

UKE 14

IT-Kontrakter + Repetisjon

Modul B

IN1030 - Gruppe 8

Program for i dag

Hva er kontrakter?

Hva er spesielt med IT-kontrakter?

Repetisjon for Modul B (modellering og systemutvikling)

Tidligere eksamensoppgaver

Kontrakter

“En kontrakt er en avtale som mellom partene etablerer en bindende forpliktelse til å gjøre eller til å unnlate å gjøre noe.”

Fra forelesning 3/5 2022

Tilbud + Aksept = Avtale

Ingen formkrav til de fleste typer kontrakter

- *Muntlig / Skriftlig blant mange former*

Kontraktens innhold

- *Partene / Leveransen / Fremdriftsplan/ Prismodell*
- *For leveranse: Bistandsforpliktelse / Spesifisert resultat / Definert tjenestenivå eller kvalitet*
- *For prismodell: Variabel pris / Fast pris / Målpris / Ytelsesbasert pris*

Hva skriver man kontrakt på?



Resultat

Pris og/eller tidsestimat

Grad av fleksibilitet ifht
produktbeskrivelsen

mm.

Utfordringer i IT-verdenen:

Spesielt i IT-verdenen:

- Komplexitet
- Utvikling av noe nytt
- Sosiotekniske systemer: Skal bruke av mennesker og kan innebære endringer i arbeidsprosesser og organisering
- Abstrakte og usynlige systemer.
- Mangel på modenhet i IT-bransjen.



Kunde

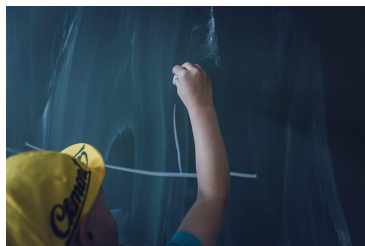


Leverandør

Kontraktutfordring i IT-verdenen:



Kunde: *Vet ikke alltid hva de vil ha eller hvilke ressurser som må til for å skape it-tjenesten.*



Leverandør: *Forsøker å forklare og tegne opp estimater, men klarer ikke alltid å forklare godt.*

Felles utfordringer:

- Uklar målsetting og manglende avgrensning
- Udefinerte suksesskriterier
- Usikkerhet håndteres ikke underveis
- Mange endringer underveis i gjennomføringen
- Systeminnføring blir undervurdert – ofte betydelige krav til omstilling i organisasjonen
- Manglende kompetanse og prosjekterfaring hos deltagerne
- Dårlig kommunikasjon mellom kunde og leverandør
- Prosjektene blir for store og komplekse
- Erfaringer underveis blir ikke tilstrekkelig hensyntatt

Kontrakter fra programvareutvikling

Resultatansvar/ "Fossefall"

- Spesifisert resultat
- Fast pris
- Høy grad av forutberegnelighet, men begrenset fleksibilitet

Iterativgjennomføring smodell / "Serie- fossefall"

- Definert gjennomføringsmodell og løpende spesifisering av delleveranser
- Målpris / Estimeringsmodell
- Forutberegneligheten øker og fleksibiliteten reduseres underveis i gjennomføringen

Bistandsforpliktelse / "Ressurskjøp"

- Avtalt hvilke ressurser og omfang de skal arbeide, men uten spesifisert resultat
- Betaling etter medgått tid og kostnader
- Høy grad av fleksibilitet, men begrenset forutberegnelighet

Den perfekte kontrakten?

Fossefall:

- *Vet hva vi skal ha og at det ikke er altfor omfattende.*
- *Når resultatet er klart spesifisert, og omfang og kompleksitet er begrenset.*

Seriefossefall:

- *Ganske klart hva vi skal ha, begge parter evner å følge et rigid system for gjennomføring.*
- *Når resultatet kan defineres på overordnet nivå og partene er enige om å følge en avtalt gjennomføringsmodell hvor resultatet i hver delleveranse spesifiseres underveis.*

Ressurskjøp:

- *Vet mindre/ikke villig til å låse oss til hva slags produkt det skal skrives kontrakt for.*
- *Når det skal utvikles etter smidig metode eller resultatet av andre grunner ikke er klart definert.*

Smidig kontraktsmodell:

Ligner ressurskjøpsmodellen; leverandør er ikke ansvarlig for kontrakt, men for ressurser, gjennomføringsmodell og definerte ikke-funksjonelle krav.

Hvordan velge kontraktform?

En balance mellom å velge fleksibilitet og forutsigbarhet

"Avoid the tail wagging the dog"

Det som skal leveres
bør bestemmes
kontraktvalget, ikke
motsatt...

Hva skal leveres?

Hvordan skal det leveres?

Hvilket ansvar skal leverandøren påta seg?

Hvilken usikkerhet skal plasseres hos leverandøren?

Prismodell?

Valg av kontraktstype

Hvilken mal?

Norske standardkontrakter

Spesifisert resultat og fastpris

SSA-T

“Avtalen er egnet der leverandørens spesifiseringsarbeid (utarbeidelse av detaljspesifikasjon) ønskes gjennomført i nært samarbeid med kunden.”

anskaffelser.no

Gradvis spesifisert resultat og pris

PS2000

PS2000SOL

Bistand betalt etter medgått tid

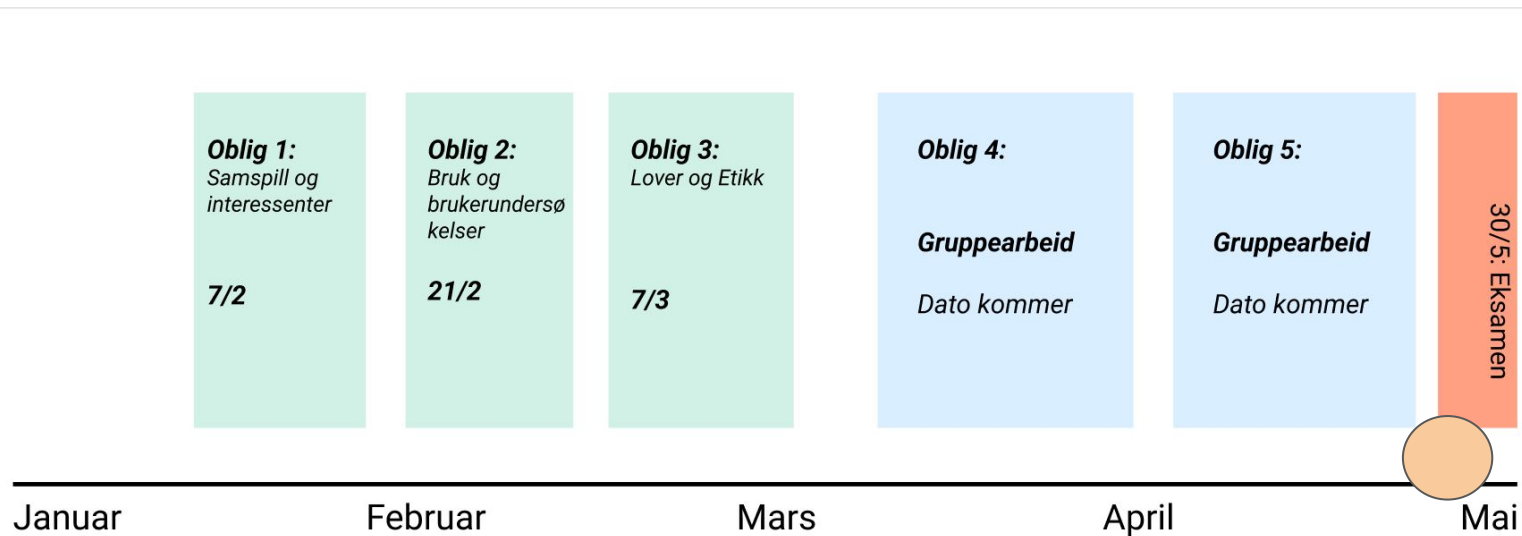
SSA-B

“Avtalen er egnet til konsulentkjøp når du har behov for kompetanse, men ikke vet hvordan sluttresultatet skal bli.”

anskaffelser.no

REPETISJON

Modul B



Modul A

Teknologi og Samfunn

Modul B

Systemutvikling

Repetisjon

Læringsmål

Etter å ha fullført IN1030:

- kan du drøfte samspillet mellom digital teknologi og individer, organisasjoner og samfunnet
- kan du utføre enkle brukerundersøkelser
- kjenner du til sentrale lover og forskrifter for utvikling av digitale systemer, og kan drøfte etiske problemstillinger
- **kjenner du til ulike faser og aktiviteter som inngår i systemutvikling**
- **har du forståelse for samspillet mellom systemutvikling og ulike bruker og interessegrupper**
- **kan du anvende metoder og teknikker for kravhåndtering, utføre modellering ved hjelp av UML, og vurdere fordeler og ulemper ved forskjellige metoder og teknologier for systemutvikling**

Felles repetisjon av Modul B i grupper

Dere inndeles i 3 grupper. Hver gruppe får et læringsmål.

Noter og diskuter for det givne læringsmålet:

- Hvordan ville dere oppsumere dette læringsmålet (forklart til medstudenter)?
- Hvilke stikkord og begreper er relevante + hva betyr de? **Skriv kort forklaring.**
- Hva har vi lært fra dette læringsmålet?
- Hvordan relaterer dette læringsmålet seg til hva deres ellers har lært i IN1030?

Lag et (eller fler) slide med oppsummeringen

Viktig: dere har bare tilgang til lysarkene når dere logger inn med uio-eposten (med G-suite aktivert)

Spørsmål?

Menti: 5514 7487

<https://www.menti.com/eo8obmgfb4>

NØKKELBEGREPER

Systemutvikling

Prosessmodeller

Smidig utvikling

Plandrevet utvikling

Reell prosess

Scrum

Kanban

Fossefallsmodellen

DevOps og kodehåndtering

Kravanalyse

Funksjonelle- og ikke-funksjonelle krav

Kontrakt

Kostnad

Prosjektplanlegging

UML modellering

Leverandør

Kravhåndtering

Implementering

Kunde

Risikoanalyse

Trusselmodellering

Testing

Produkteier

Vedlikehold

Hvilke hovedaktiviteter inngår i en systemutviklingsprosess?

Planlegging

Kravinnsamling

Kravanalyse

Design

Programmering

Testing

Konfigurasjonsstyring

Versjonshåndtering

hva som skal lages og innenfor hvilke rammer/krav.

design og programmering.

validerer at systemet er det kunden vil ha.

modifiseres etter kunden og markedets krav/behov.

Hvordan velge riktig prosessmodell?

Hva slags system skal bygges?

System av det annet system

Individuelle applikasjoner

Interaktive transaksjons-baserte applikasjoner

→ vurder: hva har vi allerede, hvilken kontekst skal vi inn i, hvem er kunden/bruker og hvilken interesse har kunden/bruker?

Spørsmål: Forskjellen på interessent og aktør?

Hvordan påvirker aktører og interessenter valg av systemutviklingsprosess?

Aktør: noen (kan også være et annet system) som bruker systemet aktivt. Enten: ha eget mål med å anvende systemet (primæraktør), Eller: hjelper primæraktøren med å oppnå sitt mål i systemet (sekundæraktør).

Interessent: blir påvirket eller påvirker systemets utvikling og/eller drift.

Plandrevet:

- Tydelig mål fra aktører
- Gjerne få interessenter og aktører
- Vanskelig å opprettholde kommunikasjon med aktør.

Smidig:

- Tvetydige mål
- Gjerne mange aktører og interessenter
- Effektiv kommunikasjon med aktører

MEN OBSOBS: Kunde har alltid "final say" ! Og det finnes gjerne en kontrakt.

Smidig vs Plandrevet: hovedforskjeller

Smidig:

- Dynamisk samarbeid med kunde/produkteier
- Dynamisk reviderbar kravspesifikasjon
- Prioriterer å håndtere kravendring sammen med kunde
- Inkrementell levering av produktets funksjoner

Plandreven:

- Må holde seg til planen til en utgave av ferdig produkt er levert (evt endringshåndtering mulig, men tidkrevende)
- Fokus på dokumentering av prosess
- Lengre og mer detaljert for- og kravanalyse.
- Statisk kravspesifikasjon
- Prioriterer å utvikle systemet basert på forhåndsbestemt plan
- Oftest kun ett endelig produkt

Hvordan bestemme seg for Smidig eller Plandrevet?

Sterkt forsimplet

Oftest: vil kravspesifikasjonen endre seg?

Ja: smidig. Nei: plandrevet.

Også: er det et stort system som krever masse ressurser (penger, tid, utviklere) og forutsigbarhet?

Ja: plandrevet. Nei: smidig.

Hvordan bestemme seg for Scrum eller Kanban?

Vurder følgende:

1. Trenger vi Daily-Standups?
2. Trenger vi retrospektiv?
3. Trenger vi struktur?
4. Er det tydelig hvilke oppgaver som får oss frem til målet?
5. Klarer vi å estimere hvor lang tid oppgavene tar?
6. Trenger vi å kunne gjøre endringer *nårsomhelst* i utviklingsprosessen?
7. Trenger kunde en garantert inkrementell levering av produktet?

Hvis svaret er ja:

Hvilke taler for Scrum?

Hvilke taler for Kanban?

Hvis svaret er ja:

Scrum: 1., 2., 3., 5., 7.

Kanban: 4., 6.

Krav

Lage kravspesifikasjon - hvordan?

Mål: både funksjonelle og ikke-funksjonelle krav.
Gjerne også måloppnåelse og testing av kravene

Brukerhistorier - basert på identifiserte interessenter → identifiserer primær- og sekundæraktører og funksjonelle krav (i hovedsak).

Kravspesifikasjonen kan ha enten veldig naturlig språk, eller veldig unaturlig. Bør følge et system.

Funksjonelle krav:

Hva skal systemet gjøre? Hvilke mål skal det tilfredstille? Hvilke funksjoner skal det ha?

Ikke-funksjonelle krav:

Produktkrav: hvordan skal produktet fungere?

Organisatoriske krav: hvordan skal systemutviklingsprosessen gjennomføres?
Interne krav?

Eksterne krav: hvilke krav har eksterne omgivelser (konteksten)?

Skal være mål/testbare - presise!

Eksempler:

Funksjonelt krav: Systemet skal generere oversikt over mest brukte kinoer.

Ikke-funksjonelle krav:

Produktkrav: Nettsiden skal håndtere opptil 5000 samtidige brukere.

Organisatoriske krav: Systemutviklingen skal holde et budsjett på maks 30 millioner NOK.

Eksterne krav: Systemets betalingsløsning må følge NF-kinos krav til samarbeidspartnere.

Eksempel med WCAG

Funksjonelle krav:

- Siden/nettstedet skal vise innhold

Ikke-funksjonelle krav:

- *Hvordan skal innholdet vises?* Her kommer eksempelvis WCAG-prinsippene inn:

At innhold skal vises etter disse retningslinjer og med disse formål

Eksempler på tester:

Stresstesting

Brukertesting

Sjekklister

Direkte målbart (ja/nei)

Ekspertisehjelp

UML-modellering

Oversikt over diagrammer

Use case diagram: Viser interaksjon mellom et system og omgivelsene. Tar utgangspunkt i primæraktørs mål og hvordan sekundæraktører assisterer dette målet gjennom systemet.

Henger sammen med tekstlig beskrivelse.

Brukes når: Man ønsker å se interaksjonen mellom aktør og system.

Sekvensdiagram: Viser interaksjon og informasjonsflyten mellom aktørene og systemet og systemkomponentene i form av objektklasser. Et kodenært diagram (eks. bruker metodekald) (mer detaljert enn use case diagrammene)

Henger sammen med tekstlig beskrivelse og klassediagram.

Brukes når: Man ønsker å se på interaksjonen på et mer systemnært nivå.

Klassediagram: Viser struktur: objektklasser av et system, deres attributter og metoder, og assosiasjonene mellom klassene.

(**Domenemodell:** Klassediagram uten metoder)

Brukes når: Man ønsker å få en oversikt ved å vise alle klasser, metoder, assosiasjoner osv. i systemet, og vise arkitekturen.

Aktivitetsdiagram: Viser aktivitetsflyten i en prosess eller dataprosessering. Grafisk representasjon av hendelsesflyten i et use case.

Brukes når: Man ønsker å vise flowet, prosesser eller omgivelser for et system.

Tekstlig beskrivelse: En tekstlig beskrivelse av en use case tar for seg interaksjonen mellom systemet og bruker, ved en nummerert liste som beskriver hvert interaksjonssteg for seg.

OBS: Det er viktig at alle modeller for et og samme system samsvarer !

En metode som blir brukt i et sekvensdiagram må også være med i et klassediagram.

USE CASE DIAGRAM

Må med: aktør og use case

Er sekundæraktør med? OBS: kun hjelpe.

Interessenter er *ikke* aktør

System er *ikke* aktør

Ikke-funksjonelle krav er *ikke* use case.

Sammenheng med funksjonelle krav !

MULTIPLE CHOICE:

Hva innebærer include-relasjonen?

- a. At man inkluderer flest mulig primærbrukere i et use case og viser relasjonen mellom disse
- b. At man inkluderer sekundærbrukere i et use case og viser relasjonen mellom primær og sekundærbrukere
- c. Et use case som kan være en del av ett eller flere andre use case
- d. Et use case som beskriver tilleggsoppførsel som utføres under gitte omstendigheter

Hva innebærer extend-relasjonen?

- e. At man inkluderer flest mulig primærbrukere i et use case og viser relasjonen mellom disse
- f. At man inkluderer sekundærbrukere i et use case og viser relasjonen mellom primær og sekundærbrukere
- g. Et use case som kan være en del av ett eller flere andre use case
- h. Et use case som beskriver tilleggsoppførsel som utføres under gitte omstendigheter

Brukerhistorier → Use-case diagram

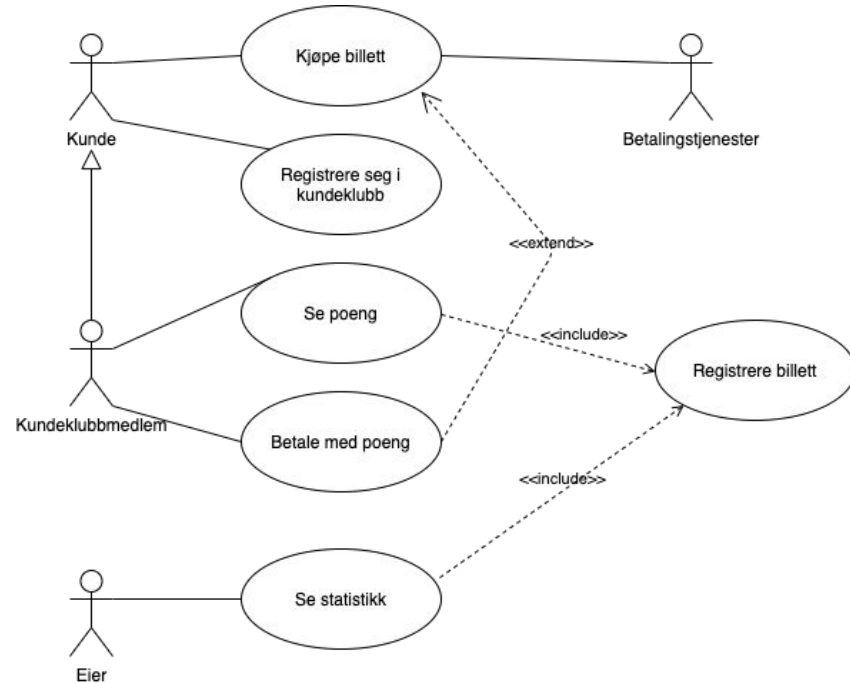
Som kunde ønsker jeg å kjøpe billett slik at jeg kan gå på kino

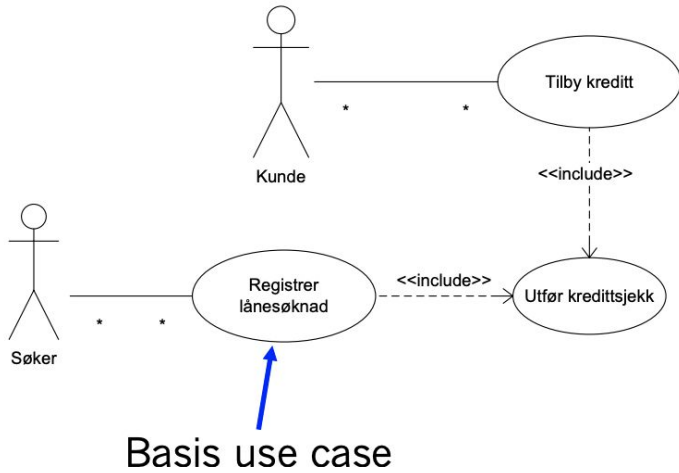
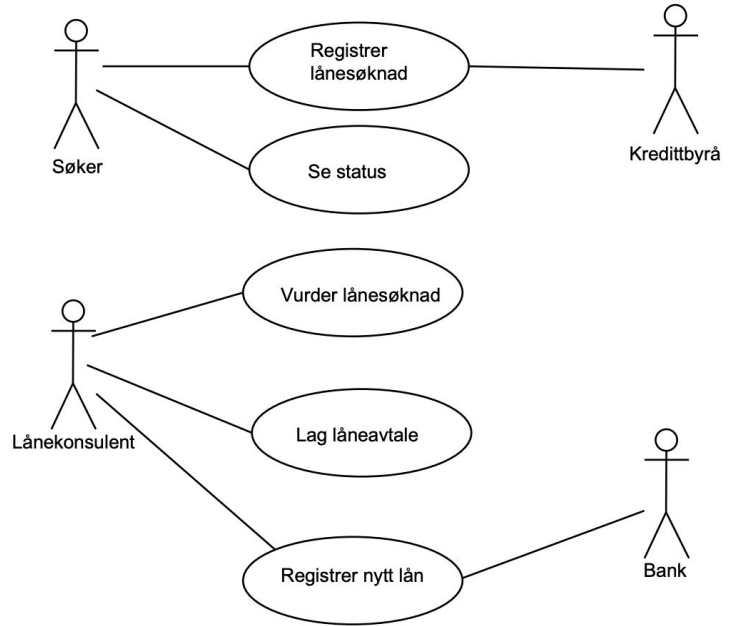
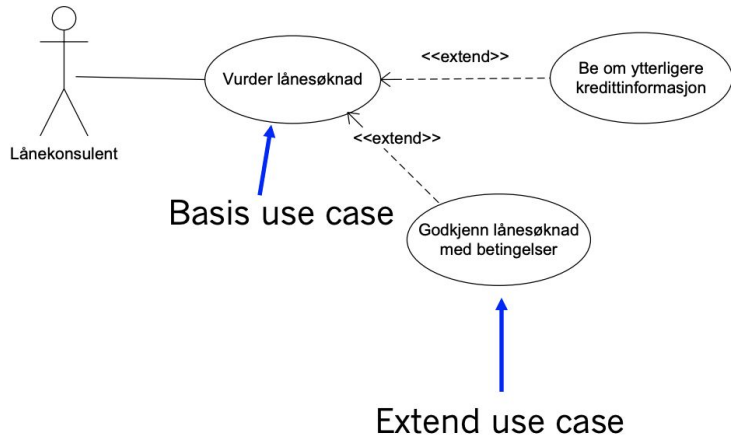
Som eier ønsker jeg å kunne se statistikk over hvor mange billetter som er solgt slik at jeg kan planlegge visninger fremover

Som kundeklubbmedlem ønsker jeg en oversikt over mine poeng for at jeg kan se om jeg har nok til å kjøpe en kinobillett

Som kundeklubbmedlem ønsker jeg å kunne tjene poeng, slik at jeg kan bruke disse til å betale for billetter

Som kunde ønsker jeg å registrere bruker for å bli medlem av kundeklubben, slik at jeg kan tjene poeng





Tekstlig beskrivelse: Hva må med?

Navn:

Primæraktør:

Sekundæraktør:

Prebetingelser:

Postbetingelser:

Hovedflyt:

Alternativ flyt:

*Tekstlige beskrivelser (især her i IN1030) følger en **fast struktur og form**.*

Denne må overholdes ved eksamensbesvarelse.

Eksempel

Navn: Registrere ny bruker via nettside

Primæraktør: Bruker

Sekundæraktør: Ingen

Prebetingelser: Ingen

Postbetingelser: Ny bruker opprettet i systemet

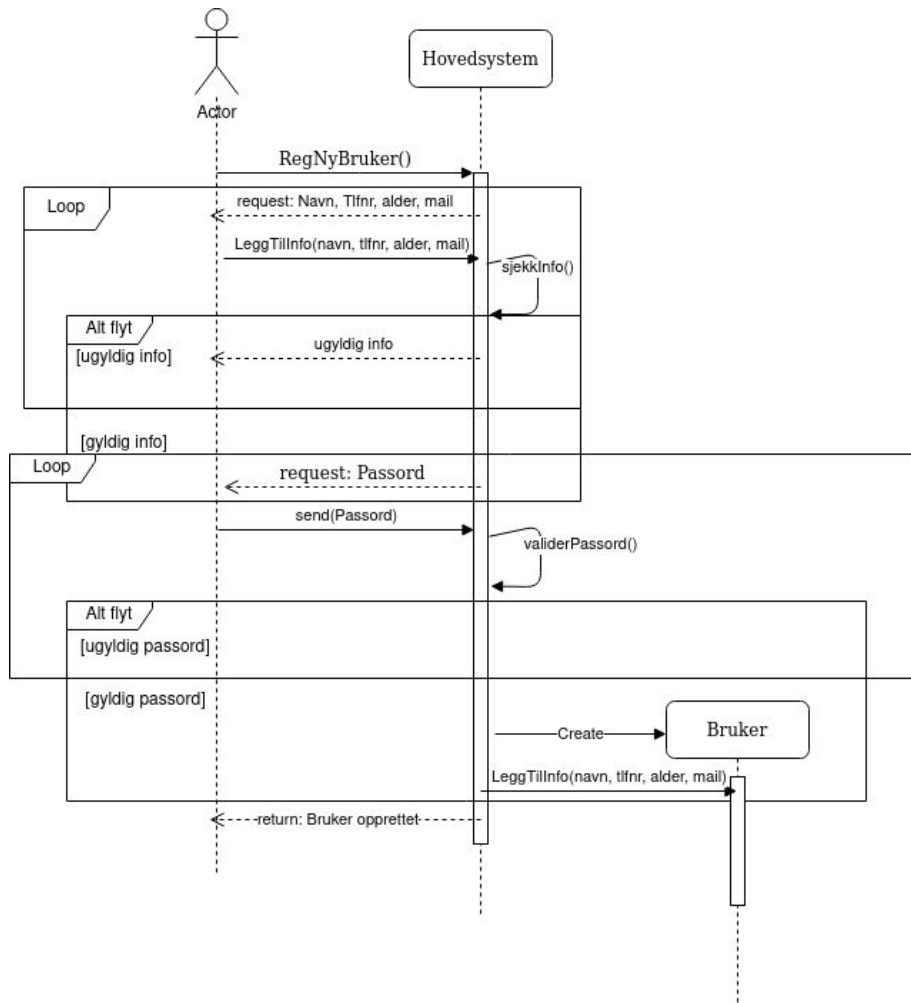
Hovedflyt:

1. Bruker velger “registrer ny bruker”
2. Systemet ber bruker om å oppgi navn, tlfnr, alder og mailadresse
3. Bruker skriver inn informasjon
4. Systemet validerer informasjonen
5. Systemet ber bruker om å opprette passord
6. Bruker skriver inn valgt passord
7. Systemet validerer passord
8. Systemet legger inn bruker i systemet
9. Systemet sender bekreftelse på skjerm og mail

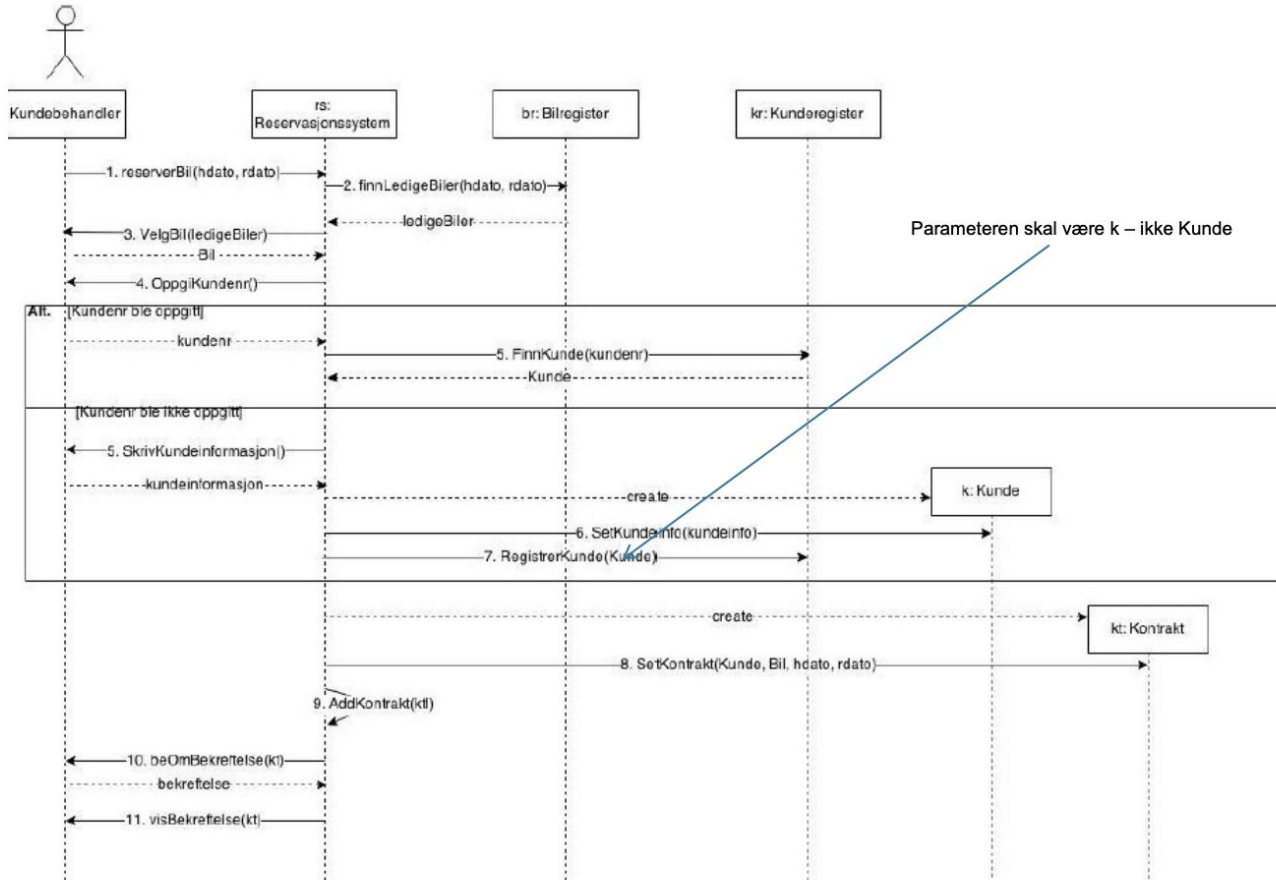
Alternativ flyt:

- 4.1. Ugyldig tlfnr, for få siffer
- 4.2. Returnerer tilbake til steg 2

- 7.1. Ugyldig passord, mangler vanskelighetsgrad
- 7.2. Systemet ber bruker taste inn et annet passord
- 7.3. Returnerer til steg 6



Sekvensdiagram - Reserver bil



EMILIE HALLGREN OG KRISTIN BRÆNDEN

Sekvensdiagram

Et **sekvensdiagram** viser sekvensene av interaksjon under et use case.

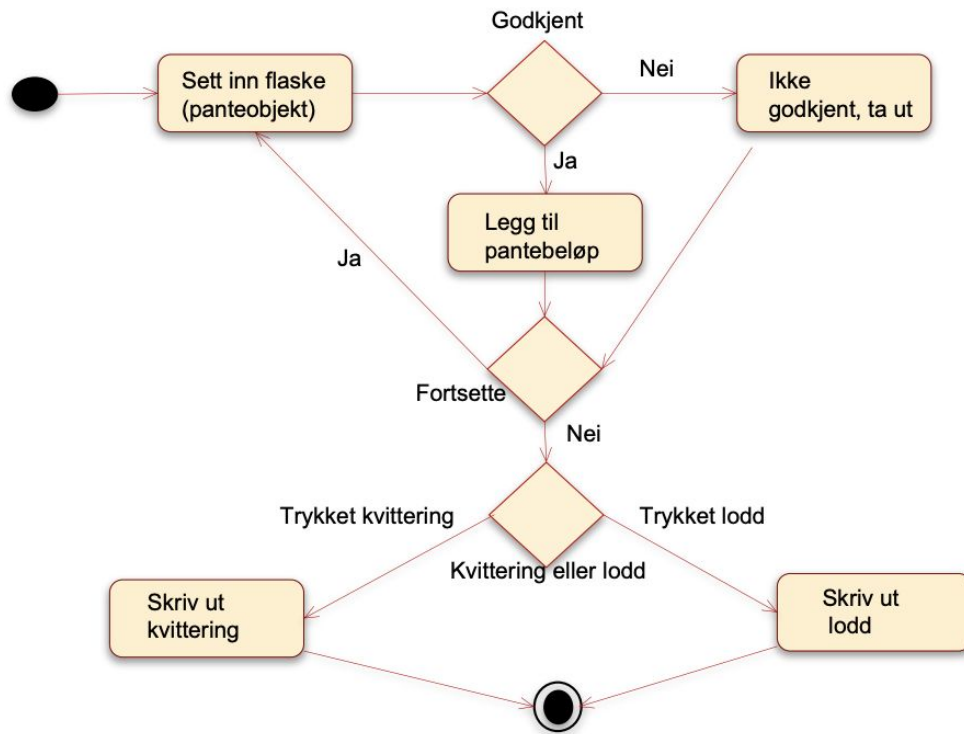
Notasjon:

- **Klasser og grensesnittet** i use caseet representeres som bokser.
Eks: Lege, Pasient og Legesystem.
- **Tidsflyten** går vertikalt nedover, slik som i sekvenstabell.
- **Aktøren** er representert som **strekfigur** på toppen av diagrammet med en stiplet linje vertikalt nedover (følger tidsflyten).
- **Interaksjon** mellom objektene er representert som **piler**.
 - Gjelder også når det er en request i systemet eller annet som ikke er reaksjon på andre beskjeder.
- **Return message** er representert som **stiplet linje**, eller beskjed tilbake fra mottakende objekt/aktor til sendende objekt/aktor.
- **Navnet på pilen** indikerer *metodekall, parameter og returverdi*.
Eks: FinnPasientInformasjon(fødselsnummer). Ønsket Pasientobjekt returneres.
- **Alternative flyt** skrives inn som **bokse** som viser de ulike alternative flyt.

Tips til modellering av sekvensdiagram (1)

1. Identifiser de ulike aktørene/objektene.
2. Lag et tenkt, tekstlig oppsett basert på hovedflyt.
3. Modeller steg for steg, basert på stegene i hovedflyten.
4. Inkluder alternativ flyt etter at du har laget en modell for hovedflyten.

Aktivitetsdiagram – pante flasker



Screen shot fra
forelesning 15/4

Aktivitetsdiagram

- Grafisk representasjon av **arbeidsflyt**
- **Aktiviteter** og tilhørende **handlinger** (actions)
- Viser overordnet **kontrollflyt**
- Beskriver hvordan mulige **utfall** av en **aktivitet påvirker flyten**
- Viser hvilke **aktiviteter** som kan utføres **parallelt**

OBS: Boken nevner aktivitetsdiagram som både Kontekstmodell og Adferdsmodell

Notasjon

Start: angir hvor flyten starter

Full sirkel

Slutt: angir hvor flyten ender

Full sirkel med ring rundt

Aktiviteter: angir ulike aktiviteter som inngår i arbeidsflyten

- Representeres med navngitte, avrundede rektangler
- Kan være fysisk (“godkjenn søknad”) eller elektronisk (“vis kjøpshistorikk”)

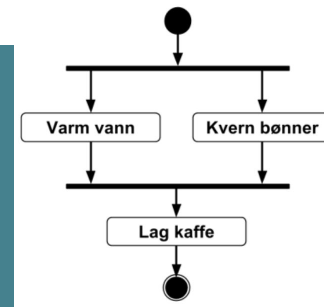
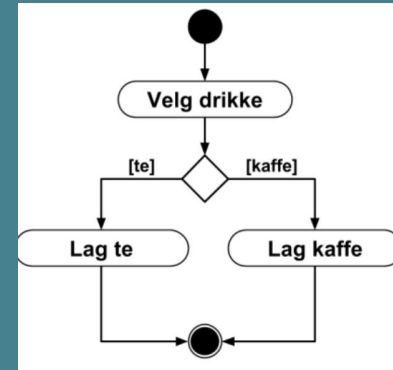
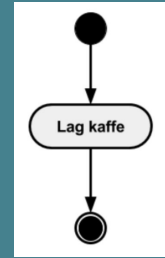
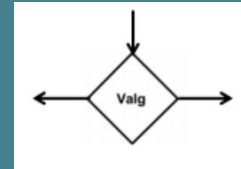
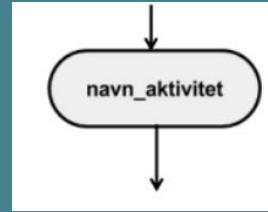
Valg: angir at man står ovenfor et valg (decision)

Eksempel: IF, IF-ELSE, CASE, osv.

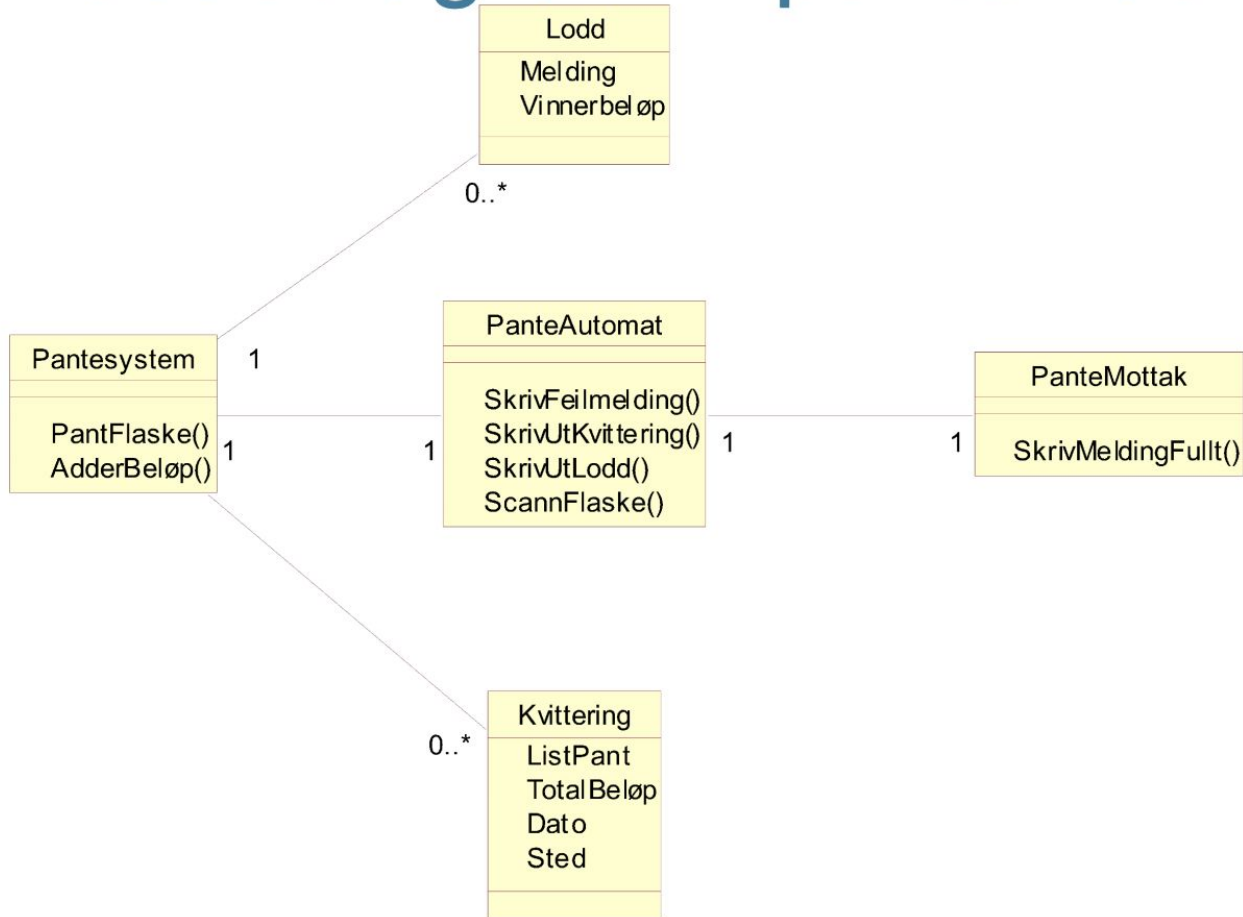
Valgdiamant

Blokkeringer (bar):

- Representerer start (split) og slutt (join) for parallelle prosesser
- Viser hvilke prosesser man må vente på, før man kan gå videre



Klassediagram – pante flasker



Klassediagram

- Strukturmodell som viser **strukturen i et system**
- Klassediagram → viser **objektklassene** i systemet og **assosiasjonene** mellom disse klassene
- Objektklasse: kan tenkes som en generell definisjon av et systemobjekt
 - Illustreres som en boks som inneholder navn, variabler og metoder.
- Assosiasjon: en link mellom klasser som indikerer at det er en relasjon mellom dem
 - Multiplisitet
 - 1 nøyaktig én
 - 0 .. 1 null eller én
 - * eller 0 .. * null eller mer
 - 1 ..* én eller mer ➤ n nøyaktig én

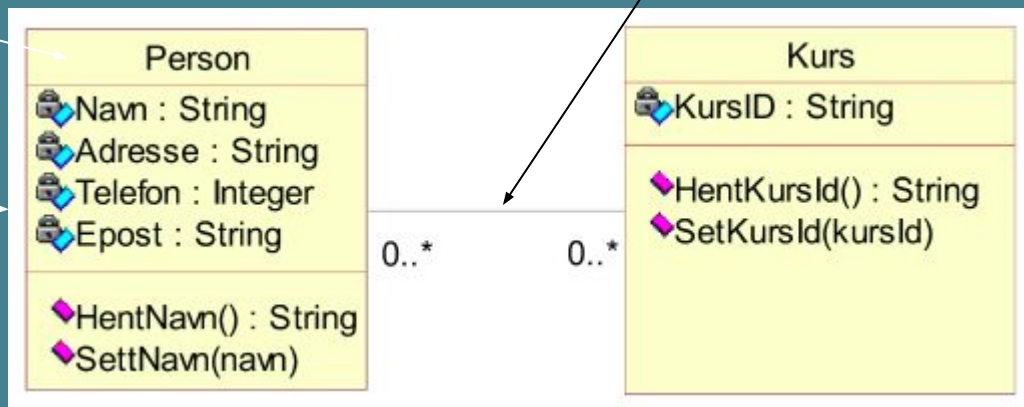
Notasjon

Navn på klasse

Attributter

Funksjoner og metoder

Relasjon mellom objekter



Informasjonssikkerhet

- **Sikkerhetsmål**
 - Konfidensialitet
 - Integritet
 - Tilgjengelighet
 - (+ Personvern)
- **Tiltak**
 - Fysiske, Tekniske og Administrative tiltak
- **Faser for tiltak**
 - Preventive, Detektive og Korrigerende
- **Centralt**
 - Verdier, trusler, sårbarheter og tiltak

Arbeidsslides fra gruppetimen 9/5

Felles repetisjon av Modul B i grupper

Dere inndeles i 3 grupper. Hver gruppe får et læringsmål.

Noter og diskuter for det givne læringsmålet:

- Hvordan ville dere oppsumere dette læringsmålet (forklart til medstudenter)?
- Hvilke stikkord og begreper er relevante + hva betyr de? **Skriv kort forklaring.**
- Hva har vi lært fra dette læringsmålet?
- Hvordan relaterer dette læringsmålet seg til hva deres ellers har lært i IN1030?

Lag et (eller fler) slide med oppsummeringen

Viktig: dere har bare tilgang til lysarkene når dere logger inn med uio-eposten (med G-suite aktivert)

Læringsmål 4: ... Kjenner du til ulike faser og aktiviteter som inngår i systemutvikling

Systemutvikling: læren om utvikling og forvaltning av IT-systemer

Inkluderer mange ulike aktiviteter, deriblant:

- planlegging
- kravinnsamling
- kravanalyse
- design
- programmering
- testing
- konfigurasjonsstyring
- versjonshåndtering

Krav?

- ulike systemer har ulike egenskaper og stiller ulike krav
For eksempel kan viktighet av sikkerhet variere, brukergrupper variere...
- innen IT håndteres kravene slik at de kan brukes i utvikling av selve systemet (utformes en kravspesifikasjon)
- enS kravspec er basis for anbud, kontrakt og design
- sørger for samsvar mellom det bedriften leverer, og det kunden trenger

*Slides laget av studentene i
gruppetimen 9/5 2022*

Begreper om krav:)

Funksjonelle krav:

- hva skal systemet gjøre
- hvilke funksjoner skal systemet tilby
- avgrense hva systemet ikke skal gjøre

Ikke-funksjonelle krav:

- hvordan skal systemet implementere de funksjonelle kravene?
- etiske krav, lovgivning, eksterne krav, produktkrav, organisasjonskrav

Kravspesifikasjon:


- dokument som beskriver kravene
- utforming varierer blant “faste” eller smidige prosjekter

Læringsmål 4: ... Kjenner du til ulike faser og aktiviteter som inngår i systemutvikling

Systemutvikling: læren om utvikling og forvaltning av IT-systemer

Utviklingen kan foregå i forskjellige prosesser (måter å jobbe på) som gir ulike resultater:

- påvirker tidsbruk
- påvirker kvalitet
- påvirker pris
- hvordan kravene oppfylles

For å lage en plan for prosessen brukes en prosessmodell! 

Hvilke ulike prosessmodeller finnes?

Vi har gjennomgått:

- Fossefall
- Kanban
- Scrum

Slides laget av studentene i gruppetimen 9/5 2022

Begreper om prosessmodeller:)

Modellene:

Fossefall

- deler prosessen i strenge faser, som må gjøres helt ferdig før man går til neste steg
- som en foss; kan kun gå én vei
- vanskelig å underveis endre kravspesifikasjon

Behov for smidighet

- planlegger for endringer
- samarbeid med kunde framfor kontraktsforhandlinger

Under smidige metoder finnes:

Scrum

Tre faser:

- 1) Planlegging
- 2) Gjennomføring
 - a) oppdelt i sprinter, der inkrementer av systemet leveres
- 3) Avslutningsfasen

Kanban

Deler også oppgavene i små biter

- forhindre flaskehals; begrense antall arbeidspakker som jobbes med samtidig
- mindre fokus på estimering av tid og konstanter

Egner seg der det er vanskelig å estimere tidsbuk; feks ved systemkrasj, vedlikehold

*Slides laget av studentene i
gruppetimen 9/5 2022*

Læringsmål 5: ... Har du forståelse for samspillet mellom systemutvikling og ulike brukere og interessegrupper

Hensikten med å utvikle/forbedre et IT-system er å løse utfordringer

- Hvilke problemer har brukere? Hvordan kan vi forbedre de?

Kravhåndtering er en prosess for å finne ulike behov/krav fra brukere som systemet utvikles for

- Identifisere kravene
- Analysere kravene
- Spesifisere kravene

Gjennom en kravspesifikasjon blir det satt basis for

- Anbud (Ulike tilbydere kan tilby ulike måter å løse brukerens behov på)
- Kontrakt (Fossefall? Smidig? Kost? Tidsrom? Fremgangsmåte?)
- Design og implementasjon

Slik oppstår det et samspill mellom systemutvikling og ulike brukere/interessegrupper

Læringsmål 6: ... Kan du anvende metoder og teknikker for kravhåndtering, utføre modellering ved hjelp av UML, og vurdere fordeler og ulemper ved forskjellige metoder og teknologier for systemutvikling

- kravhåndtering:
 - identifisere, analysere og spesifisere kravene
 - f.eks: risikoanalyse, kravspesifisering
- uml:
- abstrakte modeller av et system
 - trenger flere typer modeller for å gi forskjellige perspektiver
 - kan kreve mye ressurser/tid
 -
 -
- fordeler og ulemper ved forskjellige metoder og teknologier for systemutvikling
- fordeler:
 - oversiktig bilde over hvordan systemet fungerer
 - optimalisere systemet
 - ulemper:
 - krever tid og ressurser

Tidligere eksamensoppgaver kan finnes på semestersiden

[Link til oversikt på semestersiden over tidligere eksamensoppgaver](#)

[Link til Emmas \(gr. 2\) presentasjon med alle tidligere eksamenssett](#)

Tips til eksamen (mine)

- Les hele oppgaven igjennom - få overblikk
 - Lag eventuelt egne eksempler som kan brukes til eksamen
- Sørg for å besvare oppgaven helt (jf. formalia)
 - Ber oppgaven om X setninger → skriv X eller flere setninger
 - Vær obs om dere får diskutert eller drøftet (påstand → begrunnelse/eksempler → konklusjon)
- Øv tidligere eksamenssett

Takk for i dag!

Har du spørsmål, så send endelig en mail på: nhmoller@uio.no