

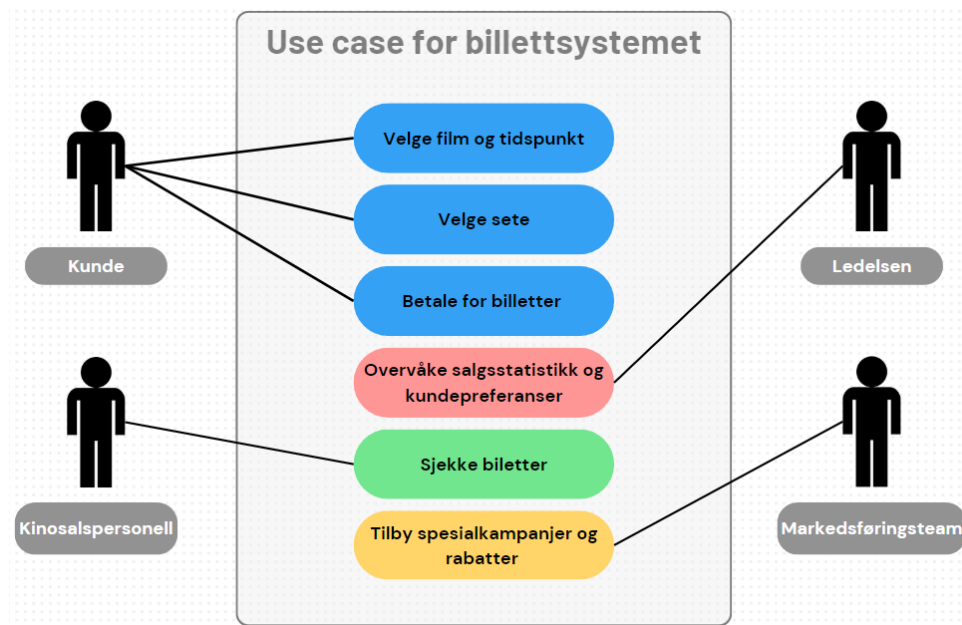
IN1030 - Systemer, krav og konsekvenser - Våren 2024

Obligatorisk oppgave 5: Modellering av krav

Del 1

Oppgave 0 - Use case for billettsystemet

a)



b)

Navn på use case: «Betale for billett»**Aktør:** Kunde**Prebetingelser:**

- Kunde har valgt en film og et tidspunkt for filmvisningen.
- Kunde har valgt hvor kunden ønsker å sitte.

Postbetingelser:

- Transaksjonen er fullført, og billetten er betalt.
- Kunde mottar bekreftelse og en digital billett.

Hovedflyt:

1. Kunden velger "Betale" etter å ha valgt filmvisning og sete.

2. Systemet viser totalprisen og betalingsmetoder.
3. Kunden velger betalingsmetode og oppgir nødvendig betalingsinformasjon.
4. Systemet validerer betalingsinformasjonen.
5. Betalingen blir behandlet.
6. Kunden mottar en bekreftelse og billett.

Alternativ flyt - Betalingsinformasjonen er ugyldig:

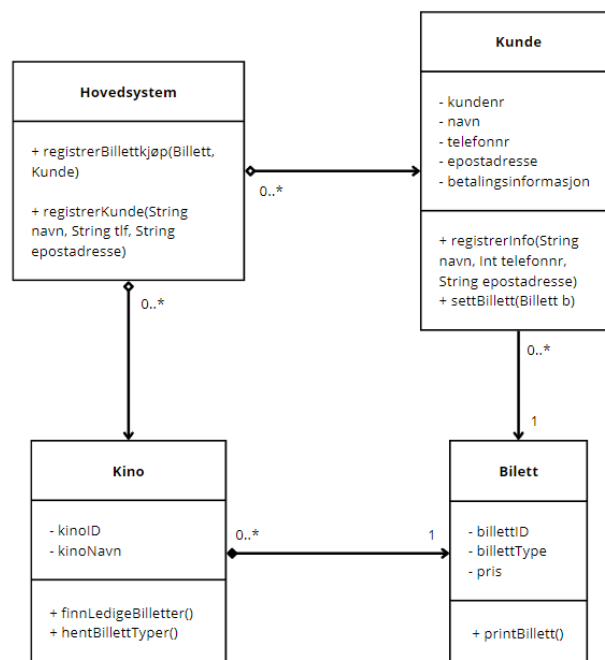
1. Systemet identifiserer at informasjonen er ugyldig.
2. Kunden informeres om feilen og blir bedt om å legge inn betalingsinformasjonen på nytt.
3. Kunden legger inn den nye betalingsinformasjonen.
4. Flyten fortsetter fra trinn 4 i hovedflyten.

Alternativ flyt - Kunde avbryter transaksjonen:

1. Kunden velger "Avbryt" i stedet for å fullføre betalingen.
2. Systemet avbryter prosessen og fører kunden tilbake til setevalg eller filmvalg, basert på kundens valg.
3. Ingen betaling blir behandlet.

Del 2

Oppgave 1 – Klassediagram



Oppgave 2 – Sekvensdiagram

a)

Aktører:

- Primær Aktør: Kunde
- Sekundære Aktører: Billettsystem

Hovedflyt:

1. Kunden logger inn med e-post og passord.
2. Systemet validerer kundens opplysninger.
3. Kunden velger filmvisning, og billetttype.
4. Kunden velger antall billetter og bekrefter valget.
5. Kunden gjennomfører betalingen.
6. Systemet genererer og sender PDF-billettene til kundens e-post.

Pre-betingelser:

- Kunden må allerede være registrert i systemet.

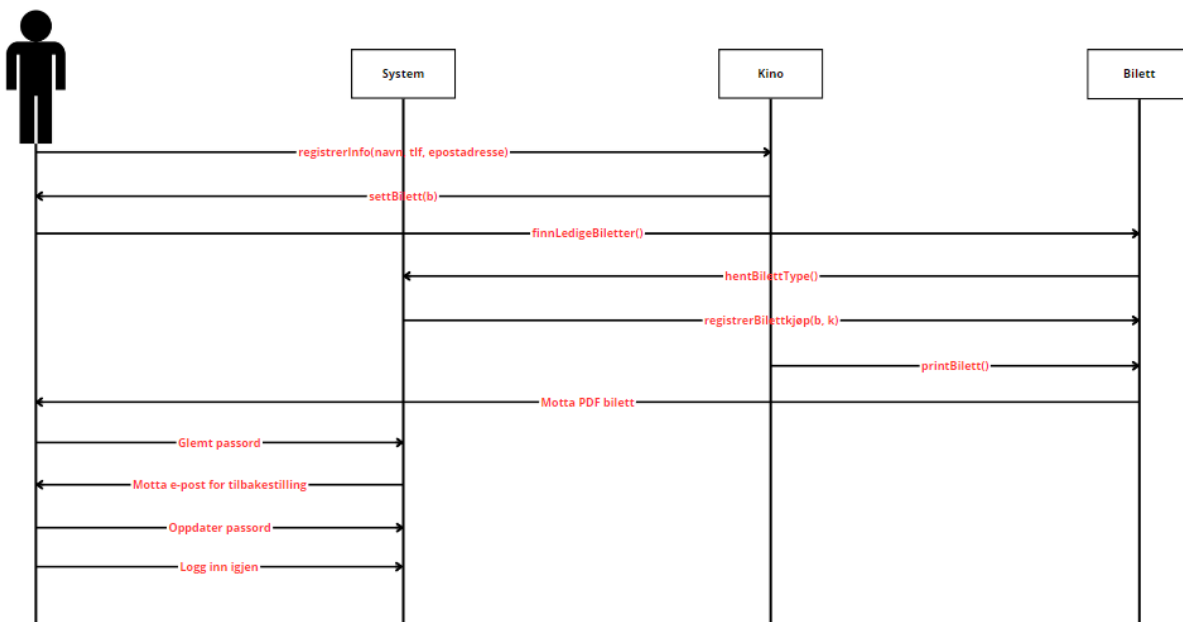
Post-betingelser:

- Kunden mottar billettene i PDF-format.

Alternativ flyt - Kunde har glemt passordet

1. Kunden velger "Glemt passord".
2. Systemet ber om kundens registrerte e-postadresse.
3. Kunden oppgir e-postadresse, og systemet sender en link for tilbakestilling av passord.
4. Kunden følger linken og setter et nytt passord.
5. Kunden logger inn med det nye passordet.
6. Flyten fortsetter fra trinn 3 i hovedflyten.

b)



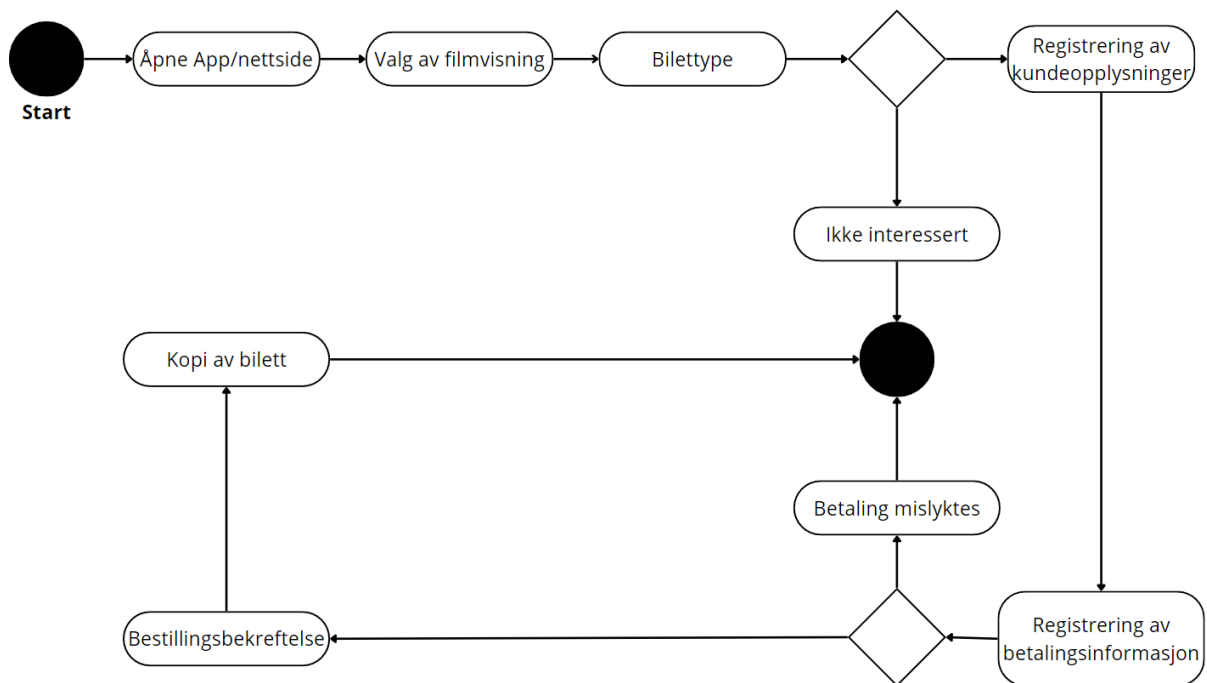
Oppgave 3 – Aktivitetsdiagram

- a) Aktivitetsdiagrammer er en del av UML og brukes for å illustrere prosesser og arbeidsflyter i et system. Hver aktivitet representeres av et rektangel, og flyten mellom aktivitetene vises med piler. Beslutninger i en prosess vises med et diamantformet symbol. Diagrammet kan også vise aktiviteter som kan utføres parallelt, noe som hjelper med å identifisere og organisere prosesser som kan kjøre samtidig uten å påvirke hverandre.

Aktivitetsdiagrammer er nyttige fordi de gir en klar visuell representasjon av rekkefølgen og vilkårene for utførelse av prosesser. Disse hjelper med å identifisere mulige problemer eller ineffektiviteter i en prosess. Diagrammer kan brukes til å dokumentere arbeidsflyter, noe som er avgjørende for opplæring og kvalitetssikring.

Et praktisk eksempel på bruken av aktivitetsdiagrammer er innen helsevesenet, hvor de kan brukes for å detaljere trinnene i behandlingen av pasienter. For eksempel kan et sykehus bruke et aktivitetsdiagram til å kartlegge hele prosessen fra en pasient blir tatt inn, gjennom ulike behandlinger, til pasienten skrives ut. Dette sikrer at alle deler av behandlingen følger et fast mønster, og hjelper med å opprettholde en høy standard på pasientomsorgen.

b)



Oppgave 4 - Smidig & DevOps

- a) Ved å jobbe i korte sykluser (sprinter) kan teamet raskt teste og få tilbakemeldinger på hva de har utviklet. Dette lar dem gjøre endringer underveis i prosessen, ikke bare på slutten av prosjektet. Dette øker kvaliteten på produktet og fører til raskere leveranser.

Ved å bygge systemet i separate deler, kan ulike team jobbe på forskjellige deler samtidig. Hver del kan utvikles, testes og settes i drift uavhengig av hverandre, noe som reduserer kompleksitet og tid brukt på å finne og fikse feil.

Når utviklere og driftsteam jobber tett sammen forbedres det kommunikasjonen og samarbeidet. Dette forbedrer også kvaliteten og forkorte tiden det tar å lansere produktet, da de tekniske utfordringer håndteres fortløpende.

Ved kontinuerlig bygging og testing av endringer som utviklerne gjør, kan feil oppdages tidlig, og store problemer med integrasjon av kode kan unngås. Dette gjør utviklingsprosessen raskere og mer effektiv.

b)

- Continuous Integration: Dette er når utviklere stadig legger inn sin nye kode i et felles prosjekt. Hver gang noen legger til kode, sjekkes denne automatisk gjennom tester for å sikre at det ikke skaper problemer. Hovedmålet er å gjøre dette ofte slik at man unngår store problemer senere.
- Continuous Delivery: Dette bygger videre på «Continuous Integration» ved at hver gang koden passerer alle tester, blir den gjort klar til å kunne brukes «live». Dette betyr at man alltid har en versjon av produktet som kan settes i drift hvis man ønsker det.
- Continuous Deployment: Dette bygger videre på «Continuous Delivery». Her går endringer som har passert alle testene automatisk ut til brukerne, uten å gjøre noe mer. Dette krever at testingen er veldig grundig siden oppdateringene går «live» uten noe ekstra sjekker. Dette kan øke risikoen for feil, men hjelper med å raskt forbedre produktet og fikse problemer.

Oppgave 5 – Risikohåndtering

Risikokategori	Risiko	Sannsynlighet	Konsekvens	Tiltak	Ansvarlig
Eksterne	Endringer i markedet	Lav	Middels	Markedsanalyser	Markedsavdelingen
Eksterne	Endring av loverket	Lav	Lav	Overvåking av lovendringer	Jurdisk avdeling
Tekniske	Feil i programvare	Middels	Høy	Iverksette automatiserte tester	Utviklingsteam
Organisatoriske	Mangel på kvalifisert personell	Lav	Middels	Rekruttering og opplæring av ansatte	HR
Organisatoriske	Interne konflikter	Høy	Høy	Kommunikasjon og jevnlig møter	Ledelse og HR
Organisatoriske	Høyt sykefravær	Middels	Høy	Fleksible arbeidsordninger, helsefremmende tiltak	HR