

Opgave 1 (20%)

Denne opgave består af 4 spørgsmål. Hvert spørgsmål udgør 5% af den samlede eksamensopgave

- a. Forklar hvordan et sprint planlægges i den agile styringsproces Scrum?

Sprints planlægges ved sprint planning, en Scrum ceremoni med deltagelse af Product Owner, Scrum Master og Team Members. Fra Product Backloggen udvælges det højest prioriterede Backlog Items. Dette gennemgås kort og hvert Team Member estimerer opgaven i at færdiggøre det pågældende emne. Ved stor spredning på estimater vil Team Members med lavest og højeste estimat redegøre for baggrunden for deres estimater. Opgavens indhold diskuteres igen kort og der genestimeres. Dette kan eventuelt gentages flere gange. Item'et tilføjes herefter Sprint Backloggen. Proceduren gentages indtil sprintet er fyldt op, hvorefter Teamet kommitter sig til sprintet.

- b. Hvad er samhørighed (cohesion) og kobling (coupling) når man snakker om gode design principper?

Design principperne samhørighed "cohesion" og kobling "coupling" er vigtige for et godt design. Der skal være lav kobling mellem klasser/komponenterne i designet og stor samhørighed i de enkelte klasser/komponenter. Den lave kobling gør at de enkelte komponenter er lettet at vedligeholde, udskifte, modificere og teste. Samhørigheden internt i klassen/komponenten kan omhandle emner som: funktionsmæssig sammenhæng, sammenhæng mellem sekvens af operationer og kommunikations sammenhæng.

- c. Hvad er en risiko matrix og hvordan kan den bruges i en risikoanalyse?

En risiko matrix består af en matrix af hændelser, som kan have en negativ konsekvens for projektet. For hver hændelse angives sandsynligheden for udfald af hændelsen. Risiko er produktet af sandsynligheden og konsekvens. En risiko matrix kan bruges til at give overblik over de hændelser som er mest kritisk for et projekt. Der udarbejdes i projektet en handlingsplan for de hændelser, hvor der både er stor konsekvens og stor sandsynlighed = høj risiko.

- d. Beskriv kort de tre type af klasser, der indgår i en applikationsmodel, og hvordan de relaterer til Use Case diagrammet?

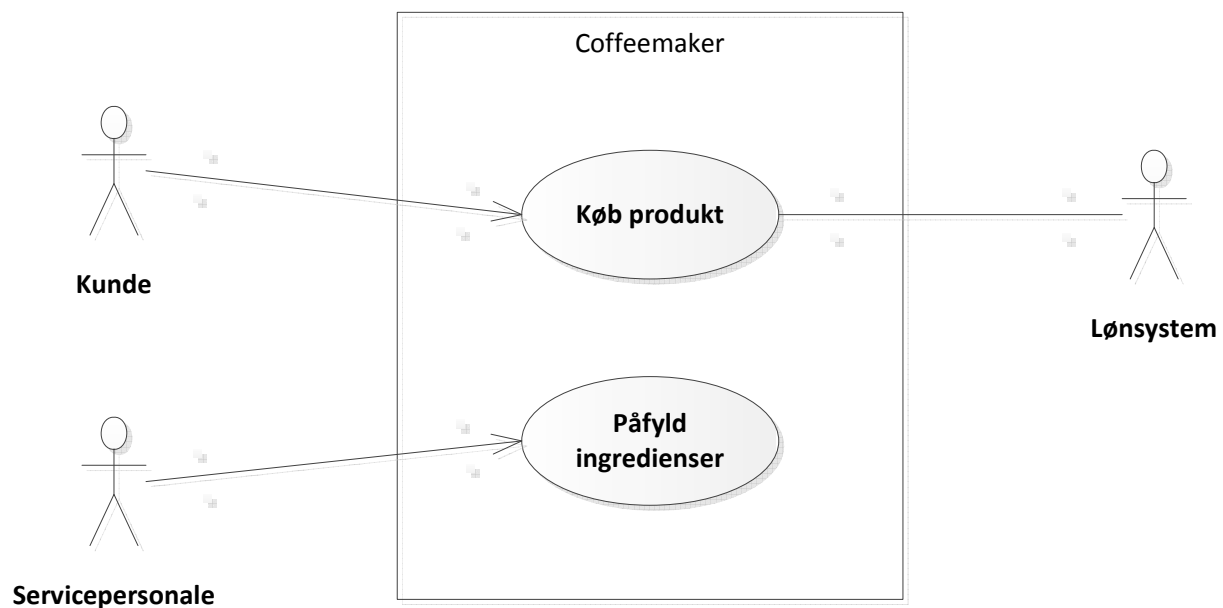
I applikationsmodellen indgår domæne- (domain), grænseflade- (boundary) og kontrol- (control) klasser.

- Domæne klasserne findes ud fra domænemodellen som er afledt af Use Case beskrivelsen og ikke diagrammet.*
- Kontrolklasserne repræsenterer funktionalitet beskrevet i Use Case diagrammet. For hver Use Case oprettes en tilsvarende kontrolklasse, med samme navn som Use Casen.*
- For hver aktør i Use Case diagrammet oprettes en grænsefladeklasse.*

Målet er en lagdelt arkitektur, hvor funktionalitet, domæne og grænseflader blive adskilt.

Opgave 2 (40 %):

Opgave 2A:



Figur 1 Use Case diagram for kaffeautomat

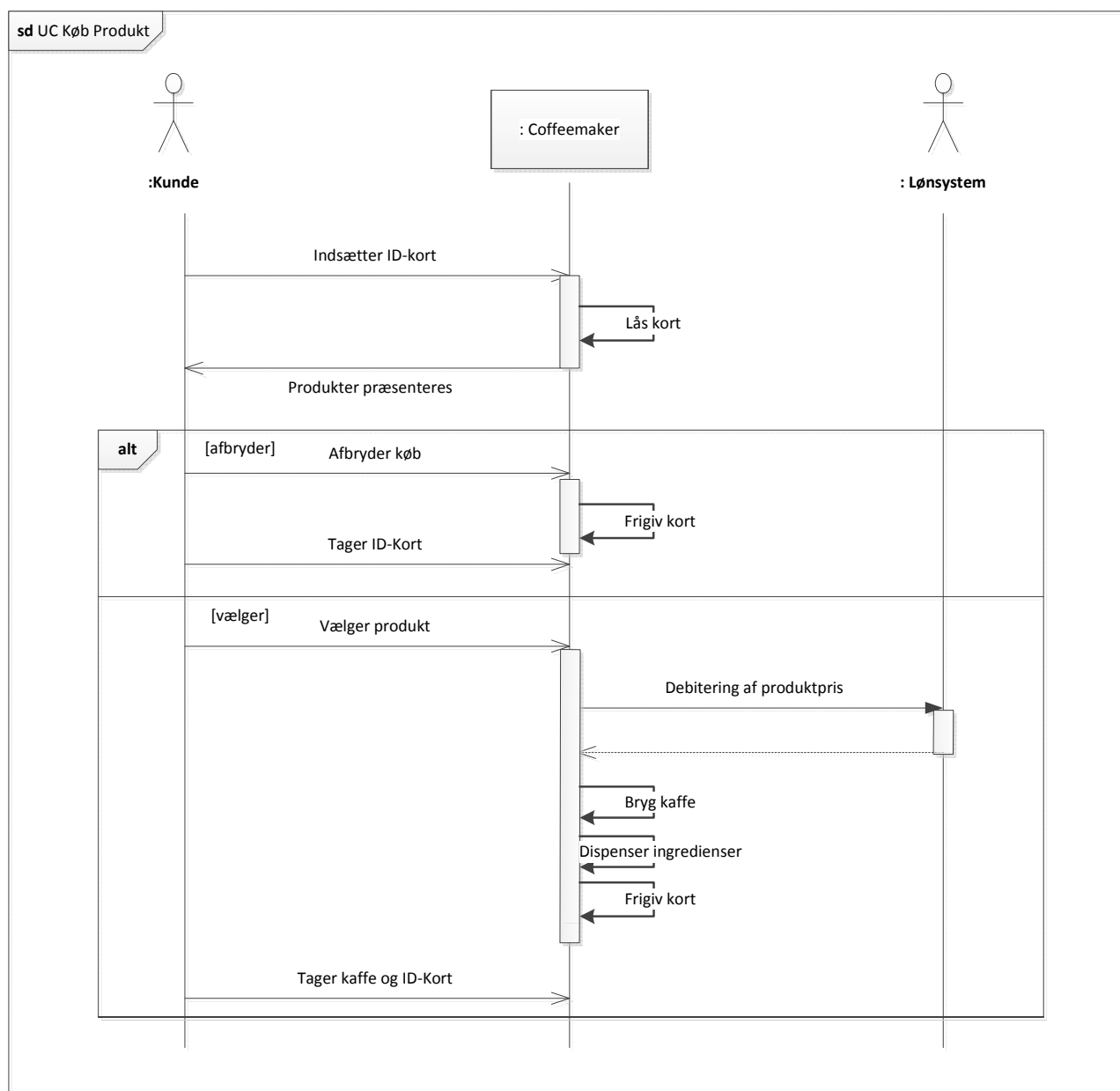
Opgave 2B:

Navn:	Køb Produkt
Mål	Kunde får serveret den type kaffe som vælges på kaffeautomaten. Beløbet svarende til produktpris er debiteret i Lønssystem .
Initiering	Kunden initierer use casen
Aktører	Primær: <i>Kunde</i> Sekundær: <i>Lønssystem</i>
Antal samtidige forekomster	En
Prækondition	Coffeemaker er påfyldt kaffepulver, mælk og sukker.
Postkondition	Kunde har fået serveret den valgte type kaffe. Beløb svarende til produktpris er debiteret på Kundes lønkonto i Lønssystem.
Hovedscenarie	<ol style="list-style-type: none"> 1. Kunde indsætter sit ID-kort i kortlæseren 2. System låser kortet fast 3. System præsenterer valgbare produkter: P1 – Kaffe P2 – Kaffe med sukker P3 – Kaffe med mælk P4 – Kaffe med sukker og mælk 4. Kunde vælger produkttype [Extension 1: Kunde afbryder køb] 5. System debiterer beløb svarende til produktpris i Lønssystem 6. System fremstiller det ønskede produkt 7. System frigiver ID-kort efter 10 sekunder 8. Kunde tager sit ID-kort og use casen afsluttes
Udvidelser/undtagelser	[Extension 1: Kunden afbryder sit køb] <ol style="list-style-type: none"> 1. ID-kort frigives 2. Kunde tager sit ID-kort og use casen afsluttes

Opgave 2C:

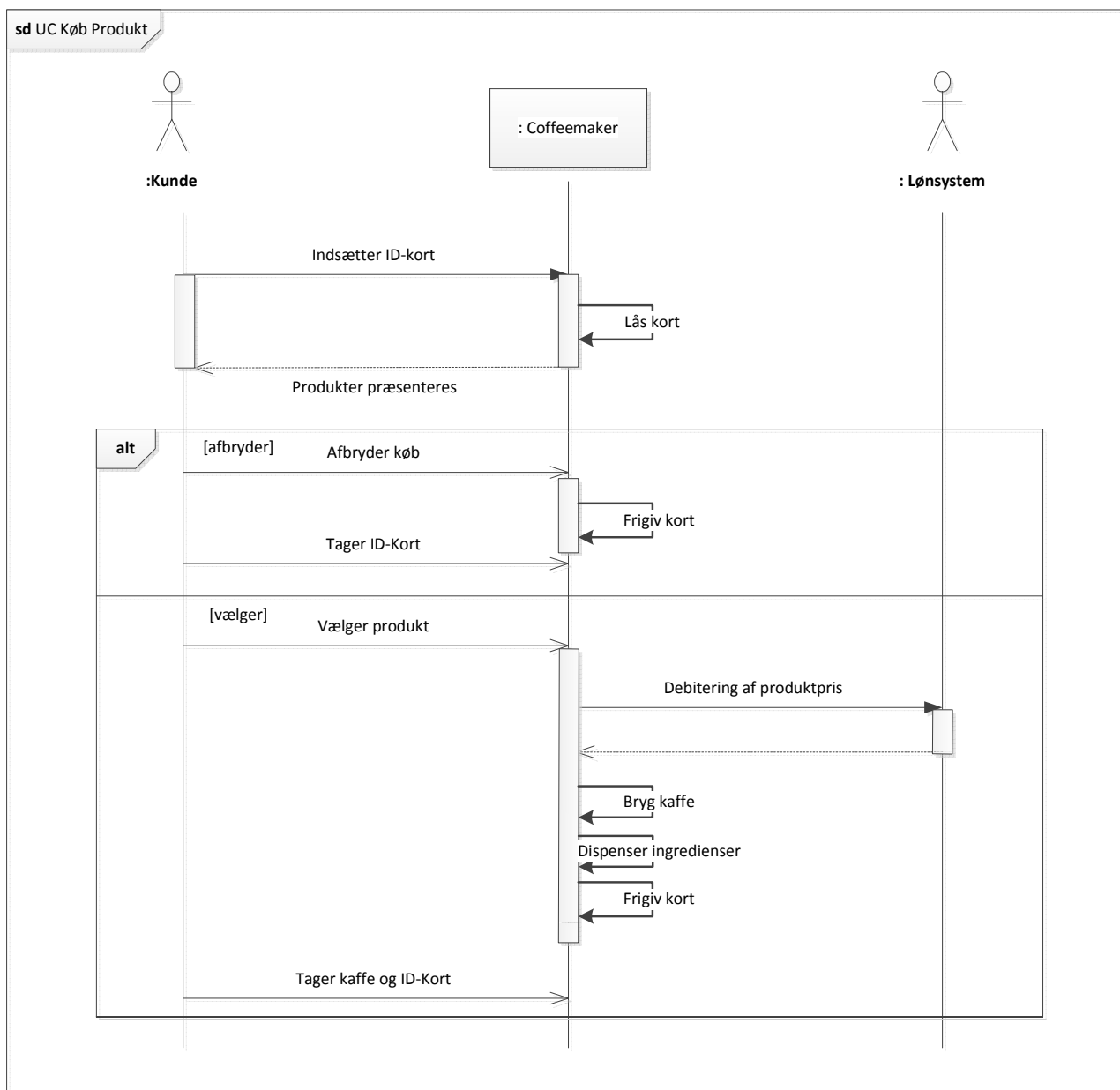
Nedenfor er skitseret 2 forskellige forslag til løsningen med henholdsvis asynkron og synkron informationsudveksling mellem kunden og kaffeautomaten.

I Figur 2 er valgt asynkron kommunikation mellem kunde og kaffeautomaten. Her er betragtning, at kunden i teorien godt kan gøre andre ting i mens han køber sin kaffe og dermed asynkron kommunikation.

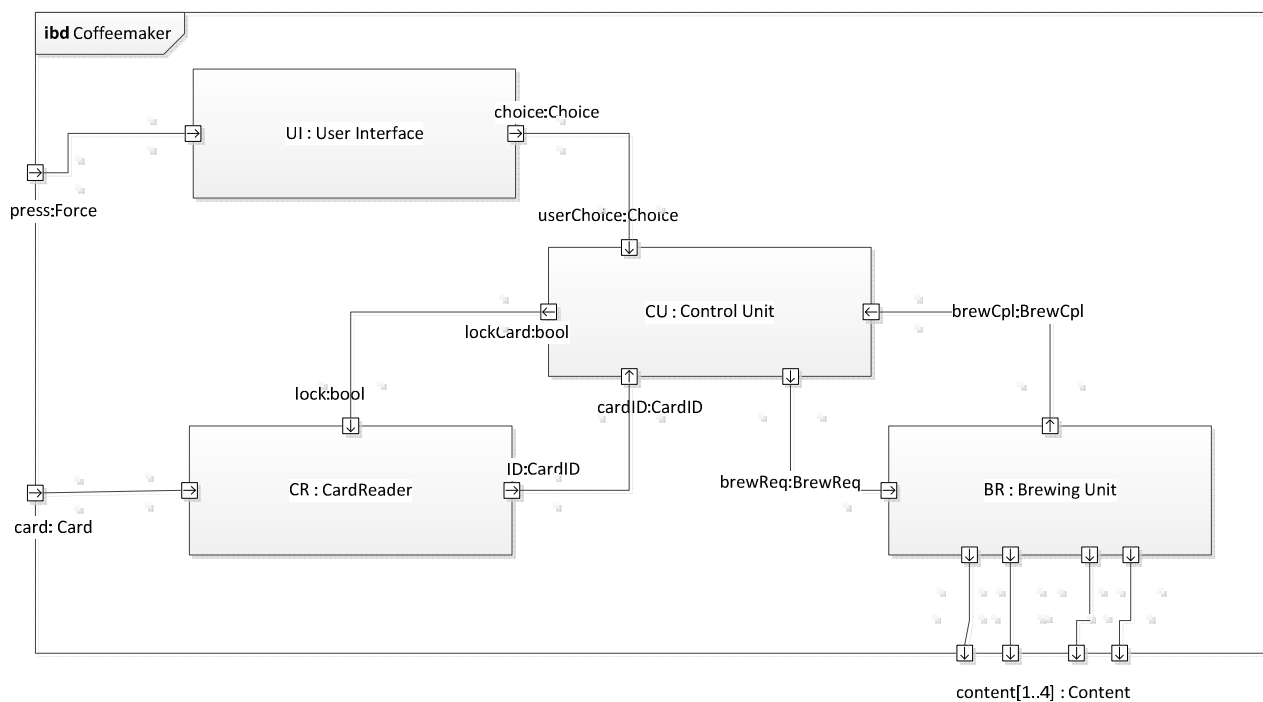


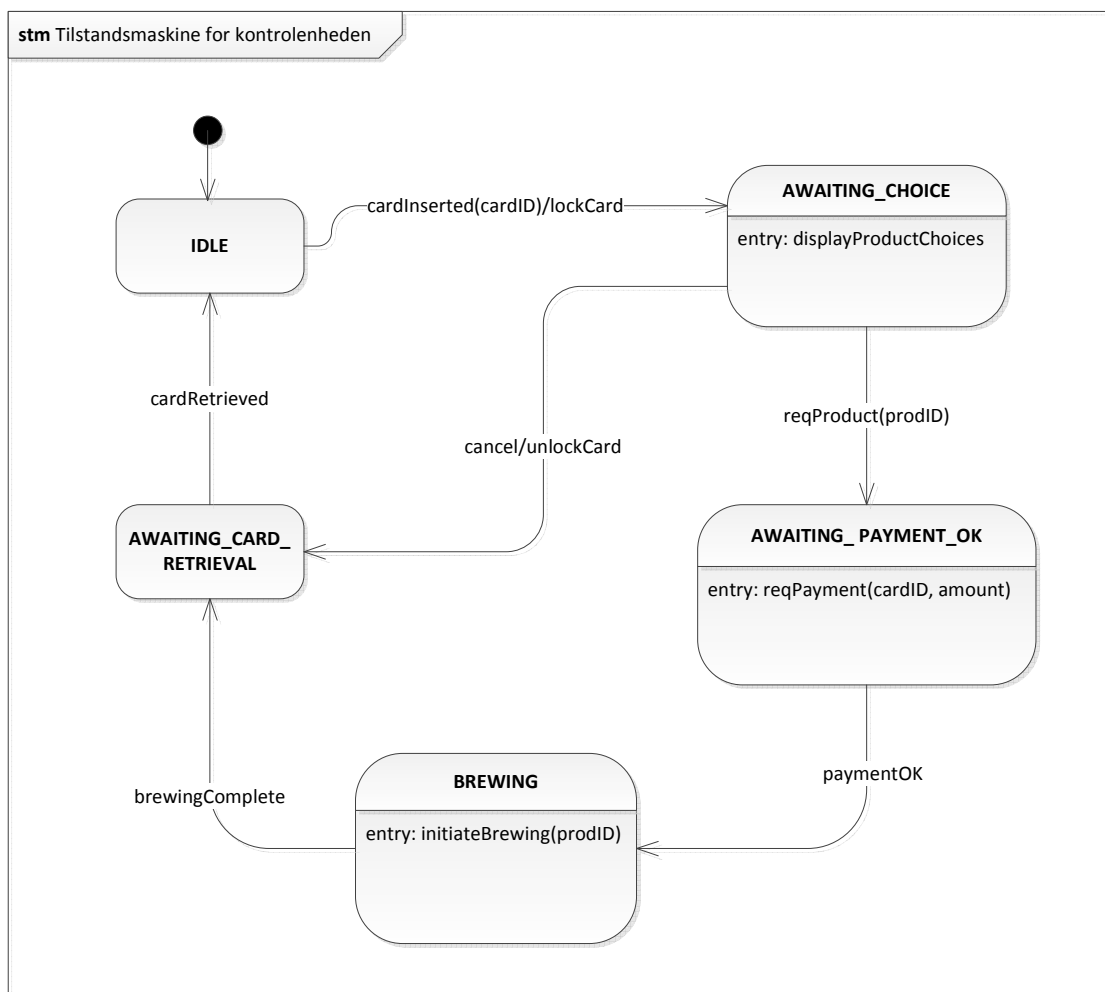
Figur 2 System-sekvensdiagram for Use Case Køb Produkt (Asynkron UI)

I nedenstående løsning er valgt synkron kommunikation mellem kunde og kaffeautomaten før valg af produkter. Dvs. kunden altid venter på at produkter præsenteres efter ID-kort er indsat.



Figur 3 System-sekvensdiagram for Use Case Køb Produkt (Synkront UI)

Opgave 3 (40 %):**Opgave 3A:****Figur 4 SysML internal block diagram (ibd) for strukturen i kaffeautomaten med parts, porte og forbindelser**

Opgave 3B:**Figur 5 Tilstandsmaskine for kontrolenheden**