5 t	20 t	Bonde skal kunne få alarm når sau er under angrep
Sette opp så man får automatisk gård	Edgar	100 %	1 t	1 t	Bonden skulle få en automatisk gård når man oppretter bruker
Skrive ferdig arkitekturforklaring	Jim	100 %	5 t	6 t	Skrive i dokumentasjonen
Debugge søkefunksjon	Hanne Marie/Fredrik	100 %	4 t	5 t	
Legge inn fødselsdato	Hanne Marie	100 %	3 t	3 t	Fødselsdato i stedet for alder på sau
Lage ikoner til aktivitet	Said	100 %	4 t	2 t	Lage ikoner som passer til aktiviteten man er i
Legge til antall sauer på brukerprofil	Fredrik	100 %	2 t	30 min	
Logg med oversikt over alle sauer	Hanne Marie	100 %	4 t	2 t	
Oppdatere diagrammer	Said	100 %	4 t	3 t	
Rapportarbeid	Alle	100 %	25 t	25 t	Skrive det som gjenstår i rapporten
Totalt antall timer			57 t	67 t 30min	

Sprint retrospective

Bra: Programvaren ble ferdig til deadline. Alle har jobbet mye.

Dårlig: For lange arbeidsdager

Bedre neste gang: Ingen neste gang

Innhold

Møtereferat	1
Torsdag 05.09.13 kl 10-12	1
Til neste gang:	1
Kundemøte	2
Kundemøte #2 Tirsdag 17.09 kl 13:00-14:00.....	2
Videre:	2

Møtereferat

Torsdag 05.09.13 kl 10-12

Ikke til stede: Jim

Til i dag:

Edgar: Lest og sett på videoer om android og servere, hvordan vi skal få det til å fungere. Begynt å sette opp/lage en database.

Said: Sett på java

HM: Sett på android og litt på hvordan man lager product backlog og WBS-diagram

Fredrik: Prøvd seg på å lage android app, sett på læringsvideoer og litt på xml.

Vi ble enige om en scrum-master, Fredrik, og en referatansvarlig for møtene, Hanne Marie.

Videre satte vi opp et førsteutkast til en product backlog, og estimerte hvor lang tid vi vil bruke på hver oppgave i den første sprinten. Til estimeringen brukte vi metoden planning poker. Til slutt lagde vi et work breakdown structure(WBS)-diagram med bubbl.us (nettseite for generering av WBS).

Avtalt at vi skal ha møte hver torsdag kl. 09 – 12 og hver mandag fra 14 – 15. Med mulighet for å jobbe fra 10 – 14 på mandager.

På kundemøtet må vi spørre stud.ass. om backlogen ser grei ut og om vi burde ha med punktene fra de ikke-funksjonelle kravene. I tillegg må vi spørre hvilken informasjon som er relevant angående sauene av det som er listet opp i oppgaveteksten.

Til neste gang:

Sette opp møte med stud.ass. Møtes kl 12 på mandag.

Edgar: Fortsette på database

Fredrik: Lese om scrummaster-rolleoppgaver, sette seg inn i app-er og androidutvikling (vil til neste gang vil kunne lage en kart-veiwer)

Said: Lese om android-utvikling

HM: Lese om android-utvikling

Se neste side for kundemøte

Kundemøte

Kundemøte #2

Tirsdag 17.09 kl 13:00-14:00

Tips fra studass:

- Lage egne sprintbackloger.
 - o Eget dokument der man ser hvem som har jobbet med hva.
- Dokumenter mer!
 - o Hvorfor vi evt. ikke skal fortsette å bruke scrumwise
- Programmet trenger bare virke på pad-en.
- Gjør alle ansvarlige for noe
 - o Said designansvarlig
 - o Edgar databaseansvarlig
- Profilklassen i klassediagrammet unødvendig?
- Endre på work breakdown strukturen så vi har mer under implementering, analyse-«pakken» kan flyttes til under pre-study og legge til database-design og oppsett i strukturen
- Endre på produktbacklogen?
- Gjøre endringer på ER-diagram?

Videre:

Neste møte torsdag 18.09 kl. 10:00

Lage gant-diagram. Spørre hvordan denne sprinten var, hva vi skal gjøre bedre neste sprint og om vi har brukt kortere eller lengre tid på de forskjellige oppgavene enn vi hadde satt opp til å begynne med. Angi hvilke aktiviteter som evt. førte til overskridelse.

VERKTØY

Innhold

Verktøy	1
1. Scrumwise.....	1
2. Trello.....	1
3. Github	1
4. Eclipse.....	2
5. PostgreSQL.....	2
6. Dropbox	2
7. Google Maps API	3
8. Office Word	3
9. Office Excel og OmniGraffle Pro.....	3
10. Photoshop, Illustrator	3
11. Facebook.....	3
12. Nettbrett og android telefon til testing.....	4

1. SCRUMWISE

En av fordelene ved scrum er at hele teamet skal kunne observere og ha oversikt over hva som skjer i prosjektet til enhver tid. Det lot seg ikke gjøre å ha stand up møte hver dag, ettersom alle har forskjellige timeplaner og ikke har tid til å møtes hver dag. Vi mente derfor at vi trengte et verktøy til å holde oversikt over hvem som jobbet på de ulike oppgavene og hvordan vi lå an med sprinten fortløpende. Allerede første møte bestemte vi oss for å prøve Scrumwise som er et verktøy der all prosessinformasjon ligger lagret på et sted. Den gir god oversikt over både backlog, sprinter, taskboard, estimatorer og genererer et burndownchart. Vi valgte på slutten av første sprint å slutte å bruke Scrumwise, fordi det ble lite brukt og utover en måneds prøveperiode ville vi måtte betale. Vi slo fast at Scrumwise var unødvendig ettersom at vi brukte det så lite, og vi ønsket å bruke mer tid på å lære oss Android programmering enn å sette oss inn i Scrumwise.

2. TRELLO

For å erstatte noe av Scrumwise-funksjonene begynte vi å bruke Trello for å få oversikt over oppgaveprogresjon. Trello er et web-basert samarbeidsverktøy som organiserer prosjekter inn i "boards", eller tavler. Vi laget et nytt board for hver sprint, og la oppgaver knyttet til sprintene inn i disse. På denne måten har vi hatt oversikt over hvilke oppgaver som var påbegynt, hvilke som var ferdige og hvilke vi ikke hadde startet på. I tillegg har vi kunnet se hvem som jobbet med hva, legge til kommentarer til oppgavene og vise eventuelle oppgaver som måtte flyttes til neste sprint. Flere på gruppa hadde erfaring med bruk av trello fra før og det har derfor blitt mer brukt, i tillegg til at vi mener Trello har vært mye mer oversiktlig enn Scrumwise.

3. GITHUB

Github er hovedsakelig et web-basert versjonskontrollsyste. I den typen utviklingsprosess som vi har vært gjennom, der flere personer koder på samme kodebase samtidig, er et versjonskontrollsyste veldig viktig. Det systemet passer på er at ingen kode går tapt, man vet hvem som har gjort hvilke endringer, og den prøver å forhindre konflikter hvis to personer har jobbet på samme fil.

Måten man jobber med Github på er at hovedkodebasen ligger i skyen på Github sine servere, og når man endrer på kodebasen lager man en forgreining av denne. Dette betyr at du kopierer kodebasen ned til din lokale datamaskin, for så å gjøre endringer. Når man har foretatt seg de endringene man skulle, må man dype koden opp til Github igjen. Når dette gjøres slår man sammen endringene med den originale koden. Github vil så prøve å flette dette sammen automatisk, noe som vanligvis går veldig bra, men noen ganger klarer den det ikke. Da må man selv ta stilling til hvilke endringer/versjoner av koden som skal være med i den nyeste versjonen.

Github har flere fine funksjoner som et hvert versjonskontrollsyste burde ha; den lagrer alle endringer siden du startet prosjektet, og du kan derfor rulle tilbake til en tidligere versjon av koden om noe skulle gå galt. I tillegg til å være et versjonskontollsyste tilbyr Github en del andre funksjonaliteter. Du har muligheter for å legge til såkalte "issues" som man delegerer til de

som jobber på prosjektet. På denne måten kan du se hva hver person skal gjøre. Hver eneste kodelinje i kodebasen kan spores til en person, og man vet da hvem som har skrevet hva.

Vi fikk tilbud om en SVN server fra studieassistenten, men siden noen på gruppa hadde erfaring med Github fra før bestemte vi oss for å bruke dette. Det skal også nevnes at det er gratis.

4. ECLIPSE

Da vi bestemte oss for å gå for Androidutvikling ble vi rådet til å laste ned ADT Bundle, ettersom det egner seg veldig bra for uerfarne Androidutviklere. ADT Bundle inkluderer viktige Android SDK komponenter og en versjon av Eclipse(et kodebehandlingsprogram) med innebygd ADT(Android Developer Tools). Fordelen ved bruk av Eclipse fra ADT Bundle var at det var raskt og enkelt å komme i gang. De innebygde ADT funksjonene gjorde det lett for oss å opprette et nytt Android prosjekt, lage UI til applikasjonen og debugge applikasjonen ved hjelp av Android SDK-verktøyene. Alt koderelatert var i tillegg lett tilgjengelig innenfor programmet, noe som gjorde det veldig oversiktlig å jobbe med.

5. POSTGRESQL

For å lagre data fra flere bønder var i nødt til å ha en database. Valget vårt havnet raskt på PostgreSQL. Årsaken til det, var at den eneste i gruppa som kunne litt om databaser fra før, kjente til dette databasehåndteringssystemet. I tillegg har PostgreSQL åpen kildekode som er veldig solid og stabil, og det er ingen restriksjoner ved bruk av den. Vi har valgt å ta backup av all dataen i databasen hver natt slik at hvis en systemkrasj skulle inntrefte vil vi ha all dataen lagret på en annen server. Databasen er ikke tilgjengelig utenfra. Dette vil si at man i vårt system må gå gjennom Raspberry PI for å få kontakt med den, noe som andre databaser ikke nødvendigvis ville krevd. Ellers ville vi ikke hatt noen mulighet til å sanitere("rengejøre") SQL spørringene som kommer inn. Dette er viktig siden systemet ellers ville vært svært åpne for SQL-injeksjon – altså å bli angrepet – og resultere i at den som angriper får tilgang på sensitiv brukerinformasjon. Vi har derfor valgt å ha et mellomlag mellom database og applikasjon som saniterer data og håndterer eventuelle feil.

6. DROPBOX

For at alle på gruppa skulle få tilgang til alt vi produserer av diagrammer, dokumenter, bilder, osv., opprettet vi en felles Dropbox-mappe. Dropbox er et delingsverktøy der man lagrer filer i mapper som man har mulighet til å dele med andre, noe som gir god støtte for samarbeid.

Når vi har arbeidet med dokumentasjonen underveis, oppdaget vi en ulempe med bruk av Dropbox. Det var umulig for flere å redigere på samme fil samtidig, uten å overskrive det den siste personen hadde gjort. Dette kunne vi løst bedre ved å bruke Google Drive, men vi valgte å opprette hver sine filer, skrive ferdig for så å få dokumentasjonsansvarlig til å samle alt i et dokument. Denne metoden ble mye brukt under utviklingen av sluttrapporten. Ut over dette er Dropbox et veldig praktisk verktøy for vår bruk, som er lett å håndtere og forstå – som igjen var mye av grunnen til at vi valgte å bruke det som delingsverktøy.

7. GOOGLE MAPS API

En av hovedgrunnene til at vi valgte Googles karttjeneste, var at å implementere tjenesten på Android er ganske enkelt og sømløst, i forhold til alternativene. Vi så også på muligheten for å erstatte kartene fra Google med kartverkets kart, og da samtidig beholde kartfunksjonene i Google API. Tanken bak var at vi ønsket å sikre et mer detaljert kart, men etter å ha vært i dialog med kunden valgte vi å prioritere andre funksjonaliteter siden kartet til Google fungerte, og det gir mulighet for å se sauene i en type topografisk kart.

8. OFFICE WORD

Til å håndtere tekst, tabeller og bilder i rapporten, og til møtereferater, har vi stort sett brukt Microsoft Office Word. Dette har ikke vært helt optimalt for de på gruppa som ikke har hatt Office tilgjengelig, da formateringene ødela visningen ved bruk av andre verktøy. Alle har fortsatt kunnet lese og behandle teksten, men det kunne kanskje vært mer aktuelt å benytte LaTeX. Vi valgte å bruke Word ettersom det gjorde det lett for oss å opprette, redigere og formatere både tekst, bilder og tabeller med et visuelt tilfredsstillende resultat.

9. OFFICE EXCEL OG OMNIGRAFFLE PRO

I tillegg til Word har vi brukt Office Excel og OmniGraffle Pro til å lage tabeller og diagrammer. Vi valgte OmniGraffle som er et gratis UML-behandlingsprogram fordi det gjorde det lettere og raskere for oss å lage UML-diagrammer. Excel ble brukt for å lage og bruke en burndown-chart.

10.PHOTOSHOP, ILLUSTRATOR

For å lage ikoner, knapper og bakgrunner til applikasjonens grafiske brukergrensesnitt har vi brukt programmene Photoshop og Illustrator fra "Adobe CS6 Master Collection". Illustrator er et redigeringsverktøy for vektorgrafikk og Photoshop er et bilderedigeringsprogram. Grunnen til at vi brukte disse programmene til design, er at gruppas designansvarlig har god erfaring med dem, og at designet skulle bli så bra som mulig.

11.FACEBOOK

Mye av kommunikasjonen i gruppa, med unntak av under møter og arbeidsøkter, har foregått på Facebook. Facebook er noe vi alle sjekker jevnlig, og det ble derfor naturlig for oss å bruke dette. Vi opprettet en egen gruppe for å holde oversikt på kommunikasjon, avtale møter, gi hverandre beskjeder og spørre om ting vi lurte på.

12.NETTBRETT OG ANDROID TELEFON TIL TESTING

Vi syntes Android-emulatoren som er innebygd i Eclipse ATD var veldig treg i oppstart og under kjøring. Derfor har vi brukt telefoner og nettbrett med Android til å teste applikasjonsfunksjonene underveis. Scrum-master gikk, på eget initiativ, til innkjøp av et Android-nettbrett som har blitt benyttet som en test- og debuggingsenhett. Enheten er av typen Samsung Galaxy Tab 2(GT-P3110), og det er denne enheten vi har lagt hovedfokuset vårt på under utviklingen, ettersom kunden forsikret oss om at applikasjonen kun trengte å virke på en enhet. Det gis derfor ingen garanti for at applikasjonen fungerer optimalt på andre enheter enn denne.

Product Backlog og Burndown Chart

Innhold

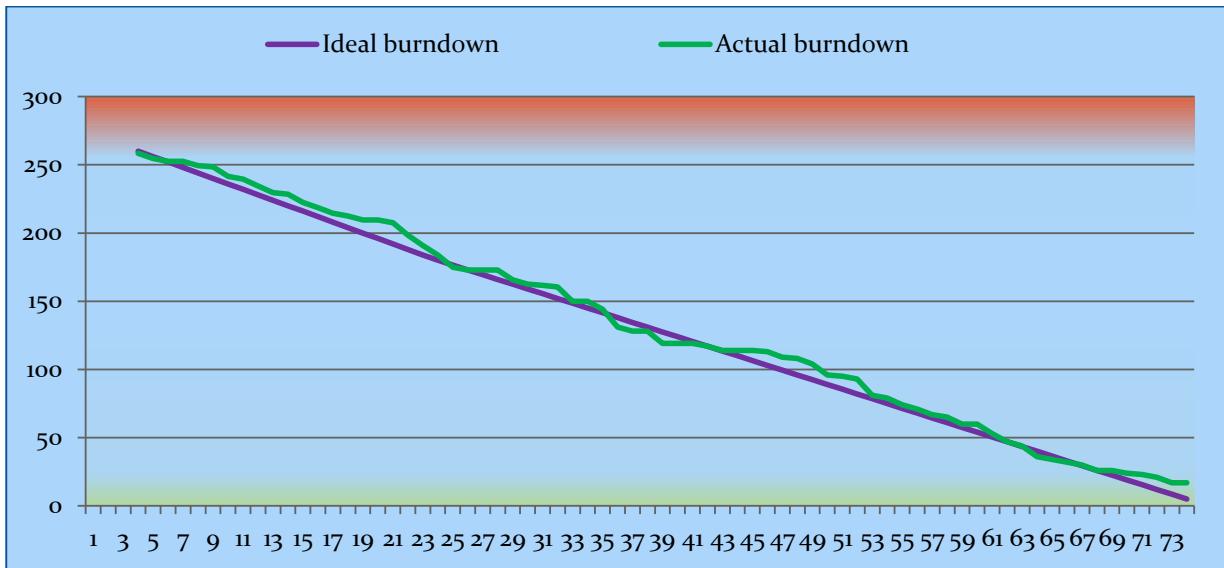
Product backlog.....	1
Burndown Chart.....	3

Product backlog

User Story	Beskrivelse	Ansvar	Rangering (1-5)	Tid estimert	Tid brukt
Sprint 1	Som en bonde ønsker jeg å kunne logge inn på en profil med brukernavn og passord				
1	Sette opp database	Edgar	5	20 t	19 t
2	Use-case diagram	Hanne Marie	1	1 t	1 t
3	Tilstands- eller aktivitetsdiagram	Hanne Marie	1	2 t	5 t
4	UI design/Lage ikon	Said	2	2 t	5 t
5	Kode loggin «klassen»	Fredrik		10 t	12 t
6	Lage klassediagram	Alle	3	3 t	3 t
7	Lage risikotabell	Alle	4	1 t	2 t
8	Møte med kunden	Alle	4	1 t	30 min
9	ER-diagram	Edgar	1	1 t	20 min
Sprint 2	Som en bonde ønsker jeg å kunne registrere opplysninger om mine sauер med id(navn, alder), vekt, helseinformasjon og geografisk lokasjon om en enkelt sau				
10	Kode brukerregistrering	Fredrik	4	8 t	6 t
11	Sauedatabase	Edgar	5	2 t	5 t
12	Kode registrering av sauер	HM	4	10 t	7 t
13	Møte med kunden	Alle	3	1 t	30 min
14	Kode testklasse	Said, Edgar	2	15 t	22 t
15	Hovedrapport	Jim	4	5 t	4 t
16	Endre WBS	Said, HM	1	30 min	30 min
17	Endre klassediagram	Said, HM	1	1 t	30 min
18	Kartforbedring	Fredrik	2	10 t	11 t
29	Lagret logg in	Fredrik		5 t	5 t
20	Ikon toppmeny	Hanne Marie		5 t	7 t
Sprint 3	Som en bonde ønsker jeg å kunne slå opp i systemet og redigere informasjon om en sau og om meg selv				
21	Kode «redigere bruker»	Fredrik	4	3 t	3 t
22	Kode «vis brukerprofil»	Fredrik	3	2 t	2 t
23	Kode «redigere sau»	Hanne Marie	4	7 t	10 t
24	Validere inputfelt	Said	2	3 t	5 t
25	Test til eksisterende funksjoner	Edgar	4	5 t	5 t
26	Møte med kunde	Alle	3	1 t	1 t
27	Skrive om prosjektorganisasjon i hovedrapport	Jim	4	3 t	3 t
28	Skrive introduksjon til hovedrapport	Jim	3	5 t	3 t
29	Lage sauе- og bruker-ikon	Said	1	5 t	5 t
30	Endre klassediagram	Alle	2	1 t	2 t
Sprint 4	Som en bonde ønsker jeg å se på et kart hvor en sau sist var registrert. Som en bonde ønsker jeg å kunne se på tidligere logger				
31	Sauesimulator	Edgar	5	5 t	6 t
32	Legge sauер i kartet	Fredrik	5	5 t	3 t
33	Lage metode for å hente	Hanne Marie	4	1 t	30 min

	koordinater fra lokal database				
34	Vise, søke og filtrere logg	Hanne Marie	5	12 t	11 t
35	Splashscreen	Said	2	5 t	7 t
36	Activitylayout	Said	3	9 t	10 t
37	Logg serverside	Edgar	4	2 t	2 t
38	Lage burndownchart	Jim	2	2 t	2 t
39	Undersøke om det er MVC vi bruker	Jim	4	5 t	1 t
40	Lage oppdateringsknapp	Jim	3	4 t	3 t
41	Se på gårdløsninger	Fredrik	2	5 t	5 t
42	Lage ny tabell i databasen som inneholder gård og lage delete-script	Edgar	2	1 t	1 t
Sprint 5	Som en bonde ønsker jeg å se på et kart hvor en sau sist var registrert. Som en bonde ønsker jeg å kunne se på tidligere logger				
43	Alarm	Edgar	4	5 t	20 t
44	Skrive om arkitektur	Jim	5	5 t	6 t
45	Debugge søkefunksjon	Hanne Marie/Fredrik	3	4 t	5 t
46	Legge inn fødselsdato	Hanne Marie	2	3 t	3 t
47	Lage iconer til aktivitet	Said	2	4 t	2 t
48	Legge til antall sauer på brukerprofil	Fredrik	1	2 t	30 min
49	Logg med oversikt over alle sauer	Hanne Marie	4	4 t	2 t
50	Oppdatere diagrammer	Said	4	4 t	3 t
51	Rapportarbeid	Alle	5	25 t	25 t
52	Sette opp så man får automatisk gård	Edgar	2	1 t	1 t

Burdndown Chart



Grafen over("Burndown Chart") representerer burndown for alle sprintene gjennom de 75 dagene vi har arbeidet med prosjektet. Den lilla linjen(Ideal burndown) representerer den ideelle tiden. Målet er å hele veien ligger under denne med den grønne(Actual burndown), som representerer den faktiske progresjonen på daglig basis. X-aksen representerer antall dager, mens y-aksen er antall arbeidstimer. Her startet vi på 260 timer, fordelt på 75 dager.

Vi har daglig lagt inn resterende arbeidstimer i en tilhørende tabell. Dette for å følge opp at man har et nyttig burndown chart hvor man kan overvåke progresjon. Vi valgt å oppdatere dette gjennom aktiv redigering i en xlsx-fil(behandles for eksempel av Excel). Her har vi satt opp en tabell med alle sprintene, som i produktbackloggen, og lagt inn resterende timer på hver oppgave hver dag. I slutten av prosjektet ble grafen seende ut som den over forrige avsnitt.

Kravspesifikasjoner

Innhold

Funksjonelle krav.....	1
De funksjonelle kravene slik de står i oppgaveteksten:	1
De funksjonelle kravene som userstories:	1
Ikke-funksjonelle krav	2
Input:	2
Forespørslar:.....	2
Dimensjonering:.....	2
Alarmer:.....	2
Dimensjonering:.....	2
Feilsituasjoner:	2
Implementasjon:	2

Funksjonelle krav

De funksjonelle kravene slik de står i oppgaveteksten:

- En bonde skal kunne registrere sine sauер med id, vekt o.l. Dersom en sau er registrert fra før skal det kunne gå an å slå opp i systemet og redigere informasjonen.
- Det skal lagres helseinformasjon, identifikasjon til sauен (navn, alder o.l.), geografisk lokasjon o.s.v.
- Bonden skal kunne sjekke på et kart hvor sau'en sist var registrert.
- Bonden skal få beskjed om en sau er under angrep.
- Bonden skal kunne gå tilbake i tid og se på logger.

De funksjonelle kravene som userstories:

- Som en bonde ønsker jeg å kunne logge inn på en profil med brukernavn og passord
- Som en bonde ønsker jeg å kunne registrere opplysninger om mine sauер med id(navn, alder), vekt, helseinformasjon og geografisk lokasjon om en enkelt sau
- Som en bonde ønsker jeg å kunne slå opp i systemet og redigere informasjon om en sau
- Som en bonde ønsker jeg å se på et kart hvor en sau sist var registrert
 - (Sauens posisjon oppdateres tre ganger i døgnet)
- Som en bonde ønsker jeg å få beskjed/alarm når en sau er under angrep
- Som en bonde ønsker jeg å kunne se på tidligere logger

Ikke-funksjonelle krav

Input:

- Input om lokalisering av de enkelte sauene kommer inn normalfordelt over et døgn.
- Bonden logger seg på systemet med brukernavn og passord.

Forespørslar:

- Responstid på forespørslar fra bønder skal maksimalt ta 2 sekunder.
- Responstid på innleggelse av lokaliseringsinformasjon om en sau skal maksimalt ta 0,5 sekund.

Dimensjonering:

- Hver sau rapporterer sin lokalisering 3 ganger per døgn.
- Hver bonde forespør lokaliseringsinformasjon om sine sauene 2 ganger i døgnet.

Alarmer:

- Innkommende alarm fra en sau gis prioritet.
- Alarmmeldinger sendes bonden og en han har valgt ut (reserve) i tillegg.
- Alarmmelding sendes ut som epost, SMS, og mobiltelefon forsøkes ringt opp.

Dimensjonering:

- Maksimalt 200 bønder skal samtidig kunne bruke systemet.
- Maksimalt 10000 sauene kan håndteres av systemet.

Feilsituasjoner:

- Ved systemfeil tillates systemet å være utilgjengelig i inntil 3 timer.
- Systemet skal ha en tilgjengelighet på minimum 95 % av tiden.
- Systemet skal være operativt døgnkontinuerlig.

Implementasjon:

- Systemet implementeres som klient / tjener-system.