

INF1050

Oblig 3

Deltakere:

Aina John

Deividas Jaskenas

Rahel Gaim

Pilasilda Antony-George

Oppgave 1: Estimering

a. Foreslå egnet måte (eller egnede måter) for å måle størrelsen eller mengde funksjonalitet til systemet for leie av markasykler. Begrunn svaret og gjør nødvendige antagelser.

- Det finnes flere måter å måle størrelsen eller mengden funksjonalitet på. Størrelsen på systemet omfatter 3 viktige begreper: **lengde, funksjonalitet** og **kompleksitet**. Her vil **funksjonaliteten** kunne måles ved hjelp av den funksjonelle kravspesifikasjonen som er gitt av kunden eller slik brukeren ser dem.
- For markasykkelsystemet kan vi f.eks. også telle antall kodelinjer, men ulempen her vil være at når man estimerer på et så tidlig tidspunkt, så er det usikkert hvor mange kode linjer det kommer til å være.

b. Foreslå egnet måte (eller egnede måter) for å måle kompleksiteten til systemet for leie av markasykler. Begrunn svaret og gjør nødvendige antakelser.

- Kompleksitet handler om effektivitet og problemkompleksitet. For kunne måle kompleksiteten til systemet for leie av markasykler kan vi bruke tall som beskriver tid og antall timer, til å vurdere hvor lang tid og innsats en oppgave tar. Tiden og antall timer kan gi oss en oversikt over hvor kompleks dette systemet egentlig er.
- Hvis et system må samarbeide med flere andre systemer for å kunne fungere, så sier det mye om hvor kompleks det er. Markasykler må i dette tilfellet integreres i markasykler, dette gjør systemet kompleks. Hadde på en annen side fungert helt alene ville den ikke vært like kompleks som den er nå.
- Her kan også målinger basert på sikkerhetskrav ytelseskrav nevnes.
- Man kan også benytte seg av COCOMO- modellen som er standard for prisestimering. Vi kan ved hjelp av denne estimere kostnader og tid.

Oppgave 2: Arkitektur

Innen arkitektur bruker man ofte ulike "views" for å illustrere systemet fra ulike perspektiver.

a. I forelesningen om arkitektur ble 4+1 view-modellen nevnt. Forklar kort hva de fire ulike view-ene innebærer. (Lærebok kap. 6.2)

1. **Logisk view:** viser funksjonalitet i systemet i form av UML her inngår klassediagram og sekvensdiagram.

2. Prosessview: viser hvordan systemet er sammensatt av prosesser og må de kommuniserer på. Dette illustreres i form av UML her inngår aktivitetsdiagram.

3. Utviklingsview: systemets implementasjon fra utviklerens ståsted. Det går i dybden på hvordan ting i systemet skal implementeres. Illustreres i form av UML ved pakkediagram.

4. Fysisk view: viser maskinvaren systemet er implementert på, og hvordan programvarekomponentene er distribuert på tvers av prosesser. Dette illustreres i UML i form av deployment-diagram.

b. Hvorfor er det nyttig å benytte seg av ulike views for å beskrive arkitekturen for et system?

- Det er nyttig å benytte seg av ulike views fordi interessentene ønsker å få forskjellig informasjon om systemet.

c. Gi et eksempel på hvordan systemet for markasykler kan settes opp som er:

1. Trelags logisk arkitektur

- **Presentasjonslaget(brukergrensesnitt):** handler om hvordan informasjon skal presenteres for ulike brukere, og vises når en kunde ønsker å gjøre en overføring. Dette kan gjøres i form av layout, design og skjerm.
- **Logiske-laget:** gir en oversikt over hva slags komponenter som inngår i programvaren. Dette innebærer all koden som er implementert og hvordan dens funksjonalitet er.
- **Datalaget:** handler om hvor informasjonen lagres. Eksempel kan være at det lagres alle kunder, stasjoner og sykler.

2. Firelags fysisk arkitektur: gir en oversikt over ulike maskiner og enheter

- **Klient:** dataprogram som brukeren kan bruke. I dette tilfellet kan det være billettautomatene f.eks. hvor data vises til bruker.
- **Webserver:** et program som lagrer og utleverer data på internett. Dette videreføres så til databasen. Leverer hovedinnholdet i systemet.
- **Applikasjonsserver:** formidler serverbaserte dataprogrammer til en bruker. Kjører programmer som gir ekstra innhold.
- **Database:** skal holde på den registrerte kunden og dens informasjon. F.eks når en kunde registrerer seg skal denne informasjonen lagres i en database slik at det skal være mulig å hente den informasjonen når en kunde skal gjøre et kjøp senere. Det skal også kunne være

mulighet for hente statistikk og rapporter om f.eks. antall registrerte kunder denne måneden.

d. Hva er fordelene ved å benytte seg av lagdelt arkitektur?

- En fordel med lagdelt arkitektur er at hvis man oppdager problemer kan hvert lag erstattes uten at det påvirker hele systemet.
- Det at systemet organiserer i lag er i seg selv en fordel, fordi hvert lag er organisert fra fellestrekkene.
- Det kan gjøres endringer enkelt på hvert lag, uten at det påvirker andre lag i systemet.

e. Kan det være en ulempe å benytte seg av lagdeling? Begrunn svaret.

- Unødvendig i systemer med lav utskiftning av moduler.
- Ikke alle systemer kan bli strukturert i gruppe eller lag.
- Det kan virke tvunget til å dele lag i noen små/mindre systemer
- Hvis lagdelingen ikke er delt slik at hver del er uavhengig av hverandre er det ingen fordel med å ha lagdeling. Hvis det er tilfellet, at hvert lag ikke er uavhengig av hverandre kan et lag kommunisere med et annet lag selv om det ikke er en nærhet mellom lagene. Og da mister man også poenget med lagdeling.

f. Redegjør for de viktigste karakteristikkene for de arkitektoniske stilene:

1. **Klient-server:** Hver tjeneste leveres fra en egen server.
2. **MVC:** Model-view-controller. Dette separer presentasjon og interaksjon.
3. **SOA:** Service-oriented architecture.
4. **Pipe and filter:** basert på dataflyt. Filteret er komponenten og røret er tilkoblingen. Filteret kommuniserer med røret og filtrerer data.

Oppgave 3: Aktivitets- og tilstandsdiagram

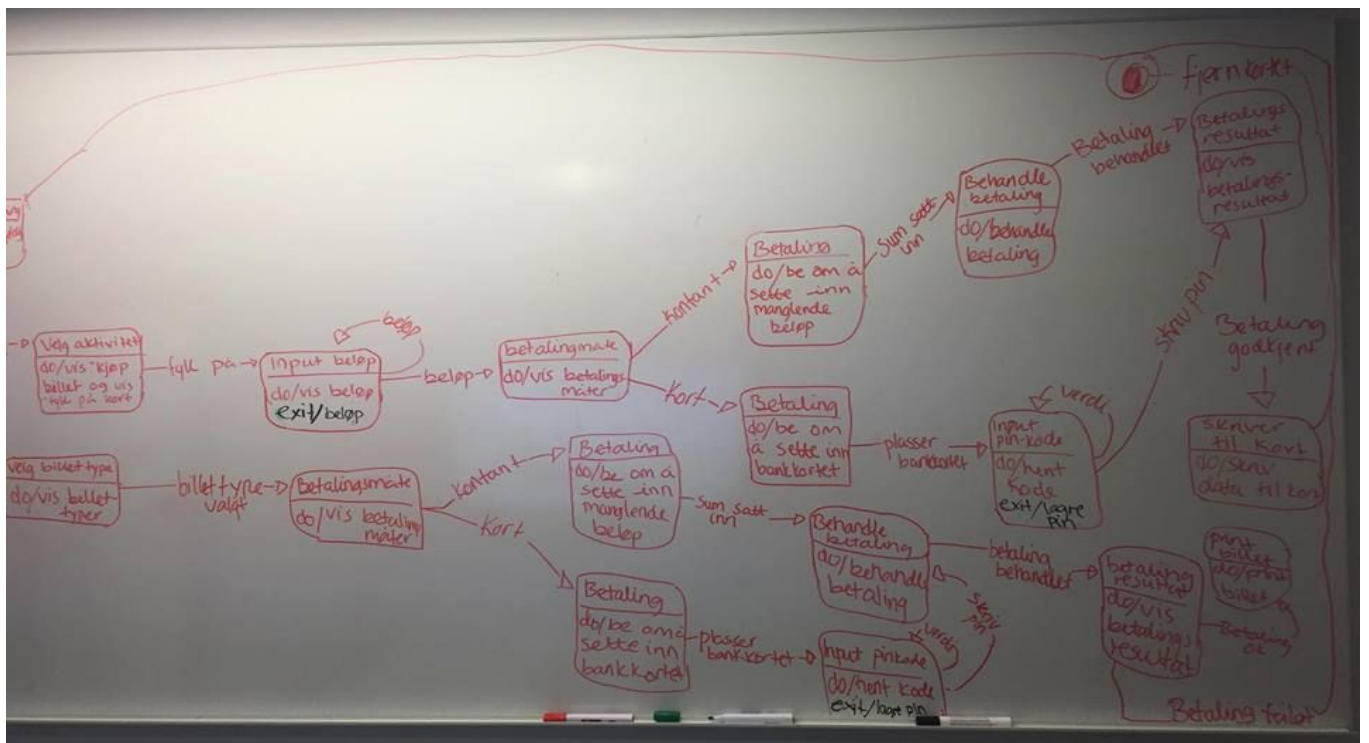
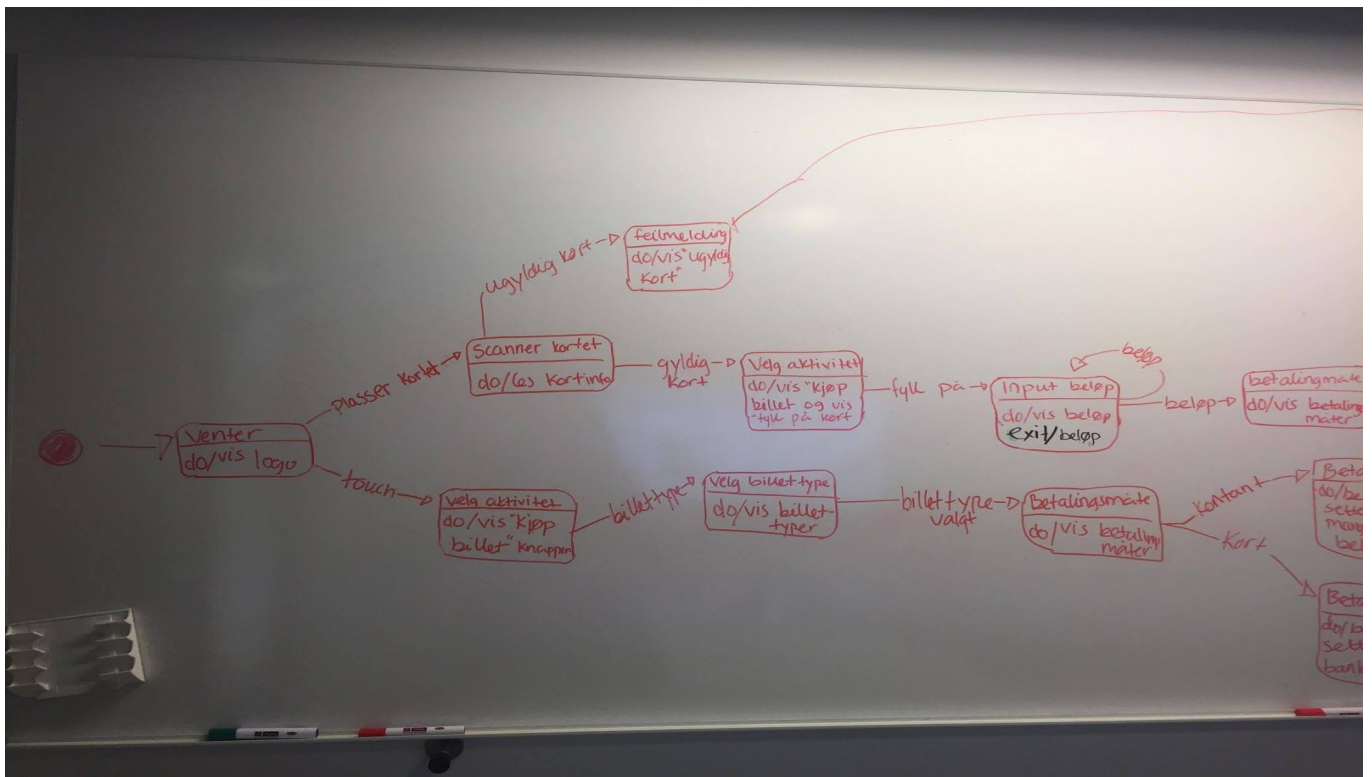
a. Hva er karakteristisk for et tilstandsdiagram, og hvorfor kan det være nyttig å benytte et slikt diagram? Begrunn svaret og kom med et eksempel på når det lønner seg.

- Et tilstandsdiagrams karakteristikk består av rektangler med avrundede hjørner. Det viser også til tilstandene til ett enkelt objekt. Rektangelet består igjen av et navn som forklarer hva slags tilstand det er. Fordelene med tilstandsdiagram er at den beskriver et objekts oppførsel og viser tilstander systemet kan være i. Den viser også hvordan vi beveger oss mellom tilstander. Som eksempel på dette så lønner det seg å bruke et slikt diagram fordi da har man kontroll over hvilken oppførsel objektene får når de påvirkes av input.

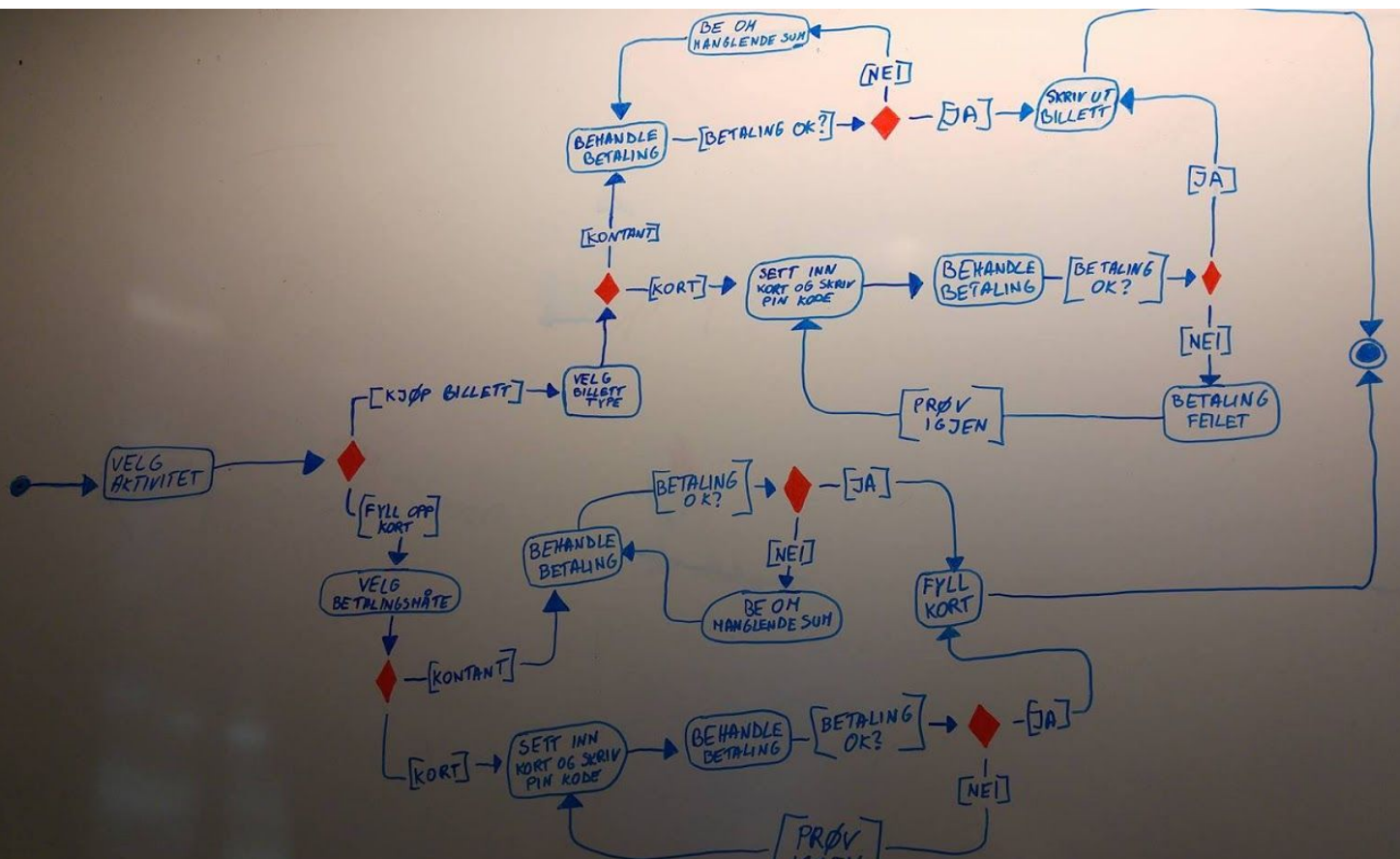
b. Hva er karakteristisk for et aktivitetsdiagram, og hvorfor kan det være nyttig å benytte et slikt diagram? Begrunn svaret og kom med et eksempel på når det lønner seg. Systemet for markasykler skal som nevnt tidligere benytte seg av Ruters betalingsautomater. Her har man mulighet til å fylle opp et reisekort med penger eller en billett, og man kan også kjøpe enkeltbilletter uten reisekort.

- Et aktivitetsdiagram viser aktiviteter i et system og flyten mellom aktivitetene. Den har en start hvor flyten starter med en sirkel og en pil, og en slutt som viser hvor flyten ender med en pil og en sirkel. Består av rektangler med avrundede hjørner. Diagrammene kan ha valg diamanter som viser alternative løp. Det lønner seg å bruke aktivitetsdiagram fordi den viser prosesser som kan kjøres parallelt.

c. Modellér et tilstandsdiagram for betalingsautomaten.



d. Modellér et aktivitetsdiagram for betalingsautomaten.



Oppgave 4: Testing

a. Forklar hva de ulike testfasene innebærer, og få med hva som skiller dem fra hverandre: - Enhetstesting - Integrasjonstesting - Systemtesting -

Akseptansetesting

- **Enhetstesting:** det er en prosess for å teste isolert til de individuelle komponenter
- **Integrasjonstesting:** det er testing som handler om å teste moduler i en gruppe.
- **Systemtesting:** prosess der systemet testes som en helhet. Her er fokuset å teste interaksjoner mellom komponenter.
- **Akseptansetesting:** evaluere systemet som leveres.
 - Det som skiller disse testfasene er at i enhetstesting testes hver og enkelt del av system individuelt. Ved integrasjonstesting testes de delene som hører sammen i grupper. Og ved systemtesting testes hele systemet samlet. Og ved akseptansetesting evalueres systemet.

b. Gi forslag på hvilke deler av systemet for markasykler som kan testes i hver av de ulike testfasene nevnt over. Sørg for at dere får med minst ett eksempel for hver av dem.

- **Enhetstesting:**
 - Vi kan ved enhetstesting teste enkle funksjoner ved systemet, for å sjekke om komponentene i systemet fungerer individuelt. Her kan vi f.eks. teste om logginn prosessen fungerer som vi ønsker. Altså at når kunde logger seg inn så må kunden kunne klare å komme til sin konto.
- **Integrasjonstesting:**
 - kan sjekke om to separate deler av systemet fungerer sammen som man ønsker. Her er poenget at man tester samspillet mellom klasser og metoder. on
- **Systemtesting:**
 - Her testes systemet som helhet. Vi kan i denne testingen sjekke om automaten og appen til Ruter er i overensstemmelse med systemet som utvikles.
- **Akseptansetesting:**

- Her ønsker vi å teste om brukernes behov, de funksjonelle kravene og forretningsplanen er i oppfylt. Ved akseptansetesting tester vi med pålitelig data som f.eks når man skal registrere en kunde så bruker vi en "ekte kunde". Oslo kommune kan f.eks. gjøre en test på om de kan logge seg inn med eget passord. Og om de kan hente rapporter og statistikker. Her tester de da brukervennligheten til systemet med det samme.