*Gem denne skabelon med et andet navn.*

*Slet vejledning til de enkelte afsnit (alt med kursiv - også dette)*

*Anvendes skabelonen i et projekt med specifikke krav til indholdet i de enkelte afsnit, så indsæt disse krav i den nye skabelon.*

*Rapportskabelonen bruges fra 1. - 4. semester til alle projekter, eksamensprojekter og til praktikrapporten i 4. semester.*

*I skal selv tilpasse rapportskabelonen i forhold til det projekt I arbejder på. Se kravene fra projekt – og eksamensbeskrivelserne.*

*Dette kan for eksempel være tilfældet i IoT-projekterne*

Titelblad med angivelse af projekttitel,

Navne,

Gruppenummer,

Klasse,

Opgave,

Vejledere,

Antal anslag

[Rapportens titel]

[Illustration]

# 

# **1. Resumé**

*Det skal indeholde en sammenfatning af hele rapporten med både problemstilling, hvilke resultater man opnåede samt konklusion.*

*Dette afsnit skrives til sidst. Resuméet skal give læseren mulighed for hurtigt at få et overblik over rapportens indhold*

***Må max fylde ½ side***

# 

# 2. Indholdsfortegnelse

# 

# **3. Indledning**

*Indledningen kridter banen op: Hvad er det for et emneområde, vi bevæger os i, og hvorfor gør vi det?*

***Indledningen*** *introducerer læseren til jeres opgaveemne og jeres behandling af det. Den er vigtig, fordi det er her, jeres læsers forventninger til opgaven skabes. Vær tydelig og informativ i* ***indledningen****. Her introducerer I til det problem, I tager fat på i opgaven, og hvordan I vil løse det.*

## **3.1 Beskrivelse af virksomhed**

*Hvis der i projektet indgår et samarbejde med en virksomhed, laves der her en kort beskrivelse af virksomheden.*

***Må max fylde 1 side***

# 

# **4. Problemformulering**

*Start med en kort komprimering af indledningen. Herefter angives kort, præcist og konkret det ene hovedspørgsmål, der skal undersøges og besvares i projektet. Hovedspørgsmålet uddybes med underspørgsmål. Der kan være 2-4 underspørgsmål.*

*Ingen spørgsmål må kunne besvares med et kort svar såsom ”ja”, ”nej” eller et tal såsom 42.*

*Brug* ***"Hvad",******"Hvorfor"*** *eller* ***"Hvordan"*** *i formulering af spørgsmål.*

**Afgrænsning af problemformulering**

*Her kan I med fordel afgrænse jeres problemformulering.*

*Problemformulering skal besvares, det ligger helt fast. Men af og til kan man havne i den situation, at problemformulering kunne indbyde til en meget større og bredere opgave end mening var. Så er det, at afgrænsningen kommer ind i billedet. Således beskriver I her hvilke dele af problemet I vil undlade at komme ind på. Bemærk at det kun er nødvendigt at afgrænse problemformuleringen hvis den er meget bred.*

*Problemformuleringen udarbejdes på baggrund af den godkendte projektbeskrivelse.*

*Husk: Der skal skrives en konklusion ud fra problemformuleringen.*

***Må max fylde 1 side***

Under vores problemformulering valgte vi at arbejde med problemet:

“**Det er et problem, at skovbrande opdages for sent”**

Efterfølgende har vi udarbejdet et par afgrænsninger, siden emnet er ret bredt.

* “**Hvordan kan man mindskes skovbrande ved hjælpe af en IOT løsning” / Hvordan kan man gennem en IOT løsning, reducere skaden der kan forkomme gennem skovbrande**
* **“Hvordan kan en løsning anvende RPI med en hjemmeside som modtagere indsamlet data”**
* **Løsning må maks anvende følgende sensor: luftfugtigheds, temperatur, røg -sensor”**
* **“Hjemmesiden skal anvende Flask og SQLite til database.”**

Vi valgte følgende afgrænsninger for at begrænse løsningen til at den passer til cirka 4 ugers skoleforløb, særligt ved begrænsning af max 3 sensorer. Vores begrundelse for dette er vi vurdere det burde kunne nå i mål med en løsning som kan modtage information ved en ESP32 med nogle sensor, og kunne sende data til en RPI (RaspBerryPi) som indsætte det i en graf og indexere dataen på en statisk hjemmeside. Vores data som bliver indsamlet gemmes i en SQLite database, som kan hentes og indsættes i vores hjemmeside.

**5. Indledende undersøgelse**

*Her beskriver I jeres ideudvikling, research, brugerundersøgelse og evt. brugen af Design Thinking hvis det er et krav i projektet.*

*Ved projekter hvor der indgår Business Case, skal denne lægges i Bilag.*

## **5.1 Ideudvikling**

*Hvordan arbejdet med ideudvikling forløb og hvad I kom frem til.*

## **5.2 Research**

*Her beskriver I jeres research i forbindelse med ideudvikling og for at opnå større viden om jeres emne.*

## **5.3 Brugerundersøgelse / Design Thinking**

*Hvilken form for brugerundersøgelse har I gennemført og hvad var resultatet af denne/disse. Hvordan kan den opnåede viden bruges i forhold til jeres projekt.*

*Hvis der bruges Design Thinking skal forløbet og resultater indsætte her.*

## **5.4 Persona og User stories**

*Hvem er Persona til projektet? Sæt gerne nogle eksempler ind. Hvilke User Stories har I beskrevet til hver Persona? Sæt gerne nogle eksempler ind. Resten af Persona og User Stories skal i bilag.*

# 

# **6. Kravspecifikation og accepttest metode**

*En kravspecifikation beskriver de krav, en kunde har til et produkt. Kravspecifikationer har mindst 10 krav, og kan være op til flere hundrede for store systemer.*

***ID:*** *Et unikt nummer som bruges til at referere til det stillede krav.*

***Kategorier****: Forsøg så vidt muligt at pakke jeres krav i standardkategorierne: Power, UI, Kommunikation og Brugsmiljø. Definer gerne egne kategorier, hvis det pågældende krav ikke passer ind i standardkategorierne.*

***Prioritet:*** *Den prioritet kravet har for projektet som en helhed. Dette fastlægges med kunden. 1. prioriteter skal være inkluderet i den endelige løsning. Der er 3 prioritetsgrader at vælge mellem.*

***Krav****: Benyt ord som “skal” når det er 1. prioritet, “bør” når det er 2. prioritet og “kan” når det er 3 prioritet. Det, man udvikler, er en “løsning” - ikke en prototype eller et produkt.*

***Accepttest****: Den test der skal udføres for at validere, at det stillede krav er opfyldt. I en accepttest bruges betegnelsen DUT = Device Under Test frem for løsningen, da løsningen kan have mange forskellige versioner igennem projektet. DUT er derimod en låst version.*

| **ID**: 2 | **Krav**: Løsningen skal have en batterilevetid på minimum 3 dage ved normal brug. Normal brug defineres som, at løsningen aktiveres 12 gange om dagen, ellers er enheden i dvaletilstand. | **Prioritet**: 1 |
| --- | --- | --- |
| **Kategori**: Power | **Accepttest**: Krav testes ved at placere DUT på et bord, og batteriet lades op ved brug af oplademetode defineret i krav 3. Når batteriet er fuldt opladet stoppes ladning, ved at udtrække lade kabel og en timer startes. Hver anden time vækkes DUT ved at trykke på “on” knappen. Testen afsluttes når “Power on” LED’en ikke længere lyser, og den endelige tid noteres. Er den endelige tid over 72 timer, så vurderes testen som at være bestået. | Passed/ Failed/ Ikke testet |

*Det er muligt at ændre krav og accepttest efter første testcyklus.*

*Det nye krav skal have argumentation for kravændring.*

*De originale krav skal være synlige, brug versionsnumre til at tydeliggøre historikken af udviklingen.*

Skabelon for krav og accepttest specifikation tabel:

| **ID:** | **Krav:** | **Prioritet:** (1-3) |
| --- | --- | --- |
| **Kategori:** | **Accepttest:** | Passed/ Failed/ Ikke testet |

## **6.1 Rationale for prioritering af krav**

*Redegør kort hvorfor kravene har de prioriteter som er sat i afsnittet ovenover. Det skal stå klart, hvorfor I vælger at prioritere specifikke krav frem for andre.*

# **7. Analyse**

*Dette afsnit beskriver jeres analyse af de teknologier og løsningsforslag der fører til fuldførelse af jeres krav som er defineret i foregående afsnit. Her kan I skrive den teoretiske analyse af hvilke teknologier der i generel form vil kunne løse jeres problemer.*

## **7.1 Arkitektur af pathfinding**

*Læg ud med at lave et blokdiagram og fremvis hvordan de grundlæggende byggeblokke i dit projekt er forbundet. Som I finder frem til hvilke teknologier der fungerer bedst til at opfylde jeres krav, så vil jeres blokdiagrammer også udvikle sig. I denne sektion fremviser I jeres progression fra den ene version til den anden.*

***Eksempel for indlejrede systemer***

| **Pathfinding version 1** | **Pathfinding version 2** |
| --- | --- |
|  |  |

Diagrammer kan med fordel laves i onlineværktøjet [diagrams.net](http://diagrams.net/).

***Eksempel for programmering***

| **Programmering pathfinding version 1** | **Programmering pathfinding version 2** |
| --- | --- |
|  |  |

***Eksempel for Netværk***

***Eksempel for server***

## **7.2 Den endelige arkitektur**

*Her fremviser I det endelige blokdiagram der afspejler den arkitektur som jeres DUT er bygget ud fra.*

## **7.3 Teori af relevante elektroniske/programmering/netværk/server arkitekturblokke**

*I dette afsnit bryder I blokkene i den endelige arkitektur ned, og beskriver;*

* *Hvilken præcis teknologi I har valgt at benytte*
* *Hvordan teknologien bag dem fungerer*
* *Hvilke begrænsninger og potentielle udfordringer teknologien har*
* *Hvorfor I specifikt har valgt at bruge netop den valgte teknologi*
* *Hvilke elementer taler sammen*
* *Hvilken vej eller hvilke veje data flyder igennem jeres arkitektur*
* *Hvilke protokoller sendes eller modtages data gennem*
* *Hvad er baggrunden for jeres valg*

*Eksempel for indlejrede systemer:*

| **Blok navn: Batteri** | |
| --- | --- |
|  | Dette batteri har en kapacitet på 1800 mAh, og har en spændingsprofil der går fra 3.0 V til 4.2 V.    Batteriet kan max. aflade med 2 C, hvilket svarer til 3600 mA, som er langt under løsningens forbrugsniveau på 230 mA.        Batteriet er 52 mm lang, 30 mm bred, 9 mm tyk, og vejer 13 gram.    I dette projekt bruges batteriet til at forsyne en ESP32, gennem en boost converter(beskrevet i blokken “Boost converter”).    Batteriet lades også via en batterilader beskrevet i blokken “Batterilader” |

*Eksempel for programmering:*

| **Blok navn: Blink LED 10 gange**    I filen xx.py fra linje xx til linje xx |
| --- |
| Blink LED 10 gange med et non-blocking delay med 500 millisekunders interval mellem hvert blink, og sluk derefter for LED og start igen. Man bruger funktionen led.value(1) og funktionen sleep\_ms(500). Derfor skal sleep\_ms() importeres fra time modulet og Pin skal importeres fra machine modulet. Hele denne blok pakkes ind I en funktion kaldet blink10x(), og denne funktion returnerer ikke noget. |

# 

# **8. Løsningsdesign**

*Det er vigtigt at man dokumenterer denne sektion meget konkret, således at et andet udviklerteam, eller konsulenthus kan genskabe jeres løsning ved at læse denne sektion.*

*Afsnittet skal tilpasses alt efter hvilket projekt I arbejder med.*

## **8.1 Elektriske skematikker, målinger og udregninger (Indlejrede systemer)**

*Hvis I har kunne lave en elektrisk skematik af jeres løsning (på DUT niveau), så kan I med fordel indsætte den her. Har I beregnet komponentværdier, strømforbrug og målinger så indsættes det også her.*

## **8.2 Router/Switch opsætning og Interface forbindelser (Netværk)**

*Her beskrives hvordan I har opsat jeres routere, switche m.m. på DUT niveau.*

***Eksempel på et interface forbindelsestabel:***

## **8.3 Kode beskrivelse (Programmering)**

*Her udvælger I* ***centrale og relevante*** *dele af jeres kode og beskriver deres virkemåde. Det er ikke acceptabelt bare at copy paste hele ens kode i denne del af rapporten. Hele koden pastes i stedet for i bilag delen.*

*I kan supplere jeres beskrivelse ved hjælp af flowdiagrammer og andre relevante værktøjer.*

*Husk at der skal være linjenumre på den vedlagte kode, så det er nemmere at referere til specifikke elementer.*

*Eksempel:*

*Kodeeksempel 1*

*I kodeeksempel 1 som er fra en fil kaldet dht11sensor.py er der oprettet en funktion til at lave en måling med DHT11 sensoren og returnere en liste med temperatur på index 0 og luftfugtigheden på index 1. Funktionen defineres på linje 4 med navnet measure og et fast argument kaldet pin\_number som får standardværdien 14. På linje 5 initialiseres et DHT-objekt, sat til pin 14. På linje 6 kaldes measure() funktionen på DHT-objektet sensor. På linje 7 returneres temperatur og luftfugtighedsmålingerne.*

## **8.4 Konfiguration af valgte servere (Server)**

*Her beskrives hvordan I har opsat jeres servere i topologi diagram sammen med netværksudstyr.*

*Hvilke konkrete konfigurationer har I benyttet i jeres servere, hvilke OS har I benyttet ….*

**9. Test af løsning**

**Afsnit skrevet af:**

*Dette afsnit skal afspejle to faser af jeres tests; jeres pathfinding test og resultater, og testen af jeres endelige DUT.*

## **9.1 Pathfinding testresultater**

*Når I laver jeres pathfinding, er det vigtigt, at jeres testopstilling er godt dokumenteret. Andre skal kunne genskabe jeres projekt og opnå de samme resultater, som I fik.*

**Eksempel på testopstilling**

*Figur nr. x. Dette viser testopstillingen………*

## **9.2 Dokumentation af DUT**

*I denne sektion skal I dokumentere, hvordan den endelige version af jeres løsning ser ud og fungerer. Denne version af jeres løsning skal testes med alle jeres accepttestspecifikationer. I tester også alle jeres krav på denne version, som I kalder DUT (Device Under Test).*

**Billeder og video af samlet løsning**

*Videoer vedlægges via korte links. Billeder skal have billedtekst og pile med beskrivelser af hvilke subkomponenter, DUT’en består af.*

**Specifikationer**

*En beskrivende liste af de specifikationer ens DUT har på stadiet af rapport aflevering, og på det ønskede endelige stadie. Listen herunder er et eksempel og er ikke udtømmende.*

|  | **Nuværende DUT stadie** | **Ønsket endelig stadie** |
| --- | --- | --- |
| **Højde, bredde, længde** |  |  |
| **Vægt** |  |  |
| **Indgang spænding** |  |  |
| **Effektforbrug** |  |  |
| **Netværksprotokoller** |  |  |
| **Sikkerhedsfunktioner** |  |  |

## **9.3 Udførsel af brugertest**

*Her skal I beskrive jeres brugertest proces, hvilke resultater I fik, hvad gik godt, hvad gik galt, og hvor lang tid I brugte under hver testcyklus.*

| **Dato** | **Test beskrivelse og resultater** |
| --- | --- |
| 19.10.21 | Løsningen var ude og blive testet hos FCK u18 holdet, på en fodboldspiller. Der blev trænet tacklinger og målspark. Vesten blev påsat fodboldspilleren, og systemet blev igangsat ved træningsstart.    Vi oplevede at systemet gik ned efter 16 minutter, hvor vi måtte genstarte elektronikken på spilleren.    Efter 48 minutter mistede vi forbindelse til vesten, og kunne ikke genskabe forbindelsen, hvorefter testen blev afsluttet.        Ved nærmere inspektion, efter hjemkomst kunne vi se at lodningen til batteriet var gået i stykker. Udover dette kunne vi se at der var kommet vand ind i fugtmåleren, som gjorde at fugtmålingerne viste fejlresultater efter 17 minutters test.        Der blev i alt testet i 48 minutter. |
| 20.10.21 | Vi fik repareret lodningen til batteriet, og placeret fugtsensoren i en 3D printet æske, på basis af resultaterne af sidste test.    Xxxx    Xxx    Xx    Der blev i alt testet i 17 minutter |
|  |  |
|  |  |
| **Samlet test tid under projektet: 156 minutter** | |

## **9.4 Udførsel af Accepttest på DUT**

*I denne sektion dokumenterer I resultaterne af jeres accepttest.*

**Test af krav 1**

| **ID**: 1 | **Krav**: Løsningen skal være batteridrevet, med udskifteligt batteri | **Prioritet**: 1 |
| --- | --- | --- |
| **Kategori**:    Power | **Accepttest**: DUT åbnes uden værktøj, batteri udtages, nyt batteri placeres ind i DUT. Hele handlingen times, og må ikke tage mere end 2 minutter. Batteriet må ikke kunne vendes forkert og være til fare for brugeren.    Dette skal testes på 10 mennesker uden teknisk baggrund. Hvis alle 10 lykkes at skifte batteriet indenfor tidsrammen, så anses testen for bestået. | **Det vurderes at kravet er fuldført.** |
| **Diskussion og evt. ønskede ændring af krav:**    Alle brugere brugte en stor del af tiden på at regne ud, hvordan man åbnede bag-skallen på DUT’en, derfor foreslås en forbedring i form af en markering på DUT’ens bagskal, således brugeren mere intuitivt kan finde åbningen. | | |

**Testopstilling**

*Her præsenteres jeres testopstilling, således jeres test kan genskabes af en anden part.*

| Bruger tilgår batteriet | Bruger udtager batteriet |
| --- | --- |
|  |  |

**Resultat**

*Det lykkedes for en testgruppe af 10 personer at skifte batteriet på DUT på 1 minut og 12 sekunder i gennemsnit, med den højeste tid på 1 minut og 49 sekunder.*

*Samme forløb gennemgås for alle krav.*

# 

# **10. Implementering af løsning i drift (eventuel)**

*Hvis I når at udvikle jeres løsning, så den er klar til at gå i drift hos kunden, skal det beskrives og vises her.*

# 

# **11. Praktisk projektplanlægning og ledelse**

*Her beskrives, hvordan projektledelse er anvendt i projektet. Beskrivelse af WBS, Kanban/Scrum og Gantt diagram, samt hvordan det er anvendt i dette projekt.*

*Desuden skal jeres analyser med i dette afsnit, hvis I har lavet analyser, der er relevante for projektledelse.*

*Fx SWOT, risiko- og interessentanalyser, virksomhedsanalyser osv.*

*Underafsnit kan struktureres, som I mener, er relevant for jeres projekt*

## **11.1 WBS**

## **11.2 Gantt**

## **11.3 Kanban / Scrum**

## **11.4 Projektanalyse**

*Indsæt jeres projektanalyser i dette afsnit, hvis I har lavet analyser der er relevante for projektledelse f.eks. SWOT, risiko- og interessentanalyser osv.*

## **11.5 Virksomhedsanalyse**

*Indsæt jeres virksomhedsanalyser med i dette afsnit, hvis I har lavet et projekt i samarbejde med en virksomhed f.eks. Mintzbergs 5 former, Mintzbergs 7 kræfte*r, PEST *osv.*

**12. Konklusion**

*Her skal der svares på problemformuleringen. Hvad fandt man ud af, blev problemformuleringen besvaret eller fejlede det hele? Konklusionen og problemformuleringen binder rapporten sammen.*

*Der skal ikke stå noget om, hvordan selve forløbet har været, f.eks. at man synes, det var sjovt, at ens makker ikke lavede noget eller at man ikke havde nok tid.*

*Konklusionen gentager problemformuleringens spørgsmål og kommer med et fuldt dækkende og velbegrundet svar. Det kan gøres ved at opsummere eventuelle delkonklusioner, der er udarbejdet undervejs, og sammenfatte dem i en fyldig opsamling af projektet*

***Må max fylde 1 side***

# 

# **13. Projektforløbet**

*Her beskrives samarbejde og kommunikation under projektet for projektgruppen.*

*Der må gerne reflekteres over, hvad der kan forbedres til næste gang.*

***Må max fylde 1 side***

**14. Perspektivering**

*Dette afsnit tillader refleksion over hvilke fremtidige muligheder projektet har. I kan tage inspiration i følgende spørgsmål:*

* *Hvad vil være næste skridt i udviklingsprocessen*
* *Ideer til hvordan kan jeres løsning bidrage til at gøre verdenen bedre?*
* *Ideer til hvordan kan man udbygge løsningen fremadrettet?*

*Der kan også peges på,* ***hvad*** *der fremover kan gøres* ***eller*** *undersøges i forhold til emnet.*

***Må max fylde 1 side.***

**15. Litteraturliste**

*I denne liste angives nummereret alle dine kilder, der sædvanligt deles op i litteratur (bøger, tidsskrifter og aviser), kilder fra internettet og personrelaterede kilder. Vigtigst er at de skal opstille i alfabetisk rækkefølge De nævnte kilder angives på forskellig vis, se eksempler nedenfor:*

***Bog:*** *Forfatterens efternavn, fornavn: bogens titel (i kursiv), forlagets hjemsted, forlagets navn samt udgivelsesår, eksempelvis:*

*[1] Kurose & Ross: Computer Networking: A Top-Down Approach, 7th ed., Harlow, Pearson, 2017.*

***Artikel i tidsskrift:*** *Forfatterens efternavn, fornavn: Artiklens titel (i citationstegn), tidsskriftets navn (i kursiv) efterfulgt af nummer, år samt sidetal til sidst, eksempelvis:*

*[2] Jensen, Hans: "Romerrigets fald", Antikke Studier, nr. 2, 1993, s. 14-21.*

***Avisartikel:*** *Forfatterens efternavn, fornavn: Artiklens titel (i citationstegn), avisens titel (i kursiv), dato for udgivelsen, eksempelvis:*

*[3] Petersen, Albert: ”Koks i senmoderniteten”, Politiken 15.12 2009.*

***Artikel på internettet (hvor et forfatternavn ikke kan findes):*** *Titel på webartiklen (i citationstegn), den fulde webadresse samt dato for besøget på siden, eksempelvis:*

*[4] ”Rainbow Tables”, https://en.wikipedia.org/wiki/Rainbow\_tables (2018-09-04).*

***Artikel på internettet (hvor et forfatternavn kan findes):*** *Forfatterens efternavn, fornavn: Titel på webartiklen (i citationstegn), den fulde webadresse samt dato for besøget på siden, eksempelvis:*

*[5] Schneier, Bruce: ”The New Way Your Computer Can Be Attacked”, https://www.schneier.com/essays/archives/2018/01/the\_new\_way\_your\_com.html (2018-01-22).*

*Bemærk at Google ikke er en kilde*

**16. Bilag**

*Alt, hvad der ikke direkte hører til i rapporten, skal heri. Det kan være brugerundersøgelser, testresultater, testopstillinger, diagrammer m.m.*

*Første side i bilag skal være en nummeret bilagsoversigt.*

*Der* ***skal*** *refereres til det respektive bilag i selve rapporten.*

*Husk at give hvert bilag det nummer, for eksempel: Bilag 16.1*

*Hvis der er krav om Business Case i projektet, skal den være i bilag.*

***Dette skal I huske i hele rapporten – Skal slettes i jeres rapport inden aflevering***

***Alt tekst i kursiv skal slettes***

***Linjeafstand***

*Tekst i dokument skal have linjeafstand på 1,15*

***Plagiering***

***Du må ikke plagiere andres arbejde (gælder også eget tidligere bedømt arbejde).***

*Hvis eget tidligere bedømt arbejde bruges, så referér til dig selv, ellers er dette også plagiat.*

*Du må derfor som hovedregel ikke kopiere fra andre kilder! Du skal benytte dine egne ord. Din rapport skal være dit arbejde.*

*Den eneste tilladte kopiering er korte citater, som klart skal markeres: "Sådan markerer du et kort citat." Du skal kun citere, når det er absolut relevant, ellers omskrives essensen af de oplysninger, du har brug for. Der skal refereres til ophavsmanden/kilden til citatet, enten som en fodnote eller i referencesektionen.*

*I hele din rapport skal du angive, hvor en information stammer fra. Hovedformålet med at citere kilder i rapporten er at give læsere en reference, som kan give yderligere oplysninger om et bestemt emne.*

***Kildehenvisninger***

*Kildehenvisninger kan udformes på to måder.*

* *Via parenteser i selve opgaveteksten, der henviser til en nummereret litteraturliste. Eksempelvis: Spectre- og Meltdown-angrebene angriber den såkaldte “speculative execution” feature, der benyttes i moderne processorer for at optimere ydelsen [5]. Glemmer man at kryptere sine kodeord med salt, kan en angriber benytte rainbow tables til at knække kodeordene på kort tid [4].*
* *Via fodnoter: Placeres nederst på siden. Fodnoter kan indeholde kortere uddybninger af kildehenvisninger. En kildeangivelse i fodnoten skal kunne genfindes i litteraturlisten.*

*Se hvordan en litteraturliste udarbejdes i afsnit 15*

***Figurer: Tabeller, illustrationer, billeder og grafer***

*Figurer skal have nummereret forklarende tekst, og der skal refereres til hver figur i teksten. En figur uden reference i teksten skal fjernes. Eksempelvis:*

*Kravene for installation af en Windows 2008 server ses i figur 1. Som det ses af figur 2, er hele skoven styret af en enkelt domænecontroller.*

**Eksempel på nummererede billedtekster:**

*Figur 1: Viser kravene til at anvende Windows Server 2008*

*Figur 2: Lokation af Active Directory domain controller i et network domain*

***Forkortelser***

*Hvis du bruger mere end ti forkortelser i din rapport, skal du lave en liste, der forklarer dem og vedhæfte listen i dine bilag. Hvis du bruger mindre end ti forkortelser, skal du skrive hver forkortelses betydning fuldt ud, første gang du bruger den.*

***Konsistens***

*Din rapport skal se ens ud hele vejen igennem! Sørg for, at dine overskrifter, mellemrum, punktopstillinger og indekseringer er ens i hele rapporten.*

***Upload program***

*Hele programmet der laves skal uploades i WiseFlow i en zip fil. Ligeledes skal programmet uploades til Github med link til det i bilag. Hele programmet skal også vedlægges i bilag som screenshots eller tekst med linjenumre og filnavne. Husk at fjerne alt kode som er udkommenteret eller på anden vis ikke er relevant til eksamen.*