



HYTTE SERVICE

2023

Gruppe 6

Fillip Skogh Linnerud - 263482

Safin Hassan - 263507

Elling Gladsø - 263504

Henrik Tanner John - 263509

Eksamensinnlevering

SYS1000 Systemutvikling

Vår 2024.

Innholdsfortegnelse

Introduksjon	4
Foranalyse.....	4
Prosjektbakgrunn og problembeskrivelse	4
Prosjektets målsetting	5
Interessent- og brukerbeskrivelser	6
Overordnet kravspesifikasjon	7
Funksjonelle krav	8
Ikke-funksjonelle krav	11
Brukerhistorier	15
Krav til Universell utforming	17
Krav til personvern	18
Hovedanalyse	19
Aktører og UseCase	19
UseCase diagram	20
UseCase beskrivelser	22
Aktivitetsdiagram.....	24
Datamodell (ER)	27
Domenemodell (UML)	28
Design	29
Systemdesign og klassediagram	29
Systemarkitektur og teknologivalg	30
Maskinvarespesifikasjon	32
Brukergrensesnitt.....	32

Prototype	33
Kvalitetssikring.....	36
Overordnet testbeskrivelse	36
Prosjektstyring	36
Systemutviklingsprosess.....	38
Systemutvikling- og prosessmodell	38
Scrum vs. Kanban	38
Vårt forslag.....	40
Prosjektrapport	44
Verktøy/teknikker	44
Individuell beskrivelse	44
Referanseliste	48
Oversikt over rapporters bilder og tabeller	49

Introduksjon

Hytteservice er et firma som tilbyr kritiske tjenester til hytteeiere over hele Norge, året rundt.

Det er under vinterhalvåret at Hytteservice merker høyest behov for deres tjenester.

Hovedsakelig involverer dette brøyting og strøing av veier opp til og rundt hytter. I tillegg tilbyr Hytteservice diverse vaktmestertjenester og levering av ved. Utover virksomhetens levetid, har de klart å operere uten et rendyrket IT system for å støtte og effektivisere deres prosesser. Dette har forårsaket unødvendig bruk av tid, penger og ressurser.

Vårt team er et av flere som har blitt kontaktet for å utarbeide et systemutviklingsprosjekt for Hytteservice. Oppgaven vil gå systematisk og punktvis gjennom de elementene som trengs for å kunne produsere Hytteservice ønskede IT system; HytteApp.

Foranalyse

Prosjektbakgrunn og problembeskrivelse

Under vinterhalvåret er brøyting en sentral aktivitet for Hytteservice sin virksomhet.

Hytteservice opplever utfordringer knyttet til manglende oversikt over hytter i bruk og behovet for deres tjenester. Den manglende oversikten resulterer i at samtlige områder, veier og parkeringsplasser brøytes ved snøfall, selv om de kanskje ikke er i bruk. Dette fører til unødvendige høye kostnader, økt tidsbruk og slitasje på utstyr.

Hytteservice ønsker å løse dette problemet ved å utvikle en applikasjon, HytteApp, som vil gi dem en oversikt over hvor det er behov for tjenester, spesielt brøyting i hytteområder.

HytteApp skal kunne forenkle prosessene, redusere kostnadene, og helhetlig forbedre effektivitet forbundet til å bestille og utføre brøytingstjenester. Samtidig skal applikasjonen øke tilgjengeligheten og kundeopplevelsen for hytteeiere ved å tilby en plattform for enkel bestilling av brøytingstjenester og kommunikasjon med Hytteservice. Den overordnede hensikten er å skape en applikasjon som kan bidra til å optimalisere ressursbruken, øke kundetilfredsheten og styrke Hytteservice sin posisjon i markedet for tjenester til hytteeiere.

Prosjektets målsetting

For øyeblikket mangler Hytteservice en strukturert oversikt over hvilke hytteområder og veier som krever brøyting. Målet med dette prosjektet er å utvikle en digital infrastruktur som muliggjør bedre oversikt for både brukere og hytteservice i forhold til hvilke tjenester som skal og har blitt utført. På bakgrunn av dette, har vi satt opp noen sentrale mål for prosjektet:

- Gi Hytteservice muligheten til å observere, hente ut og bearbeide statistiske data for å kunne organisere arbeidet sitt på en mer effektiv og økonomisk hensiktsmessig måte.
- Lage et system som forebygger overflødig bruk av ressurser som tid, penger og arbeidskraft innad i virksomheten.
- Opprette klarere og mer direkte kommunikasjonskanaler mellom Hytteservice og hytteeiere.
- Gi hytteeiere et bedre og mer raffinert brukergrensesnitt hvor hytteeiere kan administrere sin egen overordnede konto, samt ha muligheten til å opprette underbrukere for eventuelle venner og familiemedlemmer og styre deres tilganger og tillatelser.
- Gi hytteeiere en entydig bestillingsprosess som lar hytteeiere enkelt foreta bestillinger samtidig som Hytteservice tilegner seg nødvendig data.
- Gjøre rede for et design som reflekterer funksjonene i applikasjonen gjennom brukervennlighet, enkelhet og modernitet.
- Opprettholde likhet i design og funksjoner mellom mobilapplikasjonen og nettsiden.
- Forsikre at systemet følger seg etter lover og «best practice» rundt personvern og sikkerhet.

Interessent- og brukerbeskrivelser

I et system er det mange individer eller enheter som er involvert på ulike nivåer, og blir påvirket ut ifra resultatet av prosjektet. Disse kan kategoriseres som interessenter og aktører.

En aktør kan betegnes som en person eller organisasjon som har direkte kontakt med systemet. Aktører har mål knyttet til sitt ønskede system, hvilket danner grunnlag for vårt prosjekt sine overordnede krav og UseCases. I dette prosjektet betegnes Hytteservice som den eneste aktøren da det er dem som skal eie, administrere og bruke.

Interessenter kan betegnes som en person eller organisasjon som nødvendigvis ikke bruker systemet, men er i det minste nærme nok til systemet. De kan både påvirke og påvirkes av systemets utvikling og bruk. I dette prosjektet betegnes Hytteeiere, Statens kartverk og Metrologisk institutt som interessenter.

Aktører	Interessenter
<ul style="list-style-type: none">• Hytteservice	<ul style="list-style-type: none">• Hytteeiere• Statens Kartverk (Matrikkelen, API)• Metrologisk Institutt (API)

Figur 1.1 Aktør og interessenter

Beskrivelse 1 Hytteeier

Mens Henrik var på jobb, fikk han fryktelig lyst til å ta en tur på hytta. Han kommer på at han må bestille brøyting minst ett døgn før ankomst. Likevel velger Henrik å sjekke brøyteoversikten for å se om det er allerede har blitt brøytet. I brøyteoversikten står det at det akkurat ble brøytet så Henrik får heldigvis ta seg en tur på hytten allikevel.

Bekrivelse 2 Hytteeier

Fillip er en hytteeier som ønsker å sørge for at veiene til hytta hans er brøytet til rett tid. Han logger inn på HytteApp for å melde sitt planlagte hyttebesøk. Etter å ha valgt hytta si og angitt ankomstdato og tidspunkt, legger han inn mobilnummeret sitt for å motta en SMS-varslings når brøytingen er fullført. Samtidig sjekker han værmeldingen for hytteområdet for å

være forberedt på værforholdene. Med HytteApp kan Fillip enkelt administrere brøytebestillingene sine og sikre at veiene er trygge og fremkommelige når han skal til hytta.

Beskrivelse 3 Brøytemannskap

Safin er en del av brøytemannskapet hos Hytteservice, han har nettopp avsluttet brøytingen av en vei i et hytteområde. Han sjekker værmeldingen i HytteApp og oppdager en uventet stormvarsling for området han nettopp har brøytet. Safin kontakter umiddelbart hytteeieren via appen og informerer om behovet for en ny runde med brøyting. Han lover å sende en SMS-melding så snart brøytingen er fullført, slik at hytteeieren kan være oppdatert og trygg på veiforholdene.

Beskrivelse 4 HytteService Ansatt

Elling er ansatt i Hytteservice og har fått ansvaret for å optimalisere brøyteoperasjonene i ulike hytteområder. Han logger seg inn på administrasjonsmodulen i HytteApp for å analysere brøyteoversikten og se på resultatene fra de siste periodene. Elling finner ut hvilke områder som har hatt mest brøyting og når dette har blitt utført har. Basert på denne informasjonen identifiserer han mønstre og trender som kan bidra til å forbedre effektiviteten til brøytemannskapene.

[Overordnet kravspesifikasjon](#)

Hytteeier:

- Kontoregistrering.
- Registrering av profiler underlagt en og samme overordnet konto.
- Registrere og administrere hytter.
- Endring og sletting av profiler og konto.
- Tilrettelegge bestillingsfunksjon for profiler til samme hytte.
- Bestilling av Hytteservice sine tjenester.
- Sjekke hva som er brøytet og strødd (sist dato).
- Sjekke hva som skal blir brøytet snart.
- Sjekke din bestillingshistorikk.

- Oversikt over hytteområder og adresser.
- En meldingstjeneste mellom brukere og Hytteservice.
- Varslinger og oppdatering om brøytingen.
- Sjekke værmelding for hytteområde.
- Universell utforming for alle brukere.

Hytteservice:

- Administrasjonsmodul for å få oversikt over og danne statistikk om brøyting.
- Kommunisere med hytteeiere.
- Oppdatere værmelding og brøyteoversikt.
- Ikke-funksjonelle krav: GDPR-kompatibilitet, tilknytning til Statens Kartverk (Matrikkelen).

Funksjonelle krav

Funksjonelle krav beskriver hvordan disse spesifikke funksjonene i et system, programvare eller produkt skal kunne bli gjennomført. Disse kravene blir satt av aktørene. Kravene handler om hvordan de ønsker at systemet skal fungere når kunden interagerer med produktet. Et eksempel på et relevant brukerscenario i denne oppgaven er at Hytteservice vil at hytteeieren skal kunne se værvarselen til sin hytteadresse i appen.

Forretningskrav

- Brukerregistrering med autentisering
 - Hytteservice ønsker at man skal kunne registrere en bruker med logg inn system
- Bestillingsfunksjonalitet
 - Hytteservice ønsker at hytteeiere skal kunne bestille brøyting etter å ha logget inn på appen og valgt din hytte.
- Kommunikasjons mellom hytteservice og hytteeier + SMS-varslinger

- Hytteservice ønsker at man kunne sende og motta meldinger mellom begge partene via appen.
- De ønsker også at hytteeiere skal få SMS på når det er klart for ankomst til hytta. (pga. Brøting osv.)
- Værvarsel
 - Hytteservice ønsker at hytteeiere skal kunne se værvarsel for hytteadressen.
- Administrasjonsmodul
 - Dette er en modul som skal vise oversikter og kunne generere statistikk.
Denne modulen skal bare være tilgjengelig for Hytteservice administrasjonen.
- Universell utforming og krav til personvern (spesielt GDPR)
 - Godt brukergrensesnitt som er lett forståelig og tilgjengelig for flest mulig.
 - Lage retningslinjer og strukturere systemet på en måte som beskytter brukerens persondata på en sikker måte.

Brukerkrav

- Registreringsfunksjon
 - Hytteeiere ønsker å kunne registrere hytten sin for å kunne få veien måkt av brøytemannskapet
- Bestillingsfunksjonalitet
 - Hytteeiere skal kunne logge inn på appen, velge hytte og deretter bestille brøyting.
- Kommunikasjon med hytteservice med SMS-funksjon
 - Hytteeiere skal kunne sende og motta meldinger fra Hytteservice om behov
 - De skal også kunne motta SMS fra Hytteservice om tid for ankomst til hytta

- Værvarsel
 - Hytteeiere skal kunne se værvarselen for hytteadressen deres gjennom bruk av appen.
- Flere brukere per hytte
 - Hytteeiere skal kunne registrere flere brukere til samme hytte, slik at flere kan bestille brøyting og motta meldinger fra hytteservice.
- Søkefunksjonalitet
 - Hytteeiere skal kunne søke etter og registrere tilknytning til sin hytte gjennom å bruke HytteApp laget av Hytteservice.
- Brukervennlighet
 - Applikasjon skal være intuitiv og enkel å bruke, med et brukervennlig grensesnitt design
- Systemkrav
 - Administrasjonsmodul
 - Administrasjonsmodulen skal gi administrasjonen i Hytteservice tilgang på data om statistikk og rapporter på brøyteaktiviteter, som f.eks. statistikk over brøytede hytter og hytteområder.
- Feilhåndtering
 - Systemet skal kunne takle å håndtere uventede feil som oppstår uventet, uten å miste brukerdata. Det skal komme tydelige feilmeldinger når feil oppstår, deretter en veiledning for videre.

- SMS-varsler
 - Systemet skal kunne sende SMS-varslinger til hytteeiere når brøytingen av nærområder deres er ferdig, SMS-varslinger skal kunne bli sendt til et annet telefonnummer enn det nummeret som er registrert til hytteeier.
- Ytelse
 - Applikasjonen bør ha en maksimal responstid på 3 sekunder for vanlige brukerinteraksjoner. Eksempler på dette er innlogging og bestilling av brøyting.
- Tilgjengelighet
 - Systemet skal være tilgjengelig for brukeren så og si 100 prosent av tiden. Det eneste unntaket er vedlikehold av systemet.
- Sikkerhet
 - Kommunikasjon mellom Hytteapp og serveren skal være kryptert. Dette blir gjort ved hjelp av HTTPS protokollen. Formålet med HTTPS-protokollen er å sikre brukerdata mot uautorisert tilgang.
- Skalerbarhet
 - Skal takle økende brukermengder, uten at det skal påvirke ytelsen på nettsiden som f.eks. lastetider på sider og generell responstid.

Ikke-funksjonelle krav

I tillegg til de oppgavene som vi må løse (funksjonelle krav), så må vi også ta hensyn til andre krav som er da ikke-funksjonelle krav. Disse kravene beskriver hva systemet skal gjøre og hvilke egenskaper og kvaliteter det skal fokusere på, enten det gjelder en komponent eller en funksjon i applikasjonen. Sammen så utgjør disse kravene med funksjonelle krav et rammeverk som er med på utviklingen og evalueringen av systemet.

Produktkrav

- Brukervennlighet
 - Rask utføring av handlinger
 - Kunne lage brukerkonto på to minutter
 - Tydelig og brukervennlig bestillingsprosess for tjenester.
 - Enkelt valg av hytter og datoer
 - Tydelig varlingspreferanser
 - Enkelt og brukervennlig meldingssystem
 - Sikkerhet i betalingssystemet
 - Universell utforming

- Effektivitetskrav Ytelse krav og Minnekrav
 - Rask lastetid
 - God databehandling
 - Optimalisert datatrafikk
 - Fleksibel bestillingsprosess
 - Effektiv varslingsprosess
 - Rask og enkel registrering av brukere
 - Effektivt meldingssystem
 - Effektiv minnebruk
 - Minnekapasitet for lokal lagring

- Pålitelighetskrav
 - Tilgjengelighet (Høy oppetid)
 - Feiltoleranse, systemet skal håndtere feil
 - Sikkerhetstiltak for brukerdata
 - Overholdelse av personvernregler
 - Varslingssystem, varslinger leveres pålitelig og i tide
 - Dataminimering og personvern/GDPR-overholdelse

- Økt sikkerhet med 2FA (to-faktorautentisering) ^[OBJ]
- Portabilitetskrav
 - Multiplattformstøtte, brukbar for forskjellige operativsystemer
 - Responsiv applikasjon
 - Nettleserkompatibilitet
 - Dataportabilitet
 - Enkel installasjon og oppdatering
 - Språkstøtte, bør støtte flere språk
 - Åpen tilgang til meteorologisk institutts målestasjoner

Prosesskrav

- Implementasjon krav
 - Teknologivalg: Valg av moderne teknologier som støtter kravene til ytelse, sikkerhet og skalerbarhet.
 - Teststrategi: Implementering av omfattende enhets-, integrasjons- og akseptansetesting for å sikre høy kvalitet og pålitelighet.
- Leveransekrav
 - Tidsramme: Fastsettelse av klare milepæler og tidsrammer for utviklings- og leveranseprosessen.
 - Leveransemetode: Bruk av kontinuerlig integrasjon og kontinuerlig leveranse for rask og pålitelig utrulling av endringer.
- Krav til standarder

- Sikkerhetsstandarter: Implementering av beste praksis for sikker utvikling.
- Brukergrensesnittstandarter: Overholde retningslinjer for universell utforming og tilgjengelighet.

Eksterne krav

- Inter-operasjonskrav
 - Integrasjon med Statens kartverk
 - Meteorologisk Institutt API-integrasjon
 - SMS-Gateway-integrasjon
 - Databasedeling gjennom Administrasjonsmodul
 - SMS-varslingstjeneste
 - Støtte for ulike plattformer og enheter
- Etiske krav
 - Håndtere data på en etisk måte
 - Miljøvennlig, redusere ressursbruk og optimalisere energiforbruket
 - Samfunnsansvar
- Lovmessige krav
 - WCAG2.1
 - Pålitelighet
 - Miljøpåvirkning
- Sikkerhetskrav
 - Autentisering
 - Autorisasjon
 - Kryptering, sikre protokoller
 - Sikkerhetskopiering og gjenoppretting

- Sikkerhet for personopplysninger
 - Beskyttelse mot uautorisert tilgang
 - Krav til leverandører og tredjepartsintegrasjoner
- Personvernkrav
- Informasjonssamtykke
 - Rett til tilgang og retting
 - Rett til sletting
 - Varsling for brudd av personopplysningssikkerheten
 - Aldersgrense og samtykke for mindreårige

Brukerhistorier

Vi lager brukerhistorier for å gjøre det lettere å forstå de funksjonelle kravene. En brukerhistorie er en kort beskrivelse av en funksjonalitet en bruker benytter seg av når de bruker en app.

Brukerhistorier består av tre elementer. En rolle, ønske og fordel. Brukerhistoriene følger samme mønster. Vi har brukt "Som" (rolle) "ønsker jeg å" (ønske) "slik at" (fordel).

Hytteieier:

- Som hytteieier ønsker jeg å enkelt, intuitivt og raskt kunne registrere meg i appen slik at jeg kan fortest mulig bruke appen.
- Som hytteieier ønsker jeg å enkelt, intuitivt og raskt kunne registrere meg i appen slik at jeg kan fortest mulig bruke appen.
- Som hytteieier ønsker jeg å kunne legge til min(e) hytte(r) i HytteApp slik at jeg har kontroll over brøyting og andre tjenester knyttet til mine eiendommer.
- Som hytteieier ønsker jeg å kunne endre og slette informasjonen jeg har oppgitt, slik at jeg har oppdatert og riktig informasjon.

- Som hytteeier ønsker jeg å effektivt bestille brøyting for hytten min i appen, slik at brøytemannskapet kan planlegge sin aktivitet i tråd med mine behov.
- Som hytteeier ønsker jeg å kunne opprette flere brukere innenfor samme overordnede HytteApp-konto, slik at jeg kan la flere kunne bestille tjenester.
- Som hytteeier ønsker jeg å tildele ulike tilgangsnivåer til brukerne jeg legger til for min hytte i appen, slik at jeg kan kontrollere hvem som kan bestille brøyting eller få tilgang til sensitive opplysninger.
- Som hytteeier ønsker jeg å kunne se en live-oversikt over hvilke veier som allerede er brøytet, slik at jeg kan planlegge reiseruten min basert på oppdatert informasjon.
- Som hytteeier ønsker jeg å kunne motta varsler om værforholdene ved hytten min, slik at jeg kan planlegge reisen min trygt og effektivt.
- Som hytteeier ønsker jeg å kunne se en historikk over tidligere bestillinger og brøyteaktiviteter for min hytte, slik at jeg kan vurdere behovet for brøyting basert på tidligere værforhold.
- Som hytteeier ønsker jeg å kunne kommunisere direkte med brøytemannskapet via appen, samt Hytteservice selv slik at jeg kan bli motta informasjon og oppdateringer.

Hytteservice:

- Som administrator i hytteservice ønsker jeg å kunne analysere brøytehistorikken slik at jeg kan effektivisere fremtidige brøyteoperasjoner.
- Som administrator i Hytteservice ønsker jeg å kunne lage en strukturert oversikt over de kommende brøyteoperasjonene slik at det kan konverteres til en arbeidsplan basert på disse dataene.
- Som administrator i Hytteservice ønsker jeg å gi mer funksjonalitet til mine kunder gjennom innovative og lettvinte funksjoner i app, samt integrering av eksterne IT systemer som værmelding og kart slik at brukeren får best mulig opplevelse.

Krav til Universell utforming

I Norge har vi lover og bestemmelser som lager rammeverk for hvordan virksomheter skal utvikle IT-systemer med tanke på design. Disse er illustrert nedenfor.

§17 Universell Utforming:

offentlige og private virksomheter rettet mot allmennheten har plikt til universell utforming av virksomhetens alminnelige funksjoner.

med universell utforming menes utforming eller tilrettelegging av hovedløsningen i de fysiske forholdene, inkludert informasjons- og kommunikasjonsteknologi (IKT), slik at virksomhetens alminnelige funksjoner kan benyttes av flest mulig, uavhengig av funksjonsnedsettelse.

(Likestillings- og diskrimineringsloven, 2017, §17).

FN-Konvensjonen:

Med ”universell utforming” menes utforming av produkter, omgivelser, programmer og tjenester på en slik måte at de kan brukes av alle mennesker, i så stor utstrekning som mulig, uten behov for tilpassing og en spesiell utforming.” Universell utforming” skal ikke utelukke hjelpemidler for bestemte grupper av mennesker med nedsatt funksjonsevne når det er behov for det

(Regjeringen, 2008, art. 2).

Disse lovverkene er til som garanti for at alle mennesker, til tross for eventuelle hemninger, skal kunne ha tilgang til alle eller flest mulig systemer. For å opprette våre krav til universell utforming, har vi valgt å fokusere designet på innbydende og forståelig informasjon.

Krav til personvern

Tilgangskontroll: Sørge for at kun de riktige folkene har tilgang til sensitiv informasjon.

Kryptering av data: Kommunikasjon mellom app og server skal krypteres for å ta vare på sensitiv informasjon.

Back-up av data: Sørge for å gjøre kontinuerlig backup av dataen slik at det dataen kan bli gjenopprettet hvis nødvendig etter evt. datatap.

Overvåking og logging: Overvåking og logging av all systemaktivitet for å oppdage sikkerhetsbrudd og unormal aktivitet.

Sikkerhetsoppdateringer: Jevnlige oppdateringer for å være oppdatert med de nyeste teknologiene og unngå nye trusler.

Personvernloven og GDPR: Sørge for at vi følger personvernloven og GDPR. Dette er viktig når vi håndterer sensitiv informasjon.

Endring av informasjon: Gi brukeren mulighet til å se, redigere, slette og overføre personopplysninger.

Samtykkebasert behandling: Brukeren må godkjenne at HytteApp kan lagre informasjon de oppgir og la brukeren trekke tilbake dette.

Begrense datainnsamling: Kun samle inn nødvendig personopplysninger.

Brukerens rettigheter: Informere om brukeren sine rettigheter og hvordan deres data blir behandlet.

Sikker innlogging: Sikker autentisering ved bruk av sterke passord og la brukeren legge til to-faktor-autentisering.

Hovedanalyse

Aktører og UseCase

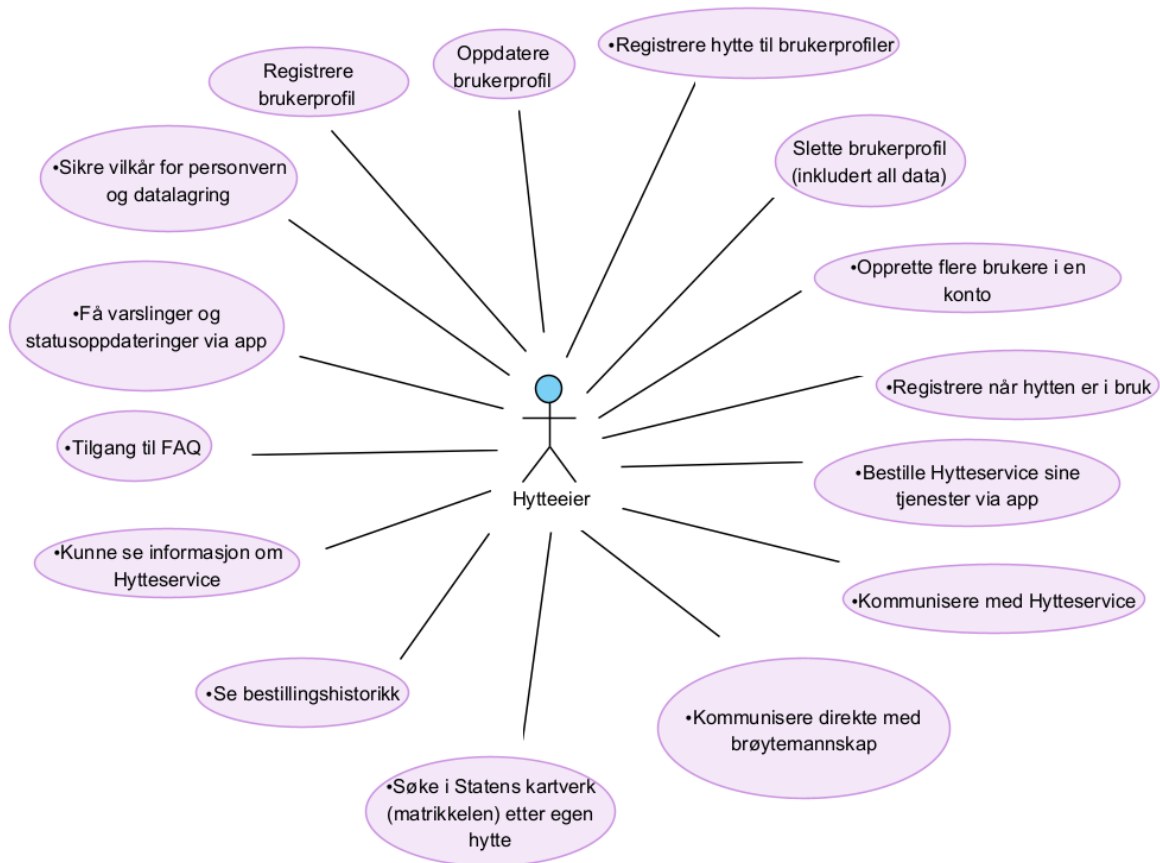
UseCase er en aktivitet hvor vi lager scenarioer av hvordan de forskjellige aktørene og interessentene kunne ønsket å bruke systemet vårt. Derav får vi en ide av hva de funksjonelle kravene til systemet bør inneholde. UseCase er en detaljert beskrivelse for hvordan brukeren skal kunne bruke systemet,

Hytteeiere	Hytteservice
<ul style="list-style-type: none">- Registrere brukerprofil- Oppdatere brukerprofil- Slette brukerprofil- Søke i Statens Kartverk etter hytte- Registrere hytte til konto- Opprette flere bruker i konto- Registrere når hytten er i bruk- Kommunisere med Hytteservice- Kommunisere med brøytemannskap- Kunne se informasjon om Hytteservice- Tilgang til FAQ- Bestille Hytteservice sine tjenester i app- Bestille Hytteservice sine tjenester via app- Få varslinger på statusoppdateringer via app- Kunne se Hytteservice sine vilkår om personvern og datalagring	<ul style="list-style-type: none">- Kunne administrere, oppdatere og endre applikasjonen- Kunne se hvilke hytter som er registrert i bruk til enhver tid- Kunne få en oversikt over hvilke bestillinger som har blitt gjennomført- Kunne utvinne statistisk rådata som gagner egen virksomhet- Kunne kommunisere med brøytemannskap- Kunne kommunisere med hytteeiere-

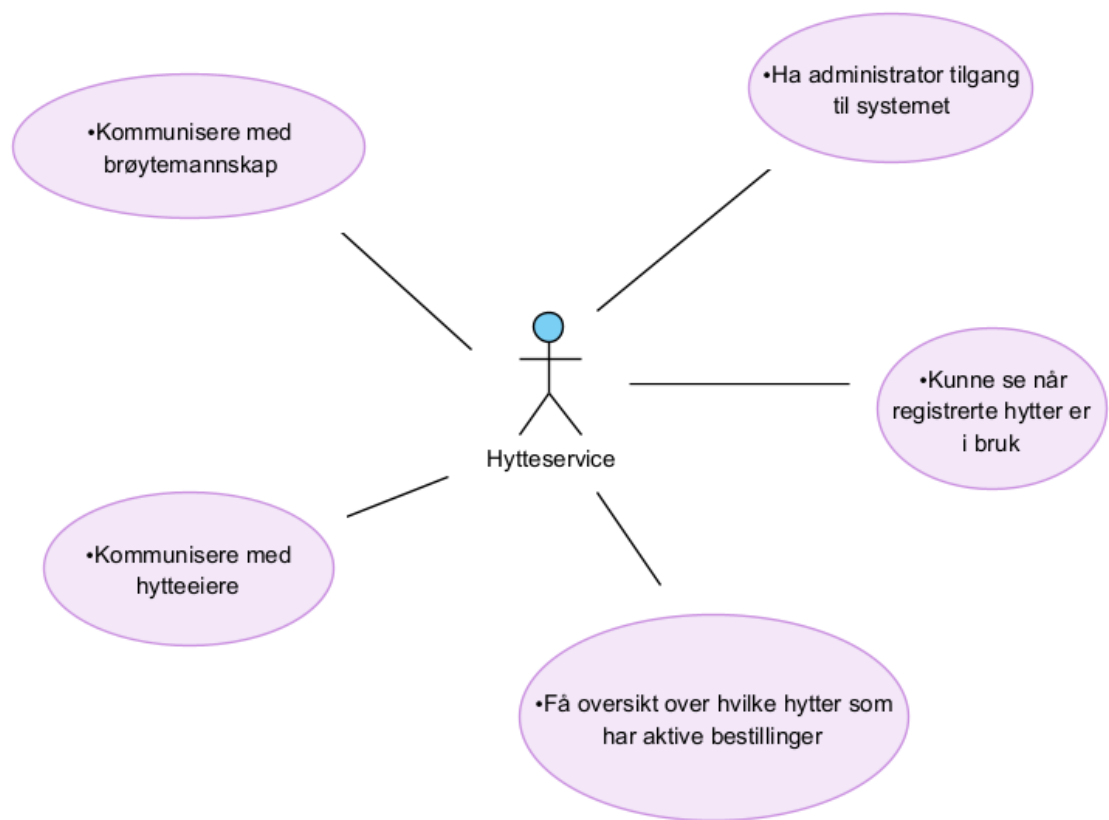
Figur 2.1 Aktør og UseCase

UseCase diagram

Vi bruker UseCase diagram når vi skal lage visualisere UseCasene. Dette kan gjøre det lettere å få en god oversikt over UseCasene.



Figur 3 UseCase Diagram 1



Figur 4 UseCase Diagram 2

UseCase beskrivelser

En UseCase beskrivelse er en detaljert forklaring av hvordan en UseCase fungerer. Det er flere trinn man må gjennom før handlingen kan bli utført, blant disse er det Pre-Betingelser, Hovedflyt (suksess) og da evt. Alternativ flyt vis det ikke er suksess. Hvis det er suksess, får man Post-betingelse altså resultatet av handlingen.

UseCase 1: Opprette brukerkonto	
Aktør	Hytteieier
Pre-betingelser	<ul style="list-style-type: none">- Hytteieieren har tilgang til registreringssystemet.- Det er ingen eksisterende konto registrert med samme brukerinformasjon.
Post-betingelser	<ul style="list-style-type: none">- Hytteieieren har en gyldig brukerkonto registrert i systemet.
Hovedflyt	<ul style="list-style-type: none">- Hytteieieren går til registreringssiden på appen.- Hytteieieren fyller ut registreringsskjemaet med nødvendig informasjon.- Hytteieieren bekrefter de gitte opplysningene.- Systemet validerer informasjonen og oppretter en brukerkonto for hytteieieren.
Alternativ flyt	<ul style="list-style-type: none">- Hvis e-postadressen allerede er knyttet til en eksisterende konto:<ol style="list-style-type: none">1. Systemet varsler hytteieieren om eksisterende konto.2. Hytteieieren kan enten forsøke å logge inn på den eksisterende kontoen eller bruke en annen e-postadresse for å opprette en ny konto.

Figur 5 UseCase Beskrivelse 1

UseCase 2: Bestilling av Hytteservice-tjenester via HytteApp	
Aktør	Hytteier
Pre-betingelser	<ul style="list-style-type: none"> - Hytteieren har lastet ned HytteApp på sin enhet. - Hytteieren har en aktiv brukerkonto i appen.
Post-betingelser	<ul style="list-style-type: none"> - Hytteieren har bestilt ønskede tjenester via HytteApp.
Hovedflyt	<ul style="list-style-type: none"> - Hytteieren går inn på appen og finner tjenesten som hytteier vil bestille. - Hytteieren angir tidspunkt og sted for tjeneste. - Hytteieren bekrefter bestillingen.
Alternativ flyt	<p>Hvis tjenester allerede har blitt utført</p> <ol style="list-style-type: none"> 1. Systemet forteller hytteieren at tjenesten har nylig blitt utført. 2. Hytteier kan avbryte eller bestille en ny tjeneste.

Figur 6 UseCase Beskrivelse 2

UseCase 3: Få oversikt over hvilke hytter som har aktive bestillinger	
Aktør	Hytteservice
Pre-betingelser	<p>Hytteservice har tilgang til systemet som administrer bestillinger.</p> <ul style="list-style-type: none"> - Det har blitt registrert bestillinger
Post-betingelser	<ul style="list-style-type: none"> - Hytteservice har fått oversikt over hvilke hytter som har aktive bestillinger.
Hovedflyt	<ul style="list-style-type: none"> - Hytteservice navigerer til systemet. - Hytteservice analyserer og filtrer bestillingene. - Hytteservice kan se informasjon om bestillingene.
Alternativ flyt	<ul style="list-style-type: none"> - Hvis det ikke er noen aktive bestillinger: <ol style="list-style-type: none"> 1. Systemet varsler Hytteservice om at det ikke er noen aktive bestillinger for øyeblikket. - Hvis det oppstår en feil under henting av bestillingsinformasjon: <ol style="list-style-type: none"> 1. Systemet varsler Hytteservice om feilen og gir instruksjoner for å rette opp problemet. 2. Hytteservice kan prøve å hente informasjonen på nytt senere eller kontakte teknisk support for assistanse.

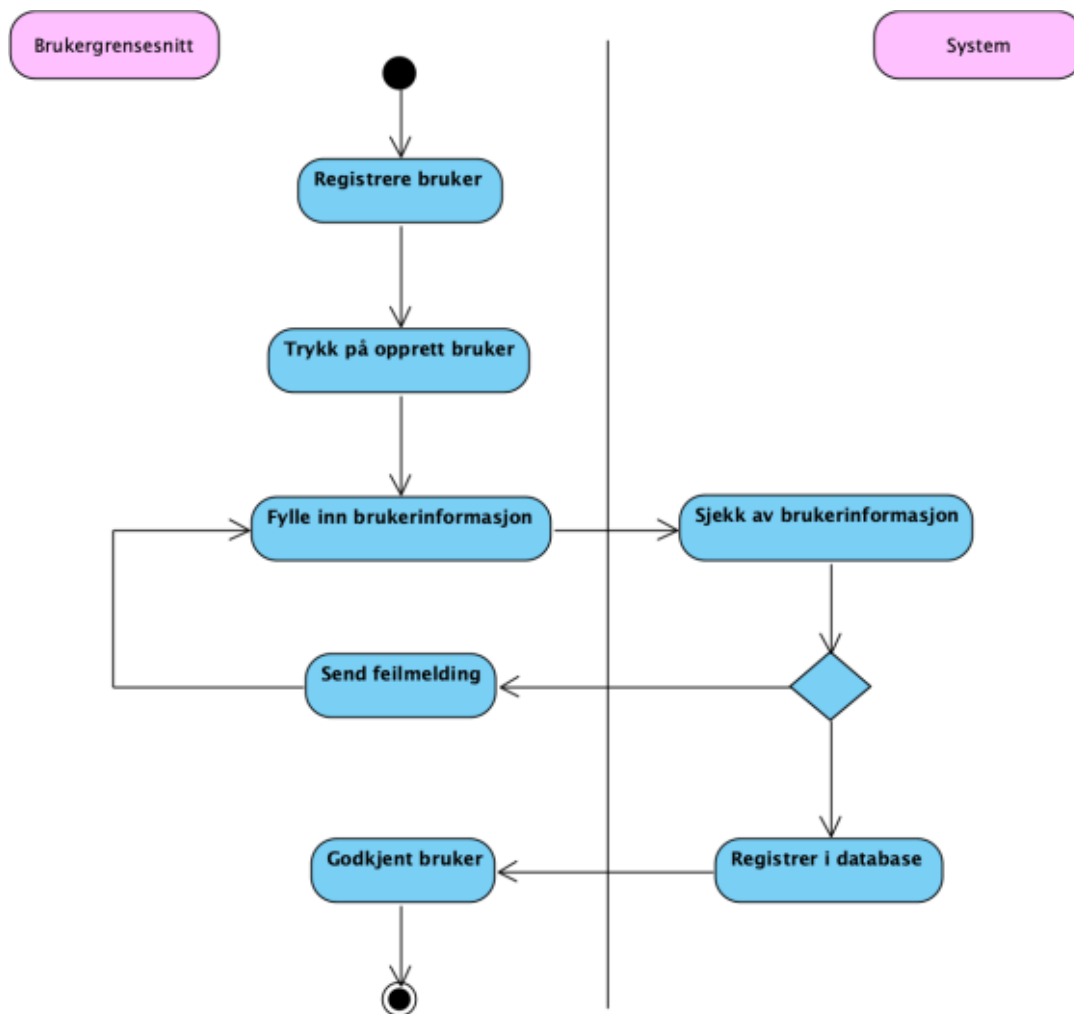
Figur 7 UseCase Beskrivelse 3

UseCase 4: Kommunisere med hytteeiere	
Aktør	Hytteservice
Pre-betingelser	<ul style="list-style-type: none"> - Hytteservice har tilgang til kommunikasjonsverktøy - Hytteservice har informasjon om hytteeieren de skal kommunisere med.
Post-betingelser	<ul style="list-style-type: none"> - Hytteservice har formidlet beskjed eller kommunisert med hytteeier.
Hovedflyt (suksess)	<ul style="list-style-type: none"> - Hytteservice velger hytteeieren de vil kommunisere med. - Hytteservice sender melding. - Hytteservice mottar eventuelle svar fra hytteeier.
Alternativ flyt	<ul style="list-style-type: none"> - Hvis det oppstår tekniske problemer under kommunikasjonen <ol style="list-style-type: none"> 1. Systemet varsler Hytteservice om feilen og gir instruksjoner for å rette opp problemet. 2. Hytteservice varsler teknisk support om problemet.

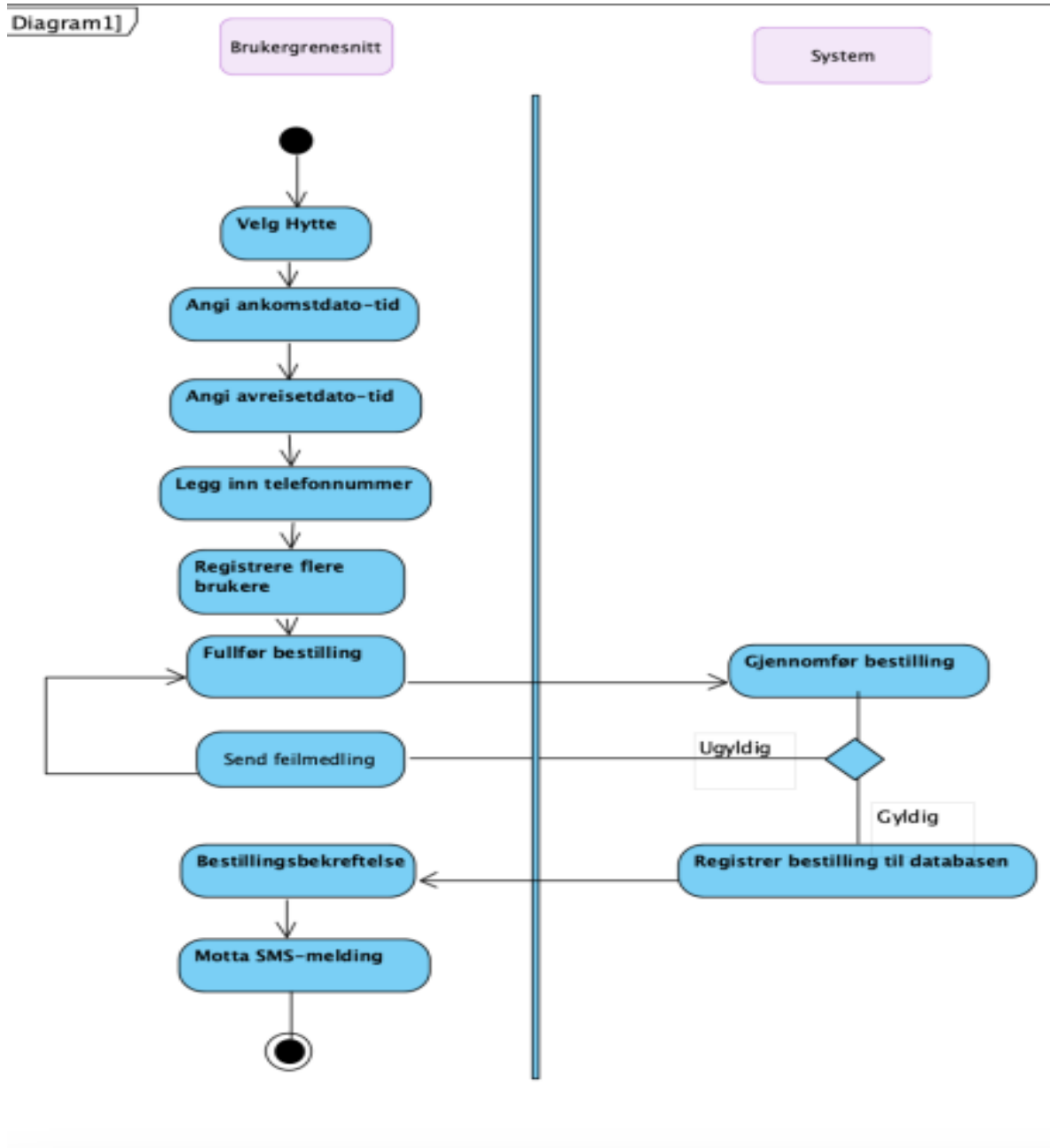
Figur 8 UseCase Beskrivelse 4

Aktivitetsdiagram

Aktivitetsdiagrammet viser handlingene eller aktivitetene som skjer i en applikasjon. Bildene nedenfor illustrerer handlingene som skjer i applikasjon til Hytteservice. Det første bilde viser hva som skjer når en person skal registrere seg en bruker på appen. Det andre bildet viser hva hytteeieren må gjøre videre for å kunne bestille en tjeneste som f.eks. brøyting eller lignende vaktmestertjenester. Bildene viser hva som skjer når man utfører handlingene både fra brukeren sitt perspektiv, men også hva som skjer i systemets interne prosesser.



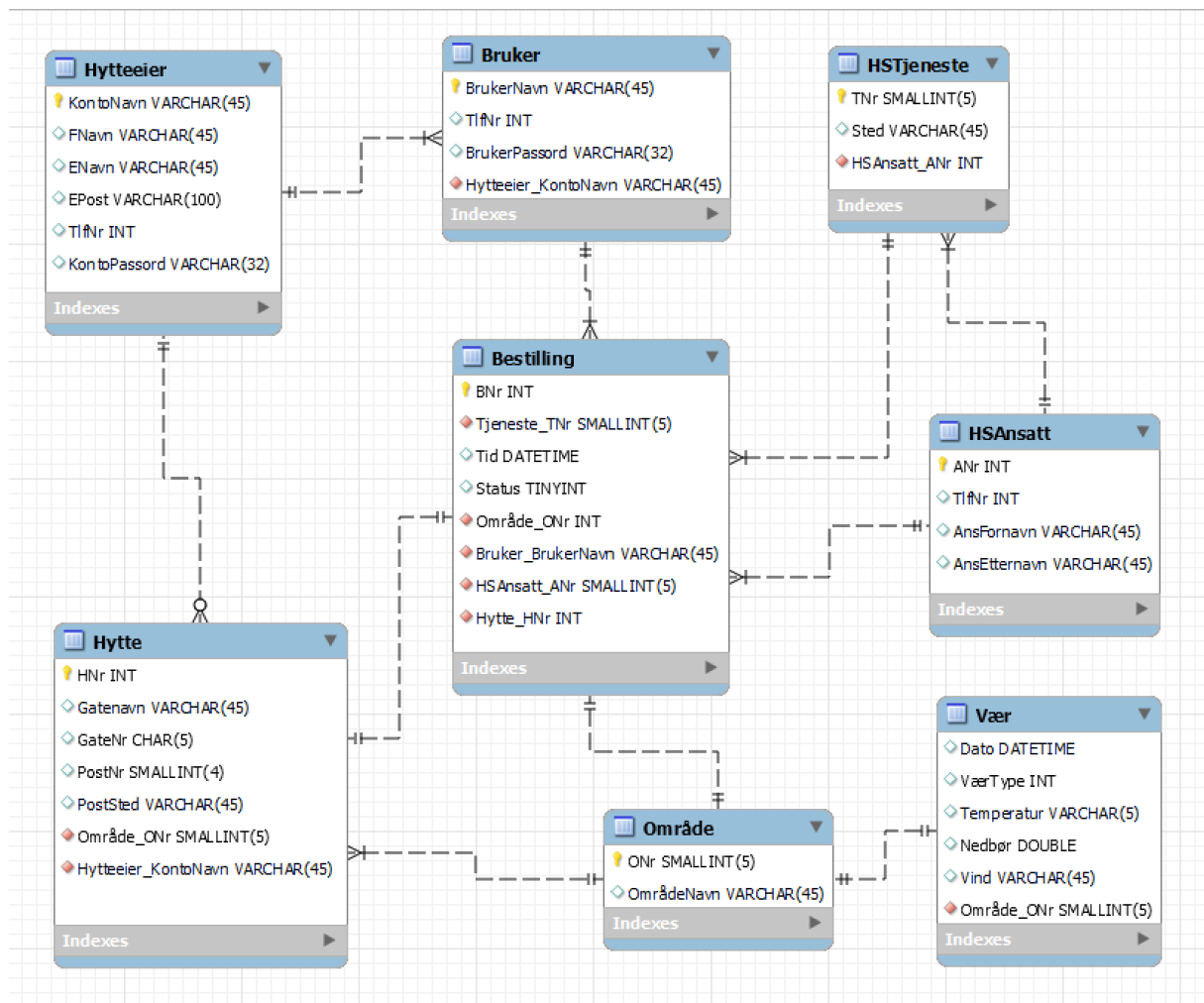
Figur 9 Aktivitetsdiagram 1



Figur 10 Aktivitetsdiagram 1

Datamodell (ER)

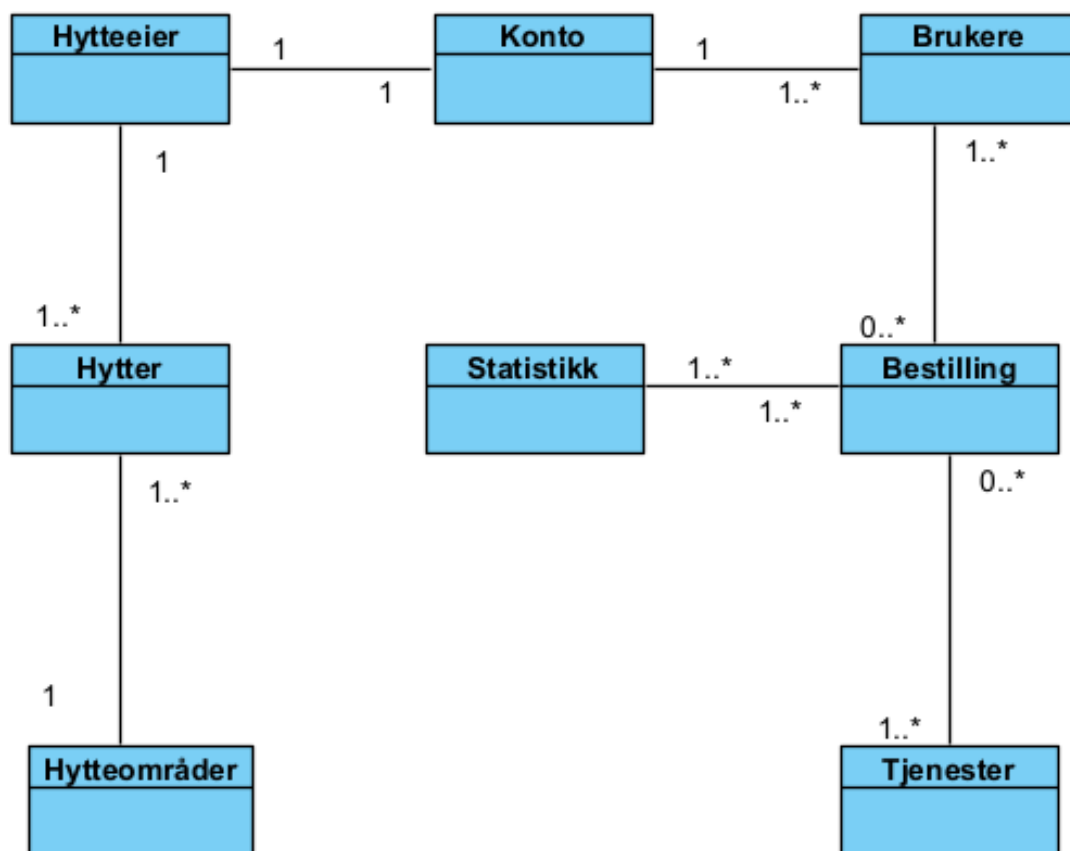
Vi bruker ER Diagram når vi skal visualisere et systems database. Ved å lage entiteter og fylle det inn med attributter får vi en oversikt over hva som skal lagres hvor. Ved å lage relasjoner mellom entitetene henger databasen sammen.



Figur 11 ER Datamodell

Domenemodell (UML)

Domenemodell er et nyttig verktøy som brukes til å visualisere og forstå objektene som systemet skal håndtere. Disse objektene blir identifisert og hentet fra kravspesifikasjonene til kunden. Ved å lage en domenemodell så gjør man det lettere å forstå objektene og hva applikasjonen skal klare å håndtere.



Figur 12 Domenemodell

Design

System design er en prosess rettet mot identifisering og utforming av ulike systemkomponenter i tråd med kundens krav og behov. Å få en dyp forståelse og presis definisjon av systeminteraksjoner er essensielt for en vellykket implementering.

Arkitekturdesign er viktig for å sette opp hvordan systemet skal se ut. Når vi identifiserer essensielle objekter, muliggjør det en metodisk analyse og intelligent utvikling av funksjonaliteter.

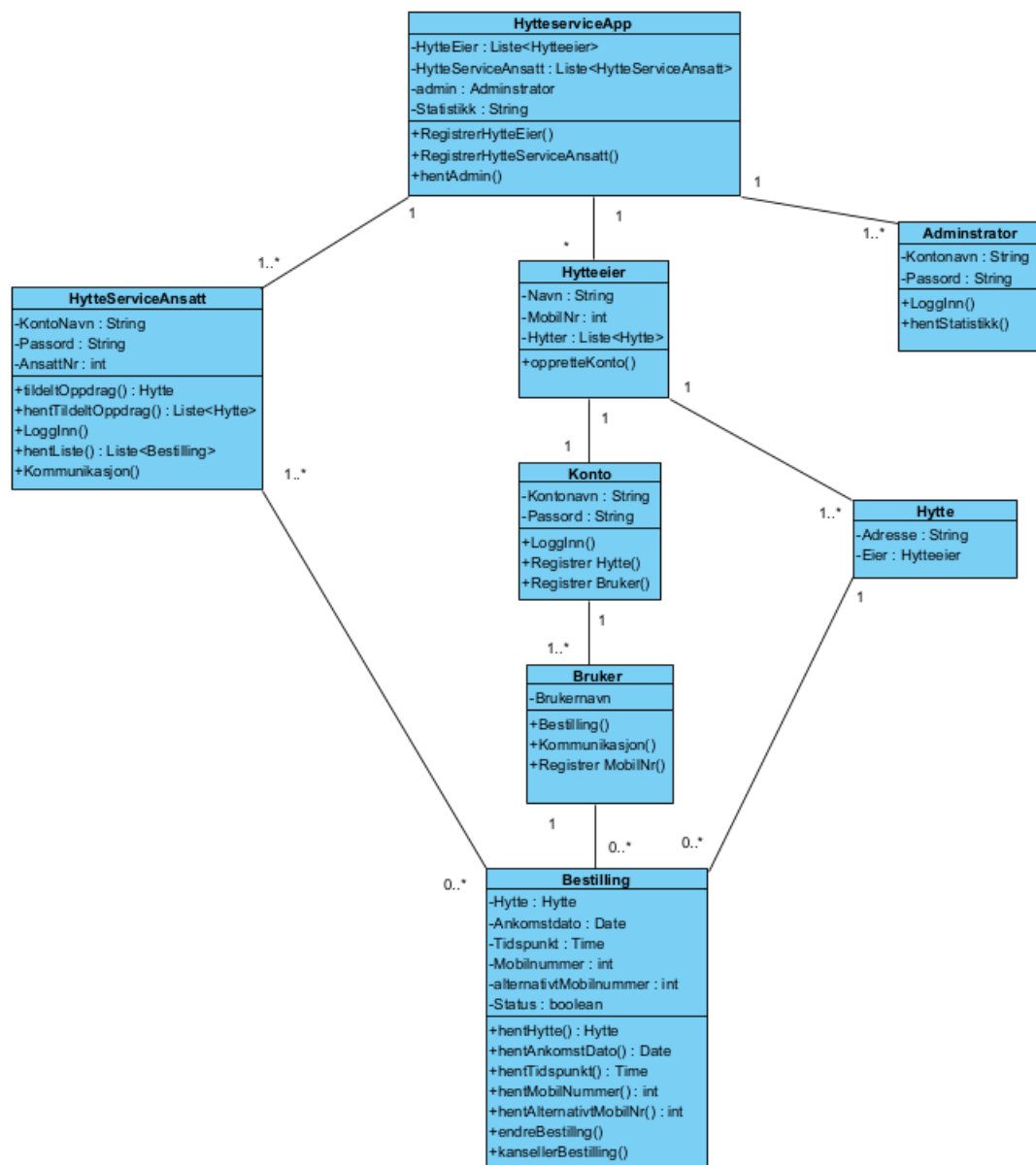
Utvikling av designmodeller gir et synlig rammeverk for å forstå systemets kompleksitet. Spesifikasjon av brukergrensesnittet er nøkkelen for best mulig brukervennlighet og funksjonalitet.

System design er avgjørende for å møte kundens behov. Hvert trinn må være nøye med planlegging for å sikre systemets implementering. Dette skaper et solid grunnlag for suksess i systemutvikling.

Systemdesign og klassediagram

Klassediagrammer viser klassene og deres forhold som brukes i systemutvikling. De utvider domenemodellen ved å utvide data og metodene. Dette gir et strukturert rammeverk for å forstå hvordan ting reagerer med hverandre. Klassediagrammer er et viktig verktøy for utviklingsteamet for å kommunisere bedre og lære om systemets dynamikk. Som et resultat er det avgjørende for vellykket utvikling av applikasjoner.

Klassediagrammet viser systemklassene, egenskapene, metodene og forbindelsene mellom dem. Ved å gi spesifikasjoner om hva hvert objekt bør inneholde og funksjonene som det utfører, utvider det domenemodellen. Sammenkoblinger viser hvordan objektene er knyttet til hverandre. Klassediagrammet gir en tydelig struktur og bidrar til at alle får en bedre forståelse av systemet.



Figur 13 Klassesdiagram

Systemarkitektur og teknologivalg

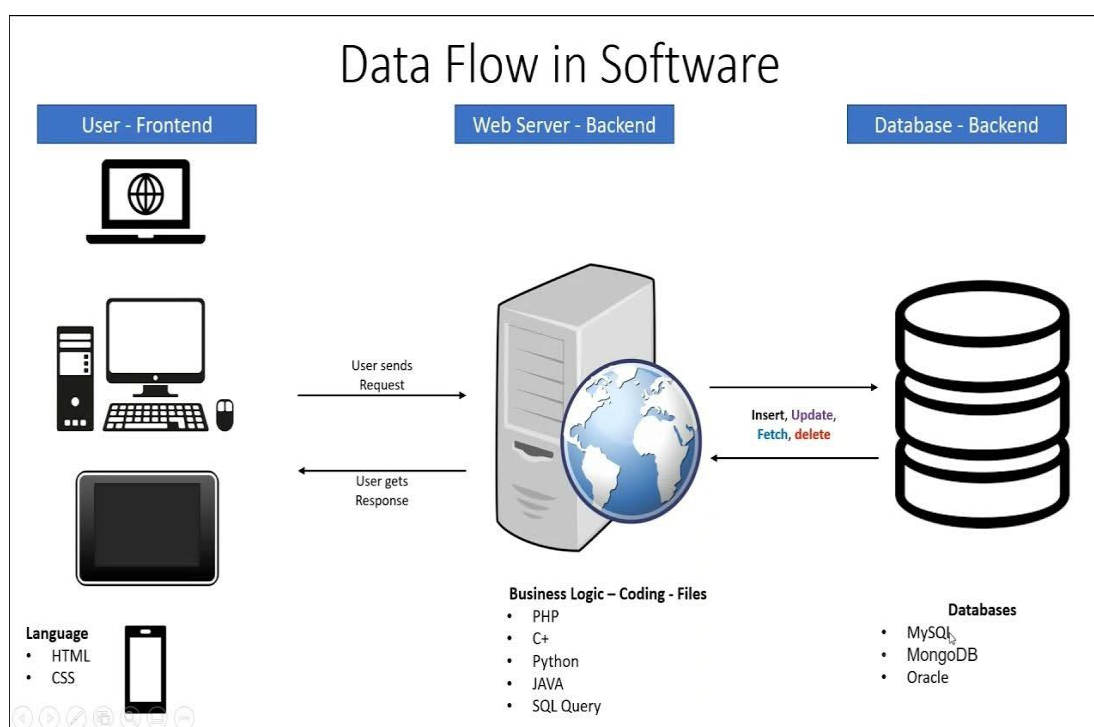
For vårt prosjekt har vi valgt å bruke en tredelt systemarkitektur. Hvert nivå i designet har ulike roller og vil bli designet på en måte hvor de minst mulig blir påvirket av hverandre.

Det første nivået er presentasjonsnivået, som består av front-end. Dette nivået kan også kalles brukergrensnittet og er det sluttbrukeren ser på sine digitale enheter. Her får sluttbruker en visuell og forståelig betraktning av applikasjonens funksjoner.

Brukergrensesnittet kan utvikles ved hjelp av HTML, CSS og Javascript.

Det andre nivået i den tredelte arkitekturen er applikasjonsnivået. Dette er en sentral del av applikasjon hvor dataen fra presentasjonsnivået blir behandlet. Her blir sluttbrukerens inndata behandlet og prosessert i tråd med bakenforliggende funksjoner definert i back-end. Programmeringsspråkene man kan bruke på dette nivået er C#, Java, C++, Ruby, Python.

Det tredje nivået i den tredelte arkitekturen er datanivået, også kalt databasen. Her blir den behandlede dataen fra applikasjonsnivået lagret og bearbeidet. Dette kan for eksempel være en relasjonsdatabase som MySQL eller Oracle.



Figur 14 Data Flow in Software

(Pramod Hanumappa, 2021, 1:40)

Maskinwarespesifikasjon

Hardware

- PC og nettbrett/mobil for de ansatte som utfører tjenester for Hytteservice, slik at de får informasjon om nye oppdrag og kan kommunisere med hytteeiere.
- Et stabilt nettverk for å koble sammen servere og brukerstyr.
- Server hvor nettsidens filer, databaser og applikasjoner er lagret, satt opp på Linux. Linux er pålitelig og er et tryggere operativsystem enn Windows.

Software

- Databasen vår bruker en MySQL-server. MySQL er lett å bruke og er pålitelig.
- Rammeverk som brukes er React Native. Det er et JavaScript-bibliotek brukt for å lage og bygge brukergrensesnitt.
- Operativsystemene som appen er ment til å kjøre på er IOS og Android. Disse er de to mest brukte operativsystemene i Norge.
- Vi bruker Xcode Instruments for feilsøking og analyse på enheter som bruker operativsystemet IOS, og Android Studio på enheter som bruker Android-operativsystemet. Eksempler på verktøy begge bruker er debugging.

Brukergrensesnitt

Designmål:

- Responsivt design.
- Lett tilgjengelig informasjon.
- Brukervennlig.
- Personlig tilpasning.
- Lett å endre informasjon.
- Kartfunksjon for å se hvor det er brøytet.
- Kunne se værdata.

Målet for HytteApp er å tilby en brukervennlig løsning for hytteeiere over hele landet.

Gjennom HytteApp skal hytteeiere kunne enkelt registrere seg, bestille tjenester, og navigere gjennom hytteområder og brøyteruter. I appen skal brukeren også kunne se værvarselen for deres hytteadresse. Systemet skal også ha en administrasjonsmodul som bare er tilgjengelig for Hytteservice-administrasjonen. Denne modulen skal vise en oversikt over hvor mange hytter som er brøytet sist periode og oversikt over hytteområder med mest brøyting sist periode.

Designet av applikasjonen vil være preget av enkelhet, oversiktlighet og brukervennlighet for alle aldersgrupper. Vi legger spesiell vekt på brukervennlighet, og ønsker å sikre at brukere har en problemfri opplevelse av HytteApp ved å gjøre det enkelt for dem å endre oppgitte opplysninger etter behov. Dette vil bidra til å være mest mulig tilfredsstillende og nyttig for våre brukere.

Prototype

Vi har brukt Figma for å lage en enkel skisse til hvordan enkelte deler av appen kunne sett ut.

Vi har tatt inspirasjon fra «Tinder» for enkelte deler av designet.

Vi valgte nøye ut fargene hvit og blå for å symbolisere vinterens skjønnhet og den avslappende atmosfæren som en hytteopplevelse kan tilby. Hvit representerer den rene snøen og den friske, åpne følelsen av en uberørt natur. Gjennom vårt valg av farger og designelementer ønsker vi å skape en hytteatmosfære som app-brukerne vil sette pris på og føle seg hjemme i. Med hver interaksjon håper vi å bringe litt av den magiske atmosfæren fra en hytteopplevelse inn i brukerens digitale verden.

HytteApp

Brøytebestilling

Hytte

Gullbringveien 30

Telefon

98002244

Ankomst

20.04.2024

Avreise

23.04.2024

☒
Legg til strøying?

Legg til en kommentar

☒
Legg til flere tjenester

Bestill

Figur 15 Figma Brøytebestilling

Brøyte Bestilling

Dette dukker opp etter at brukerne har trykket på brøytebilen nede i venstre hjørnet. Her kan brukeren bestille brøyting ved å legge inn nødvendig informasjon. Brukeren kan også bestille flere tjenester hvis brukeren ønsker det.

HytteApp

Min Profil

Navn

Henrik Tanner John

Telefon

98002244

E-post

henrik@gmail.com

Dine hytter

Gullbringveien 30, 3800, Bø.
Grivibakken 1, 3801, Bø.
Breisåstunet 4, 3802, Bø.

Brukere

Safin Doski
Fillip Skogh

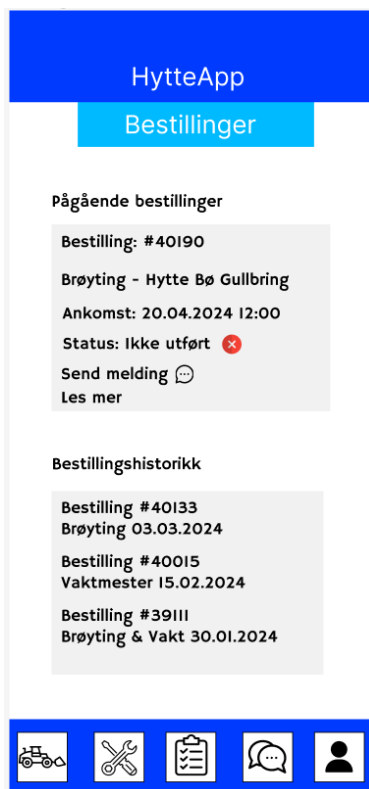
Legg til flere

Lagre

Figur 16 Figma Min Profil

Min Profil

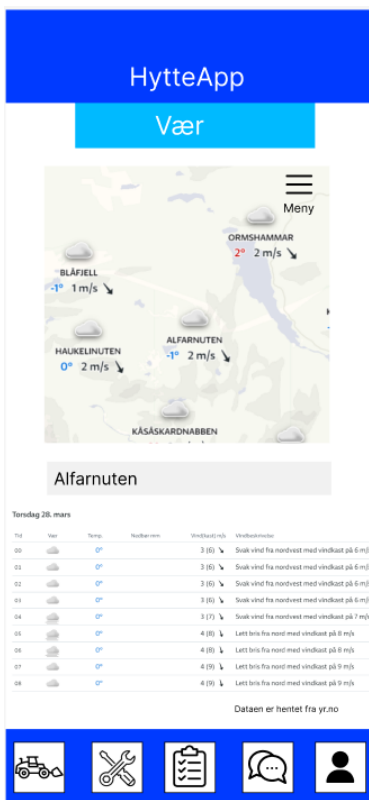
Min profil siden gjør at brukeren kan se, endre og slette på informasjonen de har oppgitt. Dette gjelder opplysninger om deg, dine hytter og hvilke brukere du har lagt til.



Figur 17 Figma Bestillingsoversikt

Bestillingsoversikt

Her kan brukeren se alle bestillingene sine. Brukeren kan se hvordan det går med de pågående bestillingene, men også eldre bestillinger som allerede er utført.



Figur 18 Figma Værvarsel

Værvarsel

Her kan brukeren se værmelding for hytten sin. Det er en søkefunksjon. Kartet viser temperatur i nærområdet. Det er mulighet for brukeren og trykke på meny knappen for å fjerne kartet og få mer detaljert informasjon om stedet brukeren har søkt. Det er et langtidsvarsel nederst på siden.

Kvalitetssikring

Overordnet testbeskrivelse

For å identifisere eventuelle feil slik at HytteApp oppfyller sitt formål før den tas i bruk, kreves det en grundig testfase. Som del av vår foreslåtte systemutviklingsplan, velger vi å ta for oss en validerings- og verifiseringsprosess.

Verifiseringsprosessen vil evaluere om vårt team sitt foreslåtte system er bygget riktig, og samsvarer med de kravene og spesifikasjonene som ble definert tidlig i utviklingsprosessen. Her vil vi gå gjennom og analysere dokumentasjonen og koden til produktet. Målet med denne prosessen er å sikre at alle obligatoriske spesifikasjoner og krav er oppfylt.

Valideringsprosessen vil evaluere om vårt team sitt foreslåtte system oppfyller Hytteservice sine behov, krav og forventninger. Målet med prosessen er å sikre at sluttbrukerens behov blir møtt gjennom bruk av produktet.

Under testfasen må appen demonstrere oppfyllelse av hvert krav, med minst én test per krav for både utviklere og hyttebrukere.

Testprosessen omfatter typisk tre hovedfaser. Først utfører utviklerne separate tester for hver komponent av hytteappen. Deretter gjennomgår appen en helhetstesting for å avdekke eventuelle feil i grensesnittet og interaksjonen mellom komponentene, samtidig som både funksjonelle og ikke-funksjonelle krav testes. Til slutt testes appen av reelle hyttebrukere for å identifisere eventuelle problemer, og sikre at deres behov blir ivaretatt.

Det er en fornuftig tilnærming å innlemme testing av hytteappens krav underveis i utviklingsprosessen, samt å involvere potensielle hyttebrukere mot slutten av prosessen for å evaluere appens funksjonalitet og brukervennlighet. Dette vil bidra til å sikre at appen oppfyller forventningene og behovene til hyttebrukerne.

Prosjektstyring

I prosjektstyring er det kritisk å sikre kontinuerlig fremdrift og å tydeliggjøre veien mot det overordnede målet. En effektiv metode for å oppnå dette er å definere ulike milepæler, som fungerer som delmål på veien mot hovedmålet - i dette tilfellet lanseringen av systemet.

Milepælene angir hva som forventes å oppnås på spesifikke tidspunkter, uten å gå i detaljer om hvordan dette skal gjøres. Vi har valgt å utvikle en helhetlig plan for denne prosjektoppgaven, som setter mål både for selve prosjektet og for fremtidige fremskritt frem til systemets lansering.

I tillegg til å etablere milepæler, er det også viktig å kontinuerlig evaluere og tilpasse planen etter behov. Dette innebærer å identifisere eventuelle hindringer eller endringer underveis og justere kursen deretter. Den helhetlige planen vår tar hensyn til denne tilpasningsdyktigheten, og gir oss et solid rammeverk for å sikre fremdrift og suksess mot lanseringen av systemet.

Systemutviklingsprosess

Systemutvikling- og prosessmodell

Hittil har det blitt illustrert prototyper av sluttproduktet. Videre fremstilles teamets foreslåtte systemutviklingsmodell og prosessmodell for dette prosjektet om oppgaven krevde prosessutførelse.

Basert på egenskapene til prosjektet slik det ble lagt frem for vårt team, mener vi det ville vært mest passende å gjennomføre prosjektet ved bruk av «agile» metoder. Nærmere bestemt tar vi utgangspunkt i den prosessfokuserte Scrum-metoden, i tillegg til implementering av Kanban-metode-elementer for økt fleksibilitet.

Scrum vs. Kanban

Scrum og Kanban er to forskjellige prosessmodeller som kan differensieres via fem forskjellige aspekter:

Metodologi

- Scrum-metoden er et ganske strukturert rammeverk for å være en agil metode. Selve prosessbeskrivelsen av en Scrum er definert gjennom sprints; Selvbestemt mengde av iterative arbeidsøkter, hvor prosessmodell-bestemte aktiviteter utføres iterativt til teamet sier seg fornøyd eller blir nødt til å avslutte på bakgrunn av mangel på tid. For hver sprint, utvinnes den spesifikke sprintens sluttprodukt, samt en liste over både gamle og nye krav/brukerhistorier. Denne listen blir videreført som mulige arbeidsoppgaver til neste sprint, og blir utført basert på Scrum-teamets valg og prioritering av kravene/brukerhistoriene. Slik bidrar Scrum-metoden til utviklingen av et produkt som blir forbedret for hver inkrementelle sprint som blir utført.
- Kanban-metoden er en mindre strukturert agil metode enn Scrum. Kanban-modellen har heller ingen prosessmodell-bestemte aktiviteter som skal utføres, men samler heller alle krav/brukerhistorier som skal tilrettelegges for i produktet og organiserer

dem i en visuell oversikt kalt Kanban tavle; En oversiktstavle hvor oppgaver klassifisert etter prioritet og sortert i kolonner som tilsier oppgavens status.

Tid

- Scrum-metoden sine sprints er betinget av tidsbokser som konvensjonelt varer mellom to til fire uker. Oppgavene i hver sprint blir også tildelt egne tidsbokser.
- Kanban-metoden er ikke knyttet til noen definerte prosessmodeller fra teorien, og har derav en mer kontinuerlig flyt av tid i sin arbeidsutførelse enn Scrum-metoden. Selv om prosjektet i sin helhet har en tidsboks, krever ikke Kanban-metoden at aktiviteter eller oppgaver får tidsbokser. Det fokuseres på å utføre oppgaver med høyere prioritet først på et effektivt vis.

Roller og ansvar

- Scrum-metoden tildeler medlemmer spesifikke roller og ansvarsområder innenfor et team. Dette danner utgangspunktet for Scrum-teamets evne til å kunne levere sitt produkt.
- Kanban-metoden tar ikke utgangspunkt i at medlemmer skal tildeles spesifikke roller og ansvarsområder. Istedenfor bygger metoden på autonomi ved at gruppen selv skal kunne ta valget om de ønsker å organisere seg, og i så fall til hvilken grad av organisering som er ønsket.

Planlegging

- Scrum-metoden planlegger arbeidsutførelsen i form av sprinter. I hver sprint befinner det seg utvalgte arbeidsoppgaver som blir klassifisert ut ifra bl.a. prioritet og tidsboks. Det er vanlig at oppgaver med høyest prioritet blir valgt som de tidligste oppgavene i en Scrum.

- Kanban-metoden samler alle oppgaver på en Kanban-tavle som gir oversikt over mengden med oppgaver, deres prioritet, og den helhetlige flyten til prosjektet ettersom oppgaver blir fullført.

Fleksibilitet

- Scrum-metoden er en mindre fleksibel agil metode. Hver sprint i en Scrum tildeles spesifikke oppgaver. Selv om oppgavene i teorien kan endres, er dette frarådet da dette kan påvirke Scrum-teamets fokus og forståelse av sprintens mål. Scrum-metoden åpner heller for endringer i oppgaver etter en sprint retrospektiv blir foretatt hvor Scrum-teamet får muligheten til å se over sitt arbeid, fremlegge det til sine aktører og interessenter, og oppdatere sine oppgaver basert på eventuelle endrede prioriteter eller ny informasjon.
- Kanban-metoden er en mer fleksibel metode. Oppgaver er ikke låst innenfor prosessmodell-spesifiserte aktiviteter, men er tilgjengelige utover hele prosjektet. Metoden tillater kontinuerlige endringer av prioriteter, oppdateringer av oppgaver og inkludering av nyere oppgaver fortløpende utover et prosjekt.

Vårt forslag

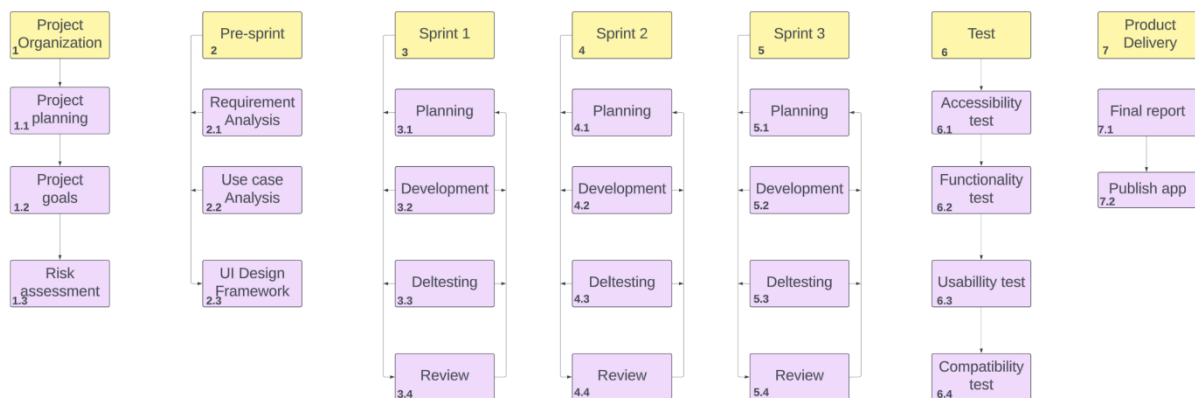
Som nevnt ovenfor, ville vår arbeidsmetode tatt utgangspunkt i Scrum-metoden, med implementerte elementer av Kanban-metoden.

Scrum er en agil metode som har trekk av ikke-fleksible elementer. Dette gjør metoden godt egnet for vårt team som hadde liten til ingen erfaring med å jobbe ut ifra agile metoder, men trengte en mer effektiv arbeidsmåte gitt tidsboksen til prosjektet.

Scrum-teamet ville bli organisert i definerte roller, men ikke ansvarsområder. Vi oppretter altså et hierarki innenfor Scrum-teamet, men holder ansvarsområdene åpne slik at alle oppgaver blir et allment gjøremål.

Arbeidsutførelsen til Scrum-teamet vil foregå i «fleksible sprints». Med dette mener vi en konvensjonell sprint modell, med trekk av Kanban-metoden. Hver sprint vil ha

forhåndsdefinerte arbeidsmål og -oppgaver. Disse vil bli plukket fra en Kanban-tavle ut ifra prioritet. Kanban-tavlen kan oppdateres av Scrum-teamet, men aktør kan bare observere og kommentere. Dette gjøres for å minimere sannsynligheten for «scope creep». Samtidig tenker vi dette øker sannsynligheten for at aktør blir mer fornøyd med prosjektets resultater gjennom egen involvering i systemutviklingsfasen.



Figur 19 WBS

Her har vi formulert en WBS for systemutviklingsprosjektet. Vi har delt den opp i fem hovedpunkter. Sprint 1-3 blir her slått sammen som en helhetlig Scrum.

Fase 1: Prosjekt organisering: Første fase av prosjektet vil bestå av planlegging, målsetting og risikoanalyser. Her legger vi grunnmuren for arbeidet fremover ved å definere og bane veien videre i prosjektet. Denne fasen er kritisk da mye av arbeidet gjort her vil være kilden for arbeidet som gjøres utover i prosjektet

Fase 2: Pre-Sprint: Andre fase av prosjektet vil omhandle forberedende arbeid for den tredje fasen; Scrum. Her vil vi foreta en UseCase analyse for å utvinne krav for applikasjonen. Disse kravene vil så konverteres til brukerhistorier som blir lagt inn i en Kanban-tavle i form av oppgaver som skal løses. Basert på rådataen fra UseCase analysen, vil en prototype av

applikasjonen bli lagd. Denne prototypen vil brukes som en mal for den faktiske applikasjonen som blir utviklet utover tredje fase.

Fase 3: Scrum: Tredje fasen av prosjektet går ut på utviklingen av applikasjonen. Arbeidet i denne fasen vil bli inndelt i sprints. Hver sprint får tildelt sine oppgaver ut ifra prioritet fra Kanban-tavlen. Aktøren vil også kunne følge med på arbeidet som blir utført i denne fasen, og komme med oppdateringer underveis eller i etterkant av en utført sprint i «Sprint Retrospective». Oppdateringer fra aktøren vil bli vurdert og bare iverksettes om tidsboksen til sprinten forblir uendret.

Fase 4: Test: Fjerde fasen av prosjektet går ut på testing av applikasjon som ble utviklet og ferdigstilt i den tredje fasen. Applikasjon vil testes ut ifra fire aspekter; tilgjengelighet, funksjonalitet, brukervennlighet og kompatibilitet.

Tilgjengelighetstesten vil teste hvorvidt applikasjonen kan brukes uavhengig av et menneskets eventuelle kognitive eller funksjonsmessige hemninger.

Funksjonalitetstesten vil teste hvorvidt applikasjonen har møtt aktørens funksjonelle krav.

Brukervennlighetstesten vil teste hvorvidt brukere av applikasjonen opplever brukergrensesnittet som intuitivt og praktisk.

Kompatibilitetstesten vil teste hvorvidt applikasjonen kan brukes på samme måte på tvers av forskjellige operativsystemer utover forskjellige brukerutstyr.

Alle testene vil bli utført gjennom utfylling av spørreskjema. Om mulig, ville vi likt å kunne velge ut spesifikke respondenter til vårt spørreskjema basert på et bredt spektrum av alder og funksjonsevne for å kunne utvinne et mer nøyaktig resultat.

Fase 5: Produkt leveranse: Femte og siste fase av prosjektet går ut på fullføring og publisering av systemutviklingsprosjektet. Her vil vi kunne fremlegge artefaktene som vårt arbeid i prosjektet har utvunnet; Dokumentasjonen av prosjektet og kildekoden for applikasjonen.

Dette inkluderer alle relevante rapporter, planer og analyser for prosjektet. Gjennom disse får vi en oversikt over prosjektets fremgang og historikk, og alle nødvendige data for videre utvikling av systemet i senere tid.

Prosjektrapport

Verktøy/teknikker

Vi begynte med å opprette en Messenger-gruppe for å kunne planlegge møter og kunne holde kontakten. Alle er bosatt i Bø så vi har hatt flere ukentlige møter. Etter hvert lagde vi en Discord-server for å gjøre det lettere å laste opp filer og ha mer arbeidsrelaterte diskusjoner. Vi benyttet oss av Discord i feriene for å ha nettbaserte møter når vi ikke kunne ha fysisk oppmøte.

Vi benyttet oss av delt Word-fil for å skrive rapporten. Dette gjorde at vi alle kunne være i samme Word-fil på likt og at den ble kontinuerlig oppdatert.

For å lage de nødvendige diagrammene i rapporten, tok vi i bruk flere verktøy. Det vi brukte mest var Visual Paradigm, som vi brukte til å lage UseCase-diagrammer, aktivitetsdiagrammer, domenemodell og klassediagram. Visual Paradigm er veldig brukervennlig og ga oss akkurat den funksjonaliteten vi trengte.

For utviklingen av ER-diagrammet valgte vi å bruke MySQL Workbench. Vi hadde allerede noe erfaring med å lage ER-diagrammer i MySQL Workbench fra høstsemesteret med DAT1000 (Database).

Vi tok i bruk Figma for å lage en skisse av appen vår. Figma er brukervennlig og man får fort gode resultater med dette verktøyet.

Vi tok også i bruk LucidChart for å lage WBS.

Individuell beskrivelse

Fillip Skogh Linnerud

- Aktivitetsdiagram
- Overordnet kravspesifikasjon
 - o Funksjonelle krav
- ER-datamodell

- Interessenter
- Systemarkitektur
- Maskinvarespesifikasjon

Etter å ha jobbet med dette eksamensprosjektet, har jeg lært om alt fra krav til brukergrensesnitt til teori om funksjonelle krav. Jeg har erfart hvor essensielt funksjonelle krav er, fordi dette definerer funksjonaliteten til systemet. Jeg har også erfart hvor viktig god kommunikasjon er for et gruppeprosjekt, både med tanke på sluttresultatet, men også for hele prosessen i seg selv.

Elling Gladsø

- Problembeskrivelse
- Bakgrunn
- Hensikt
- Brukerbeskrivelser
- Krav til personvern
- Brukerhistorier
- UseCase
 - Diagram
 - Beskrivelse
 - Aktører og UseCase
- ER-datamodell
- Domenemodell
- Klassediagram
- Brukergrensesnitt
- Prototype
- Verktøy/teknikker

Gjennom arbeidet ved å lage brukerhistorier i dette faget så har det hjulpet meg å få en bedre forståelse av brukerhistorier, det har bidratt til at jeg fikk skrevet bedre brukerhistorier i PRO1000. Erfaringen min med å bruke Figma for å utvikle prototypen til HytteApp har

forbedret kompetansen min innenfor design. Denne erfaringen hjalp meg i PRO1000, hvor vi har brukt Figma i store deler av prosjektet.

Gjennom dette prosjektet har jeg lært at det er viktig å få en god felles forståelse av oppgaven, gjennom å skrive en nøye problembeskrivelse, bakgrunn og hensikt til prosjektet gjorde det at hele gruppa fikk en bedre forståelse av oppgaven. Jeg har også blitt en bedre medarbeider ved å forsøke og sette meg godt inni oppgaven til mine medarbeidere når de trenger hjelp. Dette har bidratt til bedre arbeidsmiljø.

Gjennom dette prosjektet har jeg bedre forståelse av systemutvikling og prosjektledelse. Jeg har blitt bedre på å analysere og forstå problemer. Jeg har lært å bruke nye verktøy som jeg er sikker på kommer til å være relevante for meg senere i karrieren.

Safin Hassan

- Ikke funksjonelle krav
- Aktivitetsdiagram
- ER-datamodell
- Bakgrunn, hensikt,
- Krav til personvern
- Brukergrensesnitt
- Prototype
- Verktøy/teknikker

Gjennom dette prosjektet har jeg styrket min forståelse av brukerhistorier og design ved å jobbe med Visual Paradigm og Figma.

Det har også lært meg viktigheten av å tydeliggjøre oppgavens mål og bakgrunn for å sikre felles forståelse i gruppen. Jeg har forbedret samarbeidsferdighetene mine ved å dykke inn i medarbeidernes oppgaver når nødvendig, noe som har skapt et bedre arbeidsmiljø.

Dette prosjektet har gitt meg innsikt i systemutvikling og prosjektledelse, samt forbedret mine analytiske ferdigheter. Jeg har også blitt kjent med nye verktøy som vil være relevante

for fremtidige prosjekter. Spesielt har jeg lært betydningen av funksjonelle krav og god kommunikasjon i gruppearbeid, både for resultatet og prosessens suksess.

Henrik Tanner John

- Problembeskrivelse, bakgrunn, hensikt og målsetting
- Overordnet kravspesifikasjon
- Krav til universell utforming
- ER-datamodell
- Maskinvarespesifikasjon
- Overordnet testbeskrivelse og testplan
- Systemutviklingsprosessmodell

SYS1000 faget har gitt meg en forbedret og fordypet kunnskapsbase innenfor systemutviklingsemnet. Faget har lært meg hvordan jeg kan på forskjellige måter bygge planer og organisere arbeid basert på forskjellige nøkkelfaktorer. I tillegg har jeg tilegnet meg en ny respekt for dokumentering og testprosessene som følger et systemutviklingsprosjekt.

Gjennom SYS1000 prosjektoppgaven, føler jeg at jeg har erfart mye og tilegnet meg sentrale ferdigheter innen prosjektplanlegging, team-arbeid og problemløsning. Fagkunnskapen i SYS1000 har vært en kritisk faktor for min egen del i hvordan jeg har klart å håndtere prosjektoppgaven på en mer effektiv måte.

Referanseliste

- Sandnes, F. (2022). *Universell utforming av IKT-systemer : brukergrensesnitt for alle* (3. utg.). Universitetsforlaget
- BARNE-, LIKESTILLINGS- OG INKLUDERINGSDEPARTEMENTET. (2008). *Konvensjon om rettighetene til mennesker med nedsatt funksjonsevne*. Regjeringen.
https://www.regjeringen.no/globalassets/upload/bld/sla/funk/konvensjon_web.pdf
- Likestillings- og diskrimineringsloven. (2017). *Lov om likestilling og forbud mot diskriminering*(LOV-2017-06-16-51). Lovdata.
<https://lovdata.no/dokument/LTI/lov/2017-06-16-51>
- Insightsoftware.(2021, 16.februar). *5 Benefits of a 3-Tier Architecture*.
<https://insightsoftware.com/blog/5-benefits-of-a-3-tier-architecture/>
- IBM.(u.å.) *What is three-tier architecture?*.
<https://www.ibm.com/topics/three-tier-architecture>
- Lucidchart. (n.d.). Hentet fra
<https://www.lucidchart.com/pages/>
- Pramod Hanumappa. (2021, 14. sep). *What is Frontend and Backend : Data Flow in Software: What is Software* [Video]. YouTube.

https://www.youtube.com/watch?app=desktop&v=YGxrvHGCJ2Y&ab_channel=PramodHanumappa

Oversikt over rapporters bilder og tabeller

Figur 1 Aktør og interessenter	6
Figur 2 Aktør og UseCase	19
Figur 3 UseCase Diagram 1.....	20
Figur 4 UseCase Diagram 2.....	21
Figur 5 UseCase Beskrivelse 1	22
Figur 6 UseCase Beskrivelse 2	23
Figur 7 UseCase Beskrivelse 3	23
Figur 8 UseCase Beskrivelse 4	24
Figur 9 Aktivitetsdiagram 1.....	25
Figur 10 Aktivitetsdiagram 1.....	26
Figur 11 ER Datamodell	27
Figur 12 Domenemodell.....	28
Figur 13 Klassediagram.....	30
Figur 14 Data Flow in Software	31
Figur 15 Figma Brøytebestilling.....	34
Figur 16 Figma Min Profil	34
Figur 17 Figma Bestillingsoversikt	35
Figur 18 Figma Værvarsel	35
Figur 19 WBS	41

