

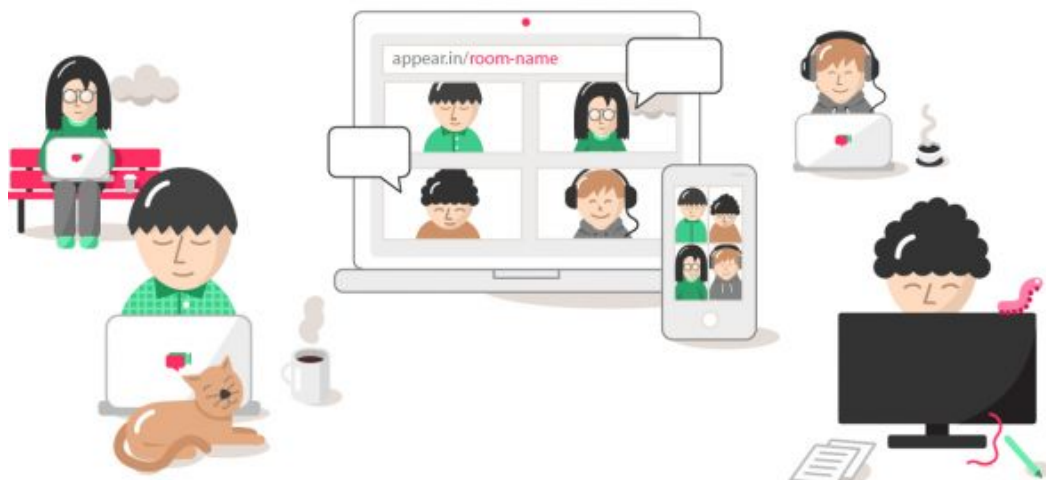
# Projektoplæg

Tværgående projekt

Fag:

Software Design

Software Construction



2. semester, forår 2020

# Indholdsfortegnelse

<b>1. Projekt Målsætning og formål</b>	<b>3</b>
1.1 Formel baggrund	3
1.2 Oplæg	3
<b>2. Problemformuleringen</b>	<b>5</b>
<b>3. Projekt Omfang</b>	<b>5</b>
<b>4. Projektrapporten</b>	<b>6</b>
<b>5. Projekt Proces Krav</b>	<b>7</b>
5.1 Projekt Grupper	7
5.2 Projekt Perioder	7
5.3 Vejledere i Projekt Perioden	7
5.4 Gode ideer til projekt processen	7
<b>6 Projekt Deadline - Aflevering</b>	<b>8</b>
6.1 Aflevering	8
<b>7 Eksamination og evaluering</b>	<b>8</b>
7.1 De formelle krav	8
7.2 Retningslinjer for Evalueringen	9
<b>8. Projektet og eksamen i en Corona-tid</b>	<b>10</b>
<b>9. Projektbeskrivelse</b>	<b>10</b>
9.1 Gruppen vælger selv en virksomhed	10
9.2 Case 1: Samkørsels Applikation	10
9.3 Case 2: Måleudstyrs Administrations Applikation	11

# 1. Projekt Målsætning og formål

## 1.1 Formel baggrund

I studieordningen (se Studieordningen på Zealands hjemmeside [https://www.zealand.dk/wp-content/uploads/2019/09/Studieordning-Datamatiker-2019\\_2.pdf](https://www.zealand.dk/wp-content/uploads/2019/09/Studieordning-Datamatiker-2019_2.pdf)) afsnit 8.1.2. i den lokale del (bagerst i studieordning) beskrives, at der skal være et projekt, der går forud for eksamen. Projektet skal munde ud en projektrapport som maksimalt må være på 40 normalsider. En normalside er 2400 tegn inklusiv mellemrum og fodnoter. Ved tvivl henvises til studieordningen.

Som følge af tiden med digital-undervisning mm. se kap 8 Projektet og eksamen i en Corona-tid

## 1.2 Oplæg

Anden semester eksamensprojektet omfatter emner fra de to undervisningsfag Software Design og Software Construction.

Det er projektets ide at designe og implementere et system (eller en del af et system) som kan køre som en UWP (universal Windows platform) Applikation. (se afsnit 8 om indholdet)

Projektet har følgende overordnede mål:

- At du får erfaring med:
  - faser i et Software Udviklings Projekt, der benytter metoden den agile udviklingsmetode.
  - sammenhængen mellem Business Analyse, Business Modellering, Analyse og Design, samt til Implementering og Test.
  - Project Management herunder teknikker til reducere af risikoen og forbedre system kvaliteten.
  - hvorledes Persistens kan opnås gennem et Object-Oriented design, Programmering og relationelle databaser (SQL-databaser).
  - hvorledes MVVM arkitekturen kan benyttes til at bygge Applikationer til Windows.
- At du anvender moderne systemudviklingsmetoder og teknikker til Object-Oriented Analyse, Design og programmering af (eller dele af) et single bruger IT system.
- At skabe grundlaget for selve eksaminationen.



## 2. Problemformuleringen

I dette projekt benyttes den samme problem formulering for alle grupper.

***”Hvordan kan den agile udviklingsmetode, C#-Programmering og Relational Databaser benyttes til at udvikle, designe og implementere et mindre IT system?”***

I forbindelse med problemformuleringen **må** du angive et eller flere del-problemstillinger, som du vil have fokus på i projektet.

*For Eksempel:*

- *Hvordan er det muligt at følge brugernes krav gennem system udviklings processen?*
- *Hvorledes er det muligt at reducere risici og forbedre kvaliteten ved anvendelse af forskellige projekt management teknikker?*
- *Hvordan kan Persistens opnås ved brug af en database, som ligger i en 'cloud'-løsning?*
- *Hvordan opnår man en god GUI, når der anvendes UWP og MVVM?*

Du kan undervejs i projektforsløbet, hvis du finder det nødvendigt, udvide, tilpasse, tydeliggøre eller begrænse din problemformulering, dvs. dine del-problemstillinger. Med udgangspunkt i problemformuleringen og del-problemstillingerne skal du gennem projektets aktiviteter opbygge erfaring, udarbejde dokumentation samt implementere en kørende prototype, så du kan redegøre for disse i din rapport og til eksamen.

## 3. Projekt Omfang

Det er et krav, at du anvender den metode, som den er beskrevet i “The agile Samurai” af Jonathan Rasmusson og andre analyse og designmetoder der har været anvendt og præsenteret i forbindelse med undervisningen i Software Design.

Det er ligeledes et krav, at den kørende prototype (del af systemet) bliver programmeret i C#, Visual Studio samt med en MSSQL relationel database (der kan ligge i 'cloud'en).

Det er tillige et krav, at du får behandlet følgende faser:

- Projekt etablering
- En række sprints til iterativt at implementere kerne arkitekturen – hvor der benyttes en cloud baseret database, såvel som UWP (MVVM) til en Windows GUI grænseflade.
- (måske kan Construction delvist berøres)

## 4. Projektrapporten

Den overordnede rapportstruktur skal være:

- **Forside og titel**
- Introduktion
- Hovedafsnit
- Konklusion
- Litteraturliste
- Bilag (inkluderer kun bilag, som er centrale for rapporten)

Desuden skal der være

- **Angivelse af sti til versionstyringsserver**, hvor kildekoden og eksekverbar kode til produktet kan hentes (hvis en sådan en forefindes)
- Angivelse af hvilke **afsnit de forskellige studerende er ansvarlige** for.
- En underskrift fra alle studerende ved aflevering i Wiseflow (foregår i Wiseflow, hvor alle bekræfter at de afleverer)

Det er værd at lægge mærke til det sidste punkt. Det er et krav, at det tydeligt angives, hvilke afsnit/dele af rapporten den enkelte studerende har været hovedansvarlig for. Hver studerende bør som minimum være ansvarlig for et afsnit/del, som berører emner inden for hver af de to undervisnings fag Software Design og Software Construction.

Det kan godtages, at centrale dele af rapporten kan have mere end en ansvarlig studerende.

## 5. Projekt Proces Krav

### 5.1 Projekt Grupper

Grupperne må ikke overstige 5 personer.

### 5.2 Projekt Perioder

Semesteret kan deles op i 4 perioder i relation til dette projekt:

1. Periode, 9 uger (uge 6-14) I den første periode er bestemt til en teoretisk basis database modellering, genopfriskning af programmering, indføring i Collections og Generics m.m.
2. Periode, 2 uger (uge 16-17) I denne periode færdiggøres teorien samtidig med der laves gruppedannelse, og de indledende aspekter af Inception deck.
3. Periode, 5 uger (uge 18-22 undtagen uge 15) Denne periode er til fuldtids studier i projektet.

### 5.3 Vejledere i Projekt Perioden

Dine undervisere vil være vejledere på projektet. De vil følge med i projektet gennem hele forløbet. Det er meningen at grupperne mindst én gang pr undervisningsdag har et møde med en af vejlederne. Hvis gruppen ønsker tilbagemelding på udvalgte dele af processen, rapporten eller prototyperne, kan dette ske ved at sende materialet forinden i god tid til underviseren.

Det er vejlederens rolle at anvise en vej, komme med råd og retningslinjer for projektet (rapport såvel som prototype), men det er ikke meningen, at underviseren skal være en del af gruppen og komme med konkret tekst til rapport eller udarbejde kode. Vejlederne vil være til rådighed i de perioder, der er afsat timer i skemaet.

### 5.4 Gode ideer til projekt processen

- Gruppen bør have en logbog af deres møder og beslutninger. Dette kan delvist gøres gennem et scrumboard f.eks. Trello. Det kan være en god ide at vedlægge logbogen i et bilag.
- Referater af vigtige møder med vejleder eller kunden.

## 6 Projekt Deadline - Aflevering

### 6.1 Aflevering

Projekt rapporten skal afleveres i Wiseflow ikke senere end:

**Onsdag 27. Maj senest kl 11:00**

Krav i forbindelse med afleveringen.

- Rapporten skal afleveres som **PDF-fil**.
- Programkode skal ligge, så det kan hentes fra GitHub, samt sti til Projektets GitHub skal fremgå tydeligt i afleveringen.
- Alle gruppe-medlemmer skal ind i Wiseflow og godkende afleveringen

## 7 Eksamination og evaluering

### 7.1 De formelle krav

Eksamen er planlagt til juni måned og oplyses af din underviser. Oplysning om tidspunkt mm. kommer senere.

De egentlige krav fremgår af studieordningen (se [https://www.zealand.dk/wp-content/uploads/2019/09/Studieordning-Datamatiker-2019\\_2.pdf](https://www.zealand.dk/wp-content/uploads/2019/09/Studieordning-Datamatiker-2019_2.pdf)) afsnit 8.1.2. i den lokale del af studieordningen (bagerste del)..

Bedømmelseskriterierne for prøven svarer til læringsmål for de nationale fagelementer: Programmering, Systemudvikling, Teknologi og Virksomheden. Se afsnit 2.1-2.4 i den nationale del af studieordningen.

Der gives én samlet karakter for rapport, fælles præsentation og den individuelle eksamination tilsammen.



## 7.2 Retningslinjer for Evalueringen

### For rapporten gælder:

- Den er velskrevet og velstruktureret.
- Rapporten fremstår som et sammenhæng forløb.
- At der er en rød tråd fra Software Design til Software Construction.
- Den relevante teori bliver behandlet og beskrevet.
- At Diagrammer er forklaret.
- At væsentlige valg fremstår som et valg på baggrund af en diskussion af emnet med alternativer, samt begrundelse for valget.
- At problemformuleringen bliver besvaret.
- At prototypen fremstår vel struktureret med kommentarer.
- At udvalgt kode er forklaret i selve rapporten

### For fælles præsentationen gælder:

- Der er 10 min. pr studerende, dog højst 30 min.
- Alle i gruppen står for en præsentation.
- Inddrag en demonstration, hvor du/I har planlagt, hvad du/I vil demonstrere.
- Forhold dig/jer kritisk til projektet – hvad var godt? Hvad kunne gøres bedre?
- Inddrag evt. nye aspekter eller vinkler på projektet end dem, der er beskrevet i rapporten.

### For den individuelle eksamen gælder:

- Der er 30 min. Inklusiv afgivning af karakter (dvs. ca. 16-17 min.)
- Spørgsmål bliver stillet ud fra projektet i begge de to undervisningsfags pensum. I praksis deles tiden mellem de to undervisere.
- Der bliver lagt vægt på den studerendes evne til at svare og reflektere over emnerne.

### Generelt:

Da eksamen er individuel, kan der afgives forskellige karakterer til studerende fra samme gruppe, selvom udgangspunktet dvs. rapporten og prototypen er fælles.

## 8. Projektet og eksamen i en Corona-tid

Dette projekt bliver desværre afviklet under en tid med folketingets udstukne linjer for samvær, derfor vil dele eller måske hele perioden blive gennemført, hvor vi kommunikerer over virtuelle medier.

Du kan i perioden følge med i udviklingen i Moodle <https://moodle.easj.dk/>

Hvis Zealand stadig er nedlukket, når der skal afholdes eksamen, vil eksamen blive afholdt digitalt efter følgende retningslinjer:

- [Studerende](#)
- [Undervisere](#)
- [Censor](#)

Den digitale platform vil være Zoom, se mere <https://helpdesk.zealand.dk/hc/da>

## 9. Projektbeskrivelse

I skal lave et mindre IT system. Der er flere muligheder, gruppen kan vælge at selv at finde en virksomhed at lave et projekt til eller kan benytte en af to case beskrivelser herunder.

Hvis gruppen vælger at arbejde med en af de to cases, vil de to undervisere agere Product Owners og gruppen vil selvfølgelig være scrum-teamet.

### 9.1 Gruppen vælger selv en virksomhed

Gruppen vælger selv en virksomhed, som den finder en kontakt til, som de kan kommunikere med.

Gruppen skal have grønt lys fra underviserne til at opgaven passer til et 2.semester projekt.

De to undervisere kan evt. benyttes som product owners, hvis der ikke er en tæt kontakt til virksomheden.

### 9.2 Case 1: Samkørsels Applikation

Der er et ønske fra Zealands side om at mindske antallet af biler på parkeringspladsen, samt understøtte transport i en mere grøn retning.

Derfor er ideen, at der skal udarbejdes en applikation, hvor studerende og undervisere kan organisere, til- og afmelde samkørsel til og fra skolen. Der er ikke penge mellem deltagere i

samkørslen, men ideen bygger på der alligevel er nogle der kører eller at man kan skiftes til at køre.

For at kunne deltage skal du oprette en bruger, som enten chauffør eller deltager, samt kunne angive din kørselsrute, samt indikere om nogle ønsker at stige på undervejs til og fra skolen.

Der skal være mulighed for at lave planlagte samkørsler ligesom spontane brug af samkørsel.

Uden sammenligning i øvrigt tænkes på et løsning inspireret af [Ubers brugergrænseflade](#), men helt uden penge. Der skal derfor ikke tænkes på skat mm.

## 9.3 Case 2: Måleudstyrs Administrations Applikation

I skal lave et mindre IT system til administration af 'Miljøvidenskab' instituttets måleudstyr, der ligger i Risø og er en underafdeling af Aarhus Universitet.

### Beskrivelse af problemstillingen

Dette er en beskrivelse af "stationsloggen" - uden forslag til IT løsning. Instituttet har ansvaret for drift af omkring 20 luft målestationer i Danmark samt én på Grønland. Målestationerne er typisk beliggende i måle skure langs veje, i master eller i offentlige bygninger<sup>1</sup>

På alle stationer, der er omfattet, findes én til flere computere som anvendes til datalogning.

Stationerne er forskellige mht. måleudstyr. Der findes fra 2 til 15 instrumenter som på en eller anden måde anvendes til at måle luftkvaliteten, herunder meteorologi. Målestationerne serviceres ugentlig samt månedlig af teknikere.

Under de ugentlige besøg har teknikerne nogle få opgaver (skift af filtre, aflæsning af tryk osv.)

Under de månedlige besøg er opgaverne mere omfattende, bl.a. skift af monitorer og gasflasker.

Under de "faste"/planlagte besøg registreres en række parametre, og der udføres diverse driftsopgaver. Principielt har registreringerne 2 formål:

- Registreringerne anvendes som til den efterfølgende data korrektion.
- Registreringerne anvendes ved den efterfølgende kvalitetskontrol af data. Hvis der ved kontrollen fx findes uventede koncentrationer, gennemses stationsloggen, for at

---

<sup>1</sup> Bemærk: Instituttet driver herudover en række stationer med passive samplere og nedbørssamplere, hvor "stationsloggen" ikke er relevant.

se om der er sket noget "uventet" på stationen i den pågældende periode. Hvis der mangler data gennemses stationsloggen for fx at konstatere om der fx har været "nedbrud" på instrumentet.

**Herunder nævnes nogle typiske eksempler på registreringer.**

- Aflæsning af tryk og temperatur
- Aflæsning af tid på instrumenterne
- Udskiftning af filtre på monitorer
- Rensning af indtags hoveder
- Rensning af slanger
- Måling af flow på instrumenterne
- Udskiftning af instrumenter
- Udskiftning af gasflasker
- Aflæsning af nul og span målinger
- Opstilling af nye instrumenter
- Nedtagning af instrumenter
- Kalibrering af instrumenter
- Udskiftning af passive samplere fx rør.
- Justering af flow på instrumenter
- Periode med strømsvigt<sup>2</sup>
- Perioder hvor et instrument har været standset.

Ud over de "faste" besøg, besøges stationerne ved akut opståede problemer, eller ekstra opgaver.

Eksisterende log<sup>2</sup>

Der eksisterer i dag et elektronisk "stationlog" system, hvor man på "stationscomputeren" kan indskrive, hvad man har foretaget på stationen samt beskrive problemer med udstyr..

Oplysningerne sendes sammen med data til en filserver, og data overføres til en database. En webapplikation anvendes til at se databasens indhold

Deciderede aflæsninger og resultat af kontrolmålinger foretages på papir. Efter hjemkomst indskrives aflæsninger/resultater i manuelt excelark.

Nogle af registreringerne i excel-arkene anvendes efterfølgende som korrektion af udeluft målinger (fx kontrolmålinger med nul/span målinger på kendt koncentration).

**Kravene til et nyt stationslog system.**

---

<sup>2</sup> Eksempel på en stationslog til sidst i dokumentet

Systemet skal være fuldt automatiseret vedr. udveksling af registreringer mellem stationer, filserver samt databaser. Registreringerne skal kunne foretages på stationscomputeren eller en medbragt tablet.

Applikationen til registrering skal kunne ske "lokalt", da internetforbindelse ikke altid er til rådighed på målestationerne.

Oplysningerne skal "struktureres". Fx. så man senere kan søge på, hvad der er foretaget på en specifik type instrument på en bestemt station.

De oplysninger ( aflæsninger & kontrolmålinger ) der i dag skrives på papir skal kunne registreres elektronisk, så man spare tid på at foretage den elektroniske registrering efterfølgende.

Data fra aflæsning af nul/span aflæsninger, udskiftning af spangasser samt monitorer skal kunne overføres til den database, hvor de anvendes til korrektion og kvalitetskontrol af data.

Der sker, udover manuelle registreringer, nogle automatiske "logninger" på stationscomputeren bl.a. genstart af computer & udstyr, temperatur overskridelser, som skal indgå i stationsloggen.

Data skal sendes til AU's filserver/databaser løbende eller manuelt under/efter besøget.

Det skal være muligt via en webapplikation, eller lignende, at slå op de registrerede data.

## Eksempel på en stationslog

### ATMI Stationslog V 8.2 nov 2010

Vælg Station		Periode	
Dato	Station	Initialer	Bemærkning
12-12-2017 11:06:00	2090	tomj	ugecheck
12-12-2017 10:30:00	2090	tomj	SMPS dataops-pc ej aktiv (sandsynligvis pga Win10 1709 opdatering)
05-12-2017 15:06:00	2090	tomj	skiftet DOVADO ud med anden DOVADO :)
05-12-2017 14:13:00	2090	kem	Router skiftet fra TP-Link til Dovado (fra Ispra kuffert)
05-12-2017 11:29:00	2090	hwm	Ugecheck: SMPS flow: 0,997 L/min v Impactor. 1,013 L/min v CPC. målt med Bobbelflowmeter. Classifier 1,04 l/min aflæst
04-12-2017 13:06:00	2090	tomj	Nedtaget DOVADO - opsat TPLink 3G router
04-12-2017 12:26:00	2090	Hwm	TEOM opsat. Total flow 16,2 l/min K-001. Main: 2,9 l/min (TSI) lige efter opstart
01-12-2017 11:14:00	2090	stib/hwm	SMPS: pm1 cyklon skiftet sample rør hævet 30 cm. FPO tid ændret 10:15 -> 10:23