

Oppgave 1: Multiple choice (20 %)

For alle oppgavene gjelder at det bare er ett riktig svar.

No	Spørsmål	Svar A	Svar B	Svar C	Svar D
1	Kanban er et eksempel på:	Prosess	Software prosess	Prosess modell	Utviklingsprosess
2	Hvem er ansvarlig for å prioritere produktkøen (product backlog)?	Scrum master	Produkteier	Teamleder	Prosjekteier
3	For at "daily scrum" møtene skal være korte anbefales det at...	... det ikke snakkes om oppgaver under møtet.	... Ingen får snakke mer enn ett minutt.	... ikke brukes elektroniske verktøy under møtet.	... alle står under møtet.
4	Hvilken av følgende er ikke en vanlig rolle i Scrum?	Produkteier	Produktleder	Team	Scrum master
5	Hvilket av følgende krav til en billettautomat er funksjonelt?	Utskrift av billett skal ikke ta mer enn 5 sekunder.	Teksten på skjermen skal være svart med hvit bakgrunn.	Det skal være et valg for utskrift av kvittering.	Koden skal være enkel å endre.
6	Visualisering av progresjonen i arbeidsoppgaver er viktig i...	både Scrum og Kanban.	Scrum, ikke Kanban.	Kanban, ikke Scrum.	verken i Scrum eller Kanban.
7	Verifisering i testing av programvare betyr at komponenter og delsystemer fungerer sammen.	... produktet bygges på riktig måte.	... systemet gjør det brukeren ønsker.	... systemet er robust.
8	Hva betyr "time boxed" innen smidig metodikk?	Rask	Fast bruk av tid	Ofte	Fleksibel bruk av tid
9	Hva er branching?	Oppretting av en ny codeline fra en versjon i en eksisterende codeline	Oppretting av en ny baseline fra en versjon i en eksisterende baseline	Fletting av ulike versjoner av codelines	En spesiell release til en kunde

Oppgave 2 – Spørsmål fra pensum (25 %)

Besvar:

- List og beskriv de fire mest populære smidige møtene.
(se slides fra Viktorias forelesning)
- Hva er versjonshåndtering? Hvorfor er versjonshåndtering viktig? Forklar forskjellen på sentralisert og distribuert versjonshåndtering.
(se slides fra Forelesning om Git – siste spm. er egentlig fra forelesninger i tidligere kurs (IN1050/1030) – GIT er eksempel på distribuert)
- Forklar begrepene "waste" og "JIT" i Lean systemutvikling og gi eksempler
Se forelesning om teamarbeid
- Beskriv begrepet teknisk gjeld. Gi de viktigste grunnene til at man opparbeider seg teknisk gjeld, og beskriv måter man kan unngå teknisk gjeld på.

Se forelesning om teknisk gjeld og arkitektur (technical dept), står også mye om teknisk gjeld i Scrum boka.

5. Hva er generalisering i UML-modellering? Hva er nytten av generalisering i programmering? Gi gjerne eksempler.

Generalisering i UML er en måte å håndtere kompleksitet. Gjøres når det er felles egenskaper i ulike klasser, for eksempel Person og Student, der Person er en generalisering av Student. Alle egenskapene i Person blir arvet i Student, i tillegg legges det på nye egenskaper ved Student. Syntaksen i UML er en pil mellom klassene med et åpent pilhode til den «generelle» klassen. I eksemplet ville pilen gå fra Student til Person. Generalisering i objekt orienterte språk er implementert ved arv (inheritance), Student blir da en sub-klasse av Person og arver automatisk alle egenskapene til Person.

Oppgave 3: Modellering (30 %)

Selskapet Bestillgodmat tilbyr tjenester for bestilling og frakt av mat i de største byene i Norge. Stort sett alle ”take away” restauranter er med i tillegg til en del andre restauranter. Utkjøring skjer enten med sykkel eller bil. Din oppgave er å modellere deler av et system for en mobilapp som gjør brukere i stand til å bestille mat og få maten levert på angitt bestemmelsessted. For å bestille mat via Bestillgodmat må brukeren på forhånd være registrert og ha installert mobilappen.

Brukeren har en unik brukerID og et kredittkort som automatisk blir belastet når maten er levert. Tilbydere av personer som kjører ut maten (sjåfører) har tilsvarende installert en sjåførapp som gir melding når noen bestiller turer i samme sone som sjåførene befinner seg i.

Ved bestilling, må brukeren først angi adresse for hvor maten skal leveres. Brukeren får da opp en liste eller et kart med alle tilgjengelige restauranter og tiden det tar fra bestilling til levering. For enkelthetens skyld så er i dette tilfelle kun mulig å bestille mat som blir levert så raskt som mulig. Avhengig av hvilken restaurant du bestiller fra tar det enten maks 30 minutter fra du bestiller til du får maten, eller maks 1 time fra du bestiller til du får maten.

Du skal nå ta for deg use-caset ”Bestill og få tilkjørt mat”. Du kan anta at appen virker og at bestilling lar seg gjøre. Hovedflyt er at brukeren bestiller og får tilkjørt mat. Når maten er levert, blir kredittkortet belastet. I denne oppgaven finnes to alternative flyt: Det ene er at maten du prøver å bestille på restauranten ikke er tilgjengelig. Den andre er at det ikke finnes ledige sjåfører på det aktuelle tidspunktet.

For use-caset ”Bestill og få tilkjørt mat” utfør følgende:

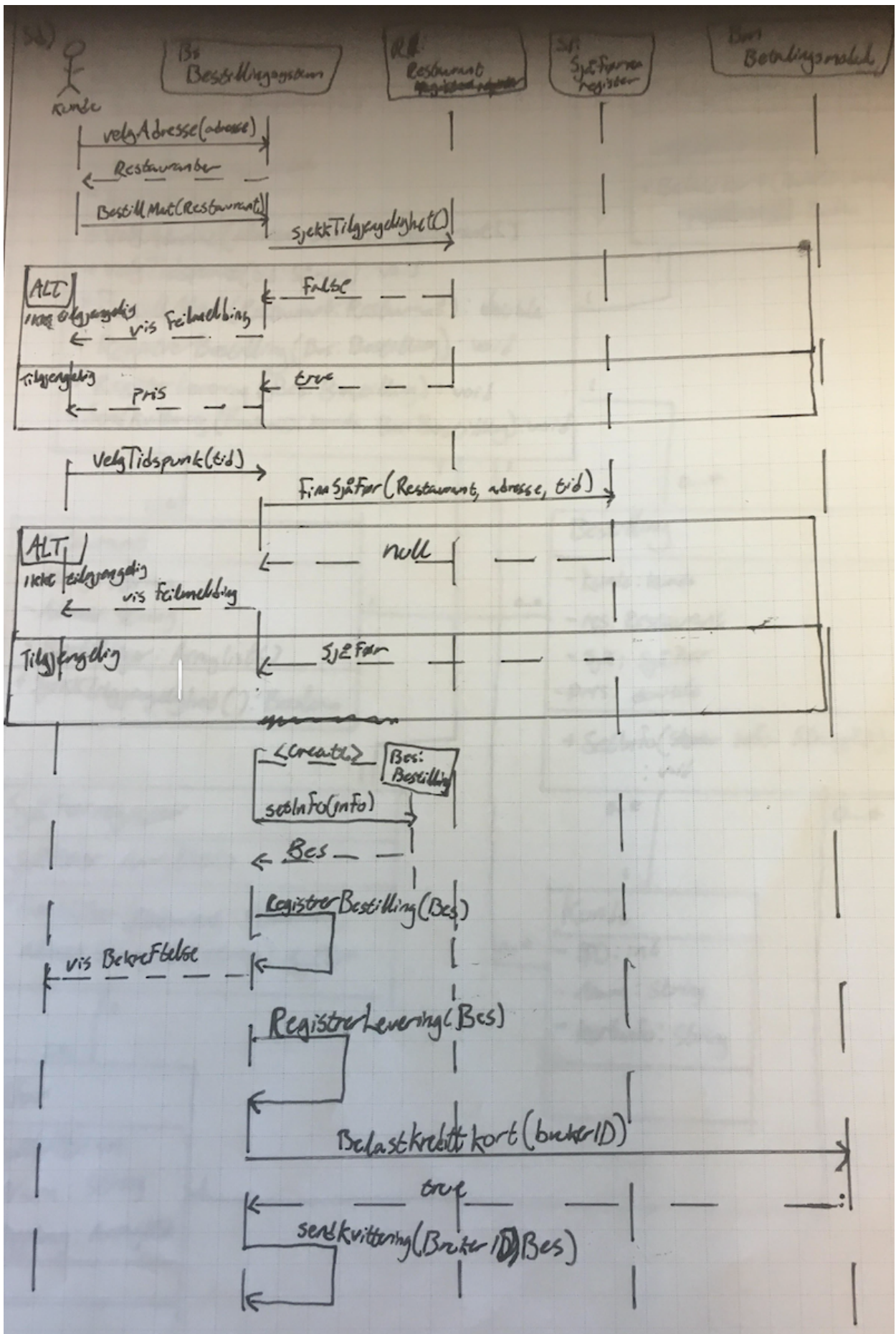
Gi en tekstlig beskrivelse.

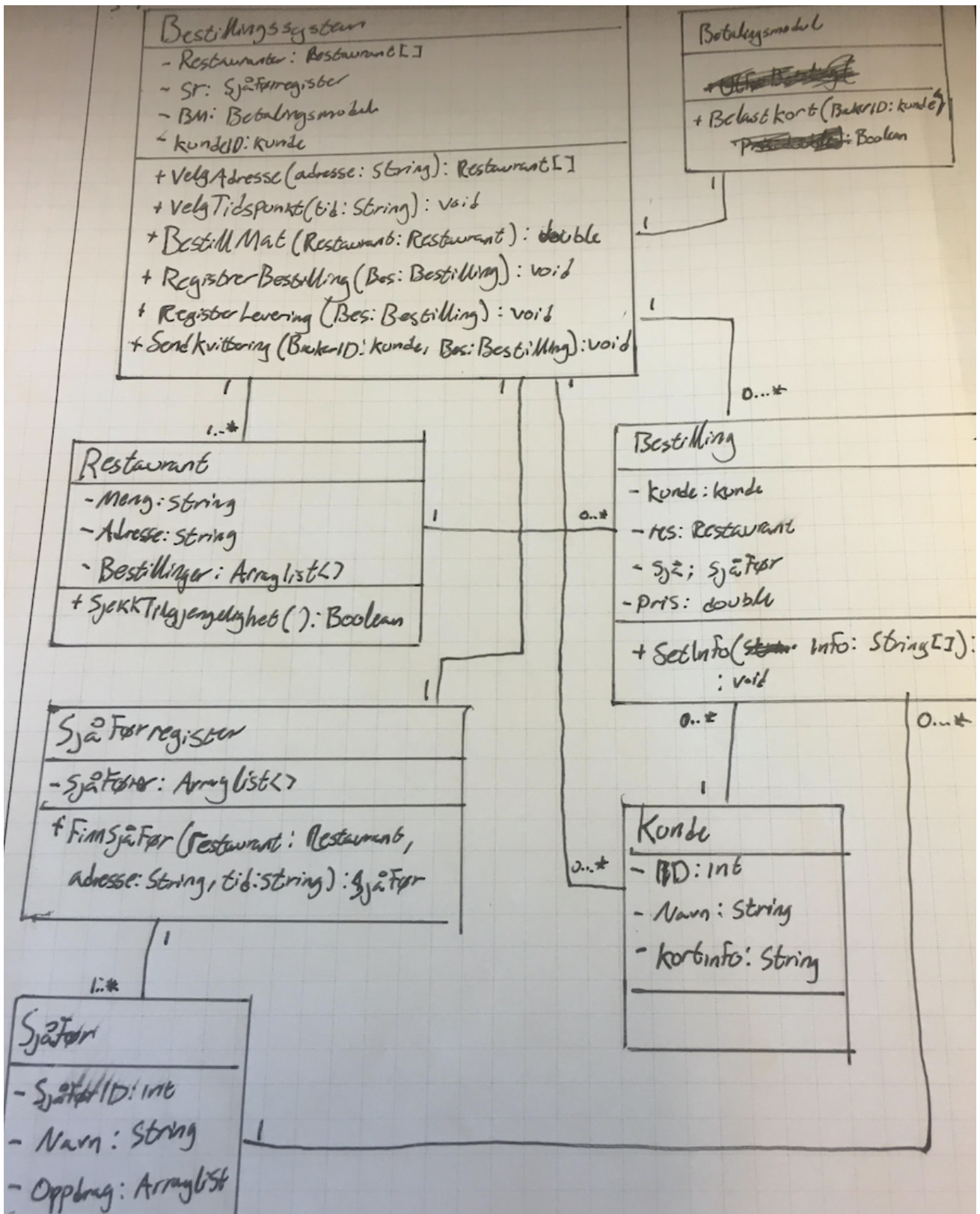
Se eksempel fra forelesning. Ganske rett fram

Tegn et sekvensdiagram. Du har følgende metoder tilgjengelig: Metoden *BestillMat (Restaurant): pris* som returnerer prisen på maten hvis bestillingen lar seg gjennomføre og metoden *FinnSjåfør (Restaurant, Bestemmelsessted, Leveringstidspunkt): sjåfør* som returnerer en sjåfør hvis det finnes ledige sjåfører. Du kan også anta at belastning av kredittkort går bra etter at turen er gjennomført, så det holder at du kaller en metode som heter *BelastKredittkort (brukerID)* som belaster kortet som brukeren har lagt inn i mobilappen.

Tegn et klassediagram som reflekterer sekvensdiagrammet du tegnet i oppgave (b). Inkluder attributter, metoder og assosiasjoner med multiplisitet.

Løsningsforslag på sekvens og klassediagram er løst av en student – dette vill gitt meget bra uttelling på disse oppgavene – her er det ikke mulig å få det til perfekt uansett.





Oppgave 4: Spørsmål knyttet til prosjektet (25 %)

Se også på slides for disse spørsmålene

Det kanskje mest viktige i denne oppgaven (og kanskje enda mer ift. spørsmålene dere får på eksamen) er at dere bruker eksempler fra deres eget prosjekt, dvs. det er viktig å ha en oversikt over prosjektet, både produktmessig og prosessmessig.

1. Hva er Recycler View i Android og forklar hvordan dette fungerer?
2. Hva er en scrummasters og produkteiers rolle og hovedoppgaver? Forklar hvordan dere brukte disse rollene i deres eget prosjekt?
3. Nevn minst tre faktorer som ofte bidrar til god effektivitet i teamarbeid, og gi eksempler på hver av disse faktorene? Beskriv hva dere selv gjorde i prosjektet for å gjøre teamarbeidet mest mulig effektivt.
4. Tenk deg at du skal videreutvikle appen dere utviklet i prosjektet og gjøre den klar for markedet. Hvilke trusselscenarioer ville du tatt hensyn til? Modellér minst ett misuse case sentralt for din applikasjon.