Iteration 1:

Hvad skal vi fremstille

En liste over de emner/begreber fra faget Teknologi I anvender i jeres løsning med en kort beskrivende kommentar. Listen skal opbevares i jeres github-repository.

Emner/begreber

- **IP-adresse** - (Internet protokol adresse) Vores hjemmeside bliver hostet på en logisk IP-adresse, og den enhed vi bruger har også en unik IP-adresse
- **Port** - For at komme ind på vores hjemmeside skal du igennem port 80
- **OSI** - Open Systems Interconnection er en syv lags model der bruges til at beskrive funktioner for et netværks system. Hvert lag har specifikke funktioner at udføre
- **TCP/IP** - TCP/IP er meget det samme som OSI modellen. TCP/IP modellen er en sammenpresset OSI model, i den forstand, at nogle af lagene er sat sammen til et, derfor har TCP/IP modellen kun 4 lag, og ikke syv. De fire lag beskrive dog de sammen funktioner som lagene i OSI modellen. OSI og TCP/IP modellen beskriver altså hvordan information bevæger sig gennem et netværk f.eks. til vores hjemmeside
- **LAN/ WLAN/ MAN/ WAN** - Forskellige typer af netværker, hvor LAN og WAN er nok de mest udbredte. Local area network er en gruppe af computerer, som er forbundet til hinanden i et mindre område eks. et kontor. Wide area network er i bund og grund internettet. For at komme på vores hjemmeside skal vores enhed have adgang til et netværk
- **Router** - Via en router er det muligt at komme på internettet så man kan komme ind på vores hjemmeside. En router sender information mellem to eller flere computer netværker
- **Mac-adresse** - Den fysiske adresse på en enhed
- **Learning** - Når noget bliver sendt til switchen lære den MAC-adressen
- **Flooding** - Sender packet til alle andre end afsenderen af dataen, for at finde en manglende MAC-adresse i switchens MAC-address table
- **Forwarding** - Der er forskellige måder at forward en frame på.
    - Store and forward: Switchen kopier framet og tjekker den igennem efter fejl før den sender framet videre. Tage længst tid.
    - Cut-Through: Switchen ser på MAC-Addressen og sender den bare videre uden at tjekke om fejl.
    - Fragment Free: En blanding af de ovenstående to.Tjekker kun de første 64 bytes igennem.
- **Filtering** - Sker oftest når der skal floodes, betyder at den sender til alle andre MAC-adresser end den, som framet kom fra.
- **ARP** - Henter MAC-adressen gennem netværket
- **ARP-tabel** - Holder styr på alle IP- og MAC-adresser samt hvordan de er forbundet

Iteration 2:

Hvad vi skal fremstille

Listen over de emner/begreber fra faget Teknologi I anvender i jeres løsning skal opdateres.

Uddyb forklaringen af de(n) protokol(ler) der bliver anvendt via jeres løsning (direkte eller indirekte)

Emner/begreber/protokoller

- **HTTP** - Hyper Text Transfer Protocol, bliver primært brugt til kommunikation på internettet. HTTP er en client-server protocol, hvilket vil sige at request bliver sendt af en enhed som oftest vil være en Web-browser. Hver request vil blive sendt til en server som vil lave en respons. HTTP bruger TCP.
- **HTTPS** - Hyper Text Transfer Protocol Secure, det er en sikker version af HTTP. Det er en protokol som sikre kommunikationen mellem Web-browser og website. HTTPS er encrypted for at øge sikkerheden for data transfer.
- **DNS** - Domain Name System, bruger UDP**.** Den laver navnene fra hjemmesider om til IP adresser, eksemple på DNS - nslookup google.com - 216.58.211.110
- **TCP** - Transmission control protocol, er en type internet protokol (IP). TCP er “connection oriented”, som betyder at før data kan blive sendt, skal der være forbindelse mellem to hosts, også kaldet “3-way-handshake”. TCP passer bedst til applikationer der kræver stor pålidelighed, og transmissions tiden er mindre vigtig. TCP arrangere datapakkerne i en bestemt rækkefølge.
- **UDP** - User datagram protocol eller Universal datagram protocol, er en type internet protokol (IP). UDP er en “connection less” protokol, som betyder at der ikke er behov for, forbindelse for at sende data. UDP passer bedst til applikationer, der har behov for en hurtig og effektiv transmission f.eks. spil. UDP arrangere ikke datapakker i en bestemt rækkefølge. Alle pakker er uafhængig af hinanden.

Iteration 3:

Hvad vi skal fremstille

Listen over de emner/begreber fra faget Teknologi I anvender i jeres løsning skal opdateres.

Giv en kort beskrivelse af hvorledes jeres MVC løsning kunne implementeres på flere (adskilte) servere. I skal altså ikke (nødvendigvis) beskrive den aktuelle implementering, men fremstille en tænkt løsning, hvor forskellige elementer af jeres løsning er placeret

(implementeret) på flere individuelle servere der er fysisk helt adskilte. Overvej hvilke fordele og hvilke ulemper det kan have ud fra et rent netværksperspektiv.

Emner/begreber/protokoller

- **Asynchronous**
- **Asynchronous unit testing**
- **Signaling**
- **Threading**
- **Thread safety**
- **Thread pool**

MVC

Fordele:
- Noget!

Ulemper:
- Intet!

Iteration 4:

Hvad skal vi fremstille

Listen over de emner/begreber fra faget Teknologi I anvender i jeres løsning skal opdateres.

Giv en kort beskrivelse af hvilke sikkerhedsmæssige problemstillinger der skal overvejes ifm. jeres løsning.

Vurder, for hver af disse problemstillinger, hvad I kan gøre for at sikkerheden for jeres løsning er så høj som man med rimelighed kan forlange

Emner/begreber

- **SQL Injection**
  SQL injection er et angreb rettet mod databaselaget i en applikation, ved at *indskyde* fjendtlig SQL-kode i et SQL-kald. Angrebet udnytter en sårbarhed i håndteringen af brugerinput og databasekald. Hvis brugerens input ikke renses for specielle tegn og sætninger, kan applikationens databasekald blive manipuleret til at få en anden effekt end den ønskede
- **Hashing**

Hashing er en one-way metode til at lave en besked om til en besked som ligner vrøvl. Det skal være med svært at dehash. (Det aldrig umulig at dehash). Vores password gemt på vores hjemmeside bliver hashed

- **Kryptering**
  Kryptering er ligesom hashing udover at det er ikke one-way. De er en dekrypteringsnøgle man kan bruge til dekrypter en krypteret besked. SSL sker gennem kryptering og dekryptering.
- **Salt**
  Salt er en unikt ikke secret værdi som bliver gemt i databasen med brugeren. Salt bliver brugt til gøre hashing mere kryptisk. Man smider den saltet værdi på en besked som skal bliver hashed før den bliver hashed. Når den nu bliver hashed så er den hashed besked mere besværlige at dehash ved fx rainbow table. Vores identity area som vi har lavet via entity framework bruger salt.
- **Pepper**
  Pepper er lidt ligesom salt i det at den bliver smidt på en besked før den bliver hashed. Forskellen er at pepper værdien er ikke unikt og den skal være secret.
- **Properties af hashing**
  Der er 3 vigtige properties som en hashing algoritme skal opfylde før den kan blive kaldt en hashing algoritme:
  - Pre-image resistance
    Pre-image resistance siger bare at ud fra en hash skal det være svært at dehash den.
  - Second pre-image resistance
    Second pre-imrage resistence siger at hvis vi kender en besked1 som ikke er hashed1.Når den bliver hashed1, og der er en anden som vi ikke kender besked2 med samme hash2, så kan vi ikke finde ud af hvad besked2 er ud fra de to hash som er det samme.
  - Collison resistance:
    samme som pre-image resistence men man bestemmer selv hvilket besked man kender til.
- **Owasp**
  Open Web Application Security Project er et online community, der producerer frit tilgængelige artikler, metoder, dokumentation, værktøjer og teknologier inden for web applikationssikkerhed
- **Blind SQL injection**
  Blind SQL injection bliver brugt når vi ikke får noget output eller error fra web applikationen. Det er en slags SQL injection der stiller sande eller falske spørgsmål til databasen, for at bestemme svaret baseret på applikationens svar
- **Rainbow table**
  Det er en forud beregnet tabel som bliver brugt til at omregne hashed passwords

Sikkerhedsmæssige problemstillinger

- Vores database skal være sikret mod SQL injection
  Der er flere ting man kan gøre for at øge sikkerheden mod SQL injection. Nogle af dem er bl.a.

    - Parameterized Statements:
      Det gøre vores entity framework for os.
    - Object Relational Mapping:
      Når man bruger ORM betyder det at man sjældent kommer til at skrive SQL statements, hvilket betyder at du er sikret mod SQL injections
    - Escaping Inputs:
      Man sikre at man kan undslippe at skrive specialtegn i input parametrene
    - Sanitizing Inputs:
      Man forbyder brugen af specielle/mistænkelige inputs som f.eks. ‘ OR 1=1;-- osv.

- Vores administratorer/bruger passwords skal være sikret
  Der er flere måder man kan sikre passwords på, dette kunne være ved at bruge enten hashing eller kryptering (se emner)

Iteration 5:

Hvad skal vi fremstille

Listen over de emner/begreber fra faget Teknologi I anvender i jeres løsning skal opdateres.

Uddyb den korte beskrivelse af hvilke sikkerhedsmæssige problemstillinger der skal overvejes ifm. jeres løsning.

Vurder, for hver af disse problemstillinger, hvad I kan gøre for at sikkerheden for jeres løsning er så høj som man med rimelighed kan forlange.

Emner/begreber

- **XSS**
  Cross-side scripting er en klientside kode “injection attack”. Det er når en hacker “injecter” ondsindet kode ind på en hjemmeside som offeret besøger. Når offeret så besøger hjemmesiden vil den ondsindet kode blive executed.
- **Kryptering**

Er processen, at gøre dine data ulæselige for uvedkommende. Der er flere måder man kan kryptere på, nogle eksempler kunne være “Cæsar” eller “Polyalphabetic”

- **Dekryptering**
  Er processen, at gøre den krypterede data læselig igen. Afhængigt af hvilken slags kryptering der er blevet brugt, kræver det en slags nøgle, for at dekryptere din data

Sikkerhedsmæssige problemstillinger

- Vores projekt skal være sikret mod XSS angreb
  Der er flere ting, der kan gøres, for at øge sikkerheden mod XSS (Cross-side scripting) angreb. Nogle eksempler kunne være.
  - Filtering for XSS:
    I denne løsning vil en web udvikler implementere et filter, som al udefrakommende data skal igennem, som så filtrerer farlige “keywords”. Dette kunne f.eks. være javascript kommandoer eller lignende
  - Escaping from XSS:
    Når vi er “escaping” fortæller vi vores web browser, at det data vi sender, skal blive behandlet som data, og ikke fortolkes på nogen anden måde. Problemet med denne løsning, er at du kan ikke bare “escape” alt, for så vil ens egne scripts og HTML ikke virke.

- Vi skal beskytte vores brugers data
  I sidste iteration snakkede vi om hvordan man bruger hashing til at sikre passwords. Hvis vi nu skulle sende noget andet personlig data over internettet, hvordan sikrer vi så at det ikke bliver læst? Dette kunne vi gøre ved at kryptere det data vi sender, så hvis nogle andre skulle få fat på data’en, kan de ikke læse den.

Iteration 6:

Hvad skal vi fremstille

Listen over de emner/begreber fra faget Teknologi I anvender i jeres løsning skal opdateres.

Uddyb den korte beskrivelse af hvilke sikkerhedsmæssige problemstillinger der skal overvejes ifm. jeres løsning.

Tidligere (iteration 3) havde I overvejelser vedr. hvordan jeres løsning kunne implementeres modulært på flere servere. Nu skal I forsøge at beskrive hvordan virtualisering kunne tænkes anvendt ifm. jeres løsning.

Emner/begreber

- **Virtualisering**

  Virtualisering er processen at lave virtuelle, ikke virkelige, versioner af noget. Dette kunne være et operativsystem.

- **Virtuel maskine (VM)**
  En virtuel maskine er et operativsystem der kører på en fysisk enhed eller “host” ved hjælp af en hypervisor

- **Hypervisor**
  En hypervisor er et ”software layer”, som koordinerer VM’er. Den fungerer som et interface mellem VM’er og ”bare metal” (computerens) ressourcer. Den sørger for at hver VM, har de ressourcer, de skal bruge for at køre. Hypervisor holder også hver VM adskilt fra hinanden.

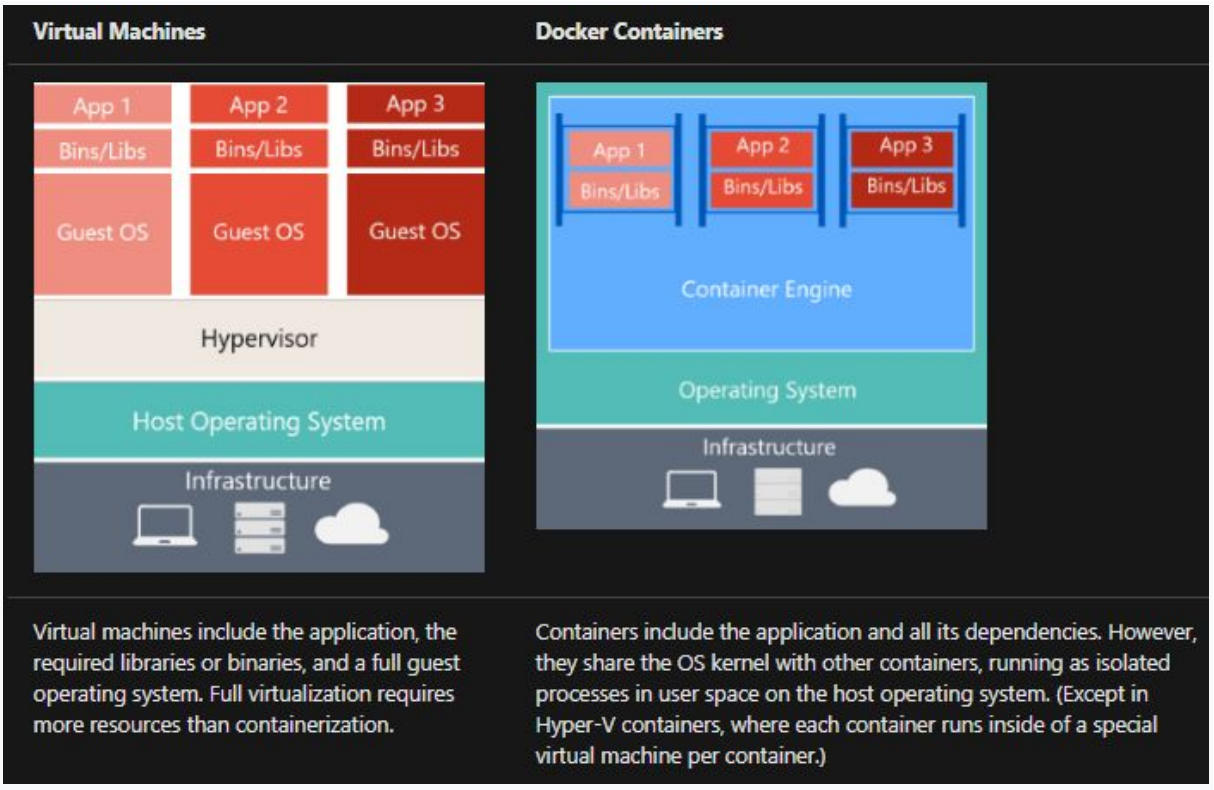
  Type 1:
  Type 1 kører direkte på den fysiske hardware, som regel en server, og tager pladsen som OS.

  Type 2:
  Type 2 kører som en applikation på en eksisterende OS, og går som regel efter enkelt bruger ”desktop” eller ”notebook” platformer.

- **Docker**
  Docker er et open-source-projekt, som automatiserer udviklingen af applikationer som bærbare, selvforsynende containere, som kan køre i “cloud” eller “on-premise”

  Docker container:
  En container er en standardenhed af software, der pakker kode og alle dens “dependencies”, så applikationen kører hurtigt og pålideligt fra et computermiljø til et andet

Forskellen på VM og Docker:



Virtualisering

Vi mener man kunne bruge VM'er, der hver kører på forskellige operativsystemer, til en alternativ måde at "teste" om vores projekt, fungere korrekt på andre slags OS'er, end vores nuværende.