

Rapport Grundforløbsprojekt

Forside:
Navn: Gabriel Visby Søgaard Ganderup (lavet sammen med Jakob Lykke Lyngsøe) Klasse: Data GF2 Skole: Hansenberg Emne: Data- og Kommunikation Titel: FlipTheBurger Projekt Fokusområde: Programmering Vejleder(e): MiJo - Michael Jørgensen Afleveret d. 15-12-2023 Underskrift:
Problemformulering:
Dette projekt fokuserer på at optimere IT-ressourcerne i den lokale fastfoodrestaurant, FlipTheBurger, for at forbedre ordrehåndtering, netværks effektivitet, PC-teknik, programmering og sikkerhed for at øge servicekvaliteten, kundeoplevelsen og produktiviteten.
Delproblemstillinger / underspørgsmål:
<ol style="list-style-type: none">1. Analyse af det eksisterende netværkslayout og implementering af et optimeret netværksdesign med fokus på stabilitet, sikkerhed (f.eks., kryptering og firewall) og skalérbarhed.2. Evaluering af nuværende hardware samt mulige erstatninger for at forbedre ydeevne, kapacitet og implementering af en effektiv ressourcestyring teknik for at sikre optimal brug af IT-ressourcerne.3. Konceptualisere et produkt vha. Python, som vil kunne blive brugt som et ordre-display system, til at optimere forretningens ordrehåndtering4. Analyse og risikovurdering af de nye implementeringer hos FlipTheBurger og sammenlign med dets tidligere infrastruktur.
Proces /tidsplan

Indholdsfortegnelse

Introduktion og kravspecifikationer.....	3
Fysisk Tegning.....	4
Logisk Tegning.....	4
Adresse Skema.....	4
Risikovurdering.....	5
Udstyrsliste.....	6
Budget.....	7
Programmering.....	7
Projektstyring.....	9
Konklusion.....	10

Introduktion og kravspecifikationer

Virksomheden FlipTheBurger søger bedre løsninger indenfor IT. De er interesserede i at digitalisere deres ordresystem, hvor der vil være et program, de ansatte vil kunne anvende til at se kunders ordrer, som vil anvendes på et display i køkkenet og ved kassen.

Det nuværende netværk hos FlipTheBurger består af et abonnementsbaseret mobilnetværk. Et af klagerne hos forretningen var at netværket ville blive langsommere senere på måneden og derfor forsinkede forretningens produktivitet og effektivitet.

Der er en del mulige alternativer til at forbedre dette, men først er vi nødt til at vurdere, hvilke typer af **internet** FlipTheBurger har til rådighed. Da vi inspicerede butikken for typer af stik, var det eneste vi fandt, et modem som en mulighed. Uheldigvis fandt vi frem til at dette stik ikke virkede, vi kunne derfor ikke bruge det. Så nu har vi været nødt til at se, om der var andre muligheder.

Vi faldt over [EWI](#) Fibernet som en mulighed, de er en del af Koncernen under kommerciel infrastruktur, og de har billige og effektive tilbud. Vi har her valgt 300/300 Mbit/s hastighed, da forretningen vil have gæstenetværk til kunder, og vi finder dette tilbud, som står til 99 DKK i 6 måneder, herefter 319 DKK i måneden bedst. Herefter kan man bedømme, om denne hastighed er passende til FlipTheBurger.

Til konfigurationen af det nye netværk, vil vi oprette 2 VLANs, én til privat og den anden til gæstenetværk. Da der er behandling af kundedata, laves der selvfølgelig sikkerhed på netværket, dette kan være WPA2 Password encryption. Udover dette, subnettes netværket til /25, for at gøre forbindelsen mest sikker. Derudover ville der også være mulighed for at implementere ACL (Access Control List) for at opsætte regler som f.eks. at AP1 og AP2 ikke ville kunne kommunikere med hinanden, samt muligheden for at sætte regler for download osv. på netværket på f.eks. GPO.

Til dette projekt skal der opsættes Router og Switch samt 2 Access Points. De 2 access points vil operere i et opdelt netværk, én til privat/virksomhed og én til kunder og gæster. Derudover opsættes der 1 bærbar, 1 tablet, 1 TV, 1 Zetlreader som skal forbindes til netværket, samt opsættes med sikker brugerstyring.

Derudover vil vi lave en analyse samt risikovurdering af det nye netværk og system vi har fremvist, samt sammenligne det med virksomhedens tidligere infrastruktur og system.

Fysisk Tegning

- Se Bilag 1 Fysisk Tegning

Den fysiske tegning har vi selv lavet, da vi allerede kendte til virksomheden, samt deres ønsker til de nye implementeringer af devices der bliver indført.

I Restaurantområdet kan man på bilaget se eksempler på devices som vil blive anvendt af kunder. Kunder vil kunne oprette forbindelse til gæsternetværket, som er på AP2. Der vil være WPA2 på dette netværk for at sørge for at netværket kun anvendes af kunder på restauranten.

Inde i køkkenet vil der være et TV som vil bruges af virksomheden til display af kundeordrer gennem en Miracast tilkoblet en tablet. I køkkenet vil Wiring Closet også være, som indeholder Router, Switch, AP1 & AP2. Positionen af Access Points vil ikke have betydning, da deres rækkevidde dækker hele bygningen. Derfor vil det ikke være nødvendigt at have dem andetsteds end samlet med andre netværk devices.

Logisk Tegning

- Se bilag 2 Logisk Tegning

På bilaget vil man se den logiske tegning udført i Cisco Packet Tracer. Her kan man se opsætningen på netværket med de to APs opdelt med subnetting og Vlan. Derudover vil man se subinterfacerne Gig0/1.10 og Gig0/1.20 som går til Switch og derefter til Vlan 10 og 20. Vi har lavet Vlan for at splitte Virksomhedens Private og Gæste netværker op, dette vil forbedre logning af aktivitet på netværkene samt sikkerhed.

Adresse Skema

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	Gig0/1.10	192.168.10.1	255.255.255.128	N/A
	Gig0/1.20	192.168.10.129	255.255.255.128	N/A
SW1	VLAN 10	192.168.10.2	255.255.255.128	192.168.10.1
SW1	VLAN 20	192.168.10.130	255.255.255.128	192.168.10.129
A-P1	NIC		255.255.255.128	
A-P2	NIC		255.255.255.128	
Laptop1	NIC		255.255.255.128	
Tablet	NIC		255.255.255.128	
Printer	NIC		255.255.255.128	
Smartphone0	NIC		255.255.255.128	
?	NIC		255.255.255.128	

Subnet	Start Address	End Address	Network Address	Broadcast Address
192.168.10.0/25	192.168.10.1	192.168.10.126	192.168.10.0	192.168.10.127
192.168.10.128/25	192.168.10.129	192.168.10.254	192.168.10.128	192.168.10.255

Risikovurdering

Når man skal opsætte et netværk, er sikkerhed altid utroligt vigtigt. Cybersecurity bliver mere populært og grunden til det er selvfølgelig øgende hacker tilfælde. Intet netværk er helt sikkert, men at bygge et stærkt netværk med store og flertallige mure vil klart mindske muligheden for brud. Her vil vi forklare de steps vi har taget, for at opsætte et sikkert, dog budget-realistisk netværk.

I forhold til antivirus vil der blive anvendt Windows Defender. Disse devices skal ikke anvendes til andet end restaurant, ordresystem og betaling. Der vil også blive opsat brugerrettigheder på disse devices for at sørge for at de kun bliver anvendt til det ønskede. Firewall vil også blive vedligeholdt af Windows Defenders Firewall. Derudover vil der selvfølgelig være login til de forskellige enheder med et stærkt password.

Skulle der ønskes et program til virus scanning, vil vi her anbefale Malwarebytes. Windows Defender vil også kunne dette, men vi finder Malwarebytes bedre til at finde nyere samt mere avancerede former for virus. Malwarebytes er gratis så længe det kun anvendes til Virusscanning, hvilket er det eneste, det vil blive anvendt til.

Skulle det ønskes ville der også være mulighed for opsætning af automatisk backup, hvor der laves sikkerhedskopier af computeren på et drev, skulle der ske en kritisk systemfejl. Fjernskrivebord ville ikke være nødvendigt, da virksomheden samt systemet drives i person.

I forhold til sikkerhed med kundeordrer, altså data breach, som bestilles online, ville dette ikke omhandle FTB. Online kundeordrer bliver behandlet og foregår hos [Mealo Aps](#), som står for online bestillinger, gennem deres egen server og database.

Udstyrsliste

Det nuværende hardware som FlipTheBurger benytter sig af består af:

- Zettle Reader 2
- Zettle Dock 2
- Receipt Printer Standard
- iPad 10,2” (2019)
- [Samsung Galaxy Tablet](#)

Fordi vi har så få devices, er der ikke mange ændringer vi ville lave. Vi vil mere lave tilføjelser til det. Vi ville gerne have en bærbar i stedet for en iPad, fordi vi mener det ville være mere robust og det ville gøre det lettere at samarbejde mellem den og tabletten, når vi skal skabe en forbindelse med vores ordre-display.

Hvis vi skulle have applikationen over på en Apple-enhed, og den skulle kunne fungere med Android enheder, ville der være en række af udfordringer vi er nødt til at tage stilling til. Her er nogle af de potentielle problemer vi ville kunne støde ind i.

Android er primært udviklet i Java ved hjælp af Android Studio, mens iOS er udviklet med Swift eller Objective-C ved brug af Xcode. Fordi de ville skulle bruge forskellige sprog og udviklingsmiljøer ville det kræve separate kode til enhederne. Et andet problem vi ville være nødt til at tage stilling til er UI/UX variationer. Android følger Material Design-principper, som er et design-system udviklet af Google. Mens iOS følger Human Interface Guidelines. Disse retningslinjer giver udviklere anbefalinger til hvordan de skaber en konsistent brugeroplevelse. Det tredje, og i vores optik største problem er licensbegrænsninger. iOS er mere restriktivt og giver os færre muligheder for at arbejde på vores app, mens Android er open-source og tilpasningsdygtig.

FlipTheBurger benytter sig af Zettle, som deres POS(Point of Service) med både Zettles software og hardware. Zettles kortlæsere overholder PCI PTS¹, hvilket er derfor vi har valgt at lade være med at lave nogle ændringer på denne front.

Vi vil gerne tilføje et TV og Mirascreen Dongle til butikken, fordi vi vil have brug for dette for at fremvise vores applikation. Tv'ets primære funktion ville være til at streame gennem Chromecast, tablettens skærm, så alt information vil være til rådighed for butikkens medarbejdere. Vi havde overvejet, om der måske kunne være en simplere metode, hvor vi skulle bruge færre devices. En metode vi fandt var at vælge et Samsung Smart TV, hvilket havde en indbygget skærm delingsfunktion, dette ville være optimal da butikken allerede har en Samsung tablet der

¹ PCI PTS står for “Payment Card Industry PIN Transaction Security”, hvilket er et sæt af sikkerhedsstandarder. Disse standarder fokuserer på at sikre hardware- og softwarekomponenter involveret i PIN-baserede betalingstransaktioner.

HB-data

understøtter den funktion. Men da vi undersøgte valgmulighederne for et Smart TV med denne funktion, fandt vi frem til at priserne for disse tilføjelser ville være for dyre. Derfor har vi besluttet os for at bruge Dongle og et billigere TV for denne funktion. Herunder har vi vores forslag:

Mirascreen G2 HDMI Dongle, 199 kr.



Phillips 24" LED TV, 1777 kr.



Budget

Vi har lavet et budget over de tilføjelser vi har planlagt, som kan ses herunder:

Type:	Navn	Pris i DKK ²
TV	Phillip HD Ready LED TV	kr 1.777,00
Bærbar	Lenovo IdeaPad 1 Cel	kr 1.617,00
Router	IGW 3000 P2P	kr 0,00
Internet	Ewii	kr 319,00
Switch	TL-SG2210P V1	kr 921,00
Accespoint	Ubiquiti UniFi x 2	kr 1.332,00
Skærmdeling	Mirascreen G2 HDMI Dongle	kr 199,00
Total pris		kr 6.165,00

Vores prioritet har været at gøre det så billigt så muligt for FlipTheBurger at skulle lave justeringer eller forbedringer til butikken og arbejdsflowet. Her foroven vises det, at det faktisk ikke er en helt vildt dyr opgave. Men alle disse tilføjelser er foretaget uden at tælle eventuelt el-forbrug og betaling af opsætningen af de nye implementeringer. (*Prisen kan måske variere*)

Programmering

Det første trin, vi foretog os, var at identificere hvilke metoder, der var nødvendige for at oprette et program, der kunne håndtere de modtagne input og foretage passende handlinger.

² Priserne kan ændre sig, da nogle af dem blev valgt med tilbudspris

HB-data

For at gøre dette, lavede vi en skitse af hvordan vores program skulle operere (se bilag 3). I denne proces fandt vi frem til at vi ville have et program som ville kunne sortere mellem de forskellige faser af ordrene restauranten ville modtage. I bilaget kan man også se at vi planlagde at lave en database, som skulle opbevare alle ordrene. Vi havde snakket om at skulle SQLite, som vores database. Men da vi begyndte på processen, fandt vi frem til at vi ville have for lidt tid til at opsætte den.

Til at starte med lavede vi et program, som ville benytte sig af terminalen som sin UI. I ville kunne se dette forslag under bilag 4, app.py.

OrderManager - klasse

`__init__(self):` er indledningen for OrderManager-klassen. Den initialiserer 3 lister: `ordered`, `prep` og en deque ved navn `ready`, som har en maks. længde på 10. De viser de forskellige stadier af ordrenes behandling. `display_orders(self)` udskriver indholdet af vores lister til konsollen. `move_to_GENSTAND(self, item)` her flytter vi på en specifik ordre og sender den til en ny lokation. `move_all_to_LOKATION(self)` her flytter vi på alle ordrene og sender dem til en ny lokation. Hvis antallet af ordre i `ready` overstiger 10, fjernes de ældste ordre. `add_order(self, item)` Her tilføjes ordre til `ordered`.

main() Funktion

```
while True:
    order_manager.display_orders()

    print("\nOptions:")
    print("1. Add Order")
    print("2. Move to Prep (type 'Z')")
    print("3. Move all to Prep (type 'ZZ')")
    print("4. Move to Ready (type 'X')")
    print("5. Move all to Ready (type 'XX')")
    print("6. Exit (type 'sudoluk')")
```

Her startes en uendelig løkke, hvor vi kan vælge forskellige handlinger i programmet. Hvis programmet skal lukkes, indtastes der `sudoluk`.

Vores program her er en simpel simulering af ordrebehandlingen restauranten skulle udføre, hvor ordre bevæger sig gennem forskellige faser af forberedelse.

Da vi havde lavet denne simulering besluttede vi os for at tilføje en GUI over den eksisterende kode vha. tkinter modulet i python. (Bilag 4, guy.py)

Vi har brugt tkinter som er standarden for Python. Tkinter understøtter en lang række af Tcl/Tk versioner, som er enten med eller uden trådet support. Det officielle Python bibliotek benytter sig af Tcl/Tk 8.6 trådet. Tcl/Tk er ikke et enkelt bibliotek, men det består af få specifikke moduler, med hver deres egen funktionalitet.

Nu var vi teknisk set færdige med opgaven. Men vi besluttede os for at se om vi kunne simplificere og forbedre koden yderligere. Det forslag vi kom frem til er det som I kan se i bilag 5, app.py.

I denne version af vores kode benytter vi os af boolean return values for sin metode, til at vurdere om en ønsket handling var foretaget. Det gør det muligt for vores kode at vurdere om handlingen var succesrig.

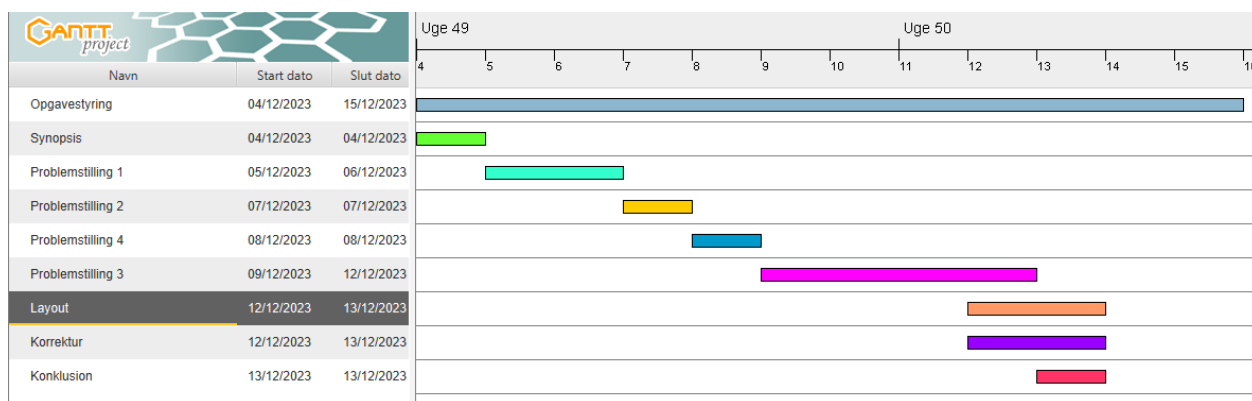
Fordi vi nu står med to forskellige koder, har vi været nødt til at vurdere, hvilken er den bedre, for at finde frem til den vi skulle bruge. Vores første kode har en simplere metode struktur, den er bedre til bruger som foretrækker tydelige handlinger. Vores nyere kode er mere kompakt og kombinerer lignende funktionaliteter ind til en metode, hvilket potentielt er med til at reducere overflødighed samt unødvendig gentagelse af processer.

Så valget mellem disse to koder afhænger mest af ens præference til simplicitet og kompakt. Begge metoder vi har lavet her er begge mulige. Vi har endt med at vælge vores nyere kode, på grund af dens kompakte struktur og dets fordele.

For at forklare vores kode visuelt har vi lavet et klassediagram. Det kan findes under bilag 6.

Nu hvor vi har alt dette på plads, har vi endelig et færdigt program, som I kan prøve af under bilag 7.

Projektstyring



Da vi lagde den plan over hvad vi skulle nå af opgaverne, fandt vi ud af at nogle af opgavernes besvarelse var afhængig af hinanden. Det betyder at vi muligvis skulle have lavet flere af opgaverne parallelt af hinanden den samme dag. Udover dette nåede vi ikke at overholde vores tidsplan, vi gik en dag over vores originale plan. Det er heldigvis ikke et kritisk problem, da den officielle deadline lå på d. 15.12 2023. Men til fremtidige projekter ville det være en god ide at give opgaven en eller flere dage, så vi ikke ville støde ind på det samme problem.

En anden detalje vi skal med til overvejelse, er at vi egentlig havde begyndt at lave projektet helt tilbage i slutningen af november. Fordi vi havde begyndt så tidligt, var vi i stand til at springe nogle af trinene over af vores programmering, og dermed komme med et resultat, som ville være på et højere niveau. For eksempelvis, så er det ikke sikkert at vi ville kunne nå at lave en GUI til vores applikation, på grund af tidsrammerne vi havde sat det for os.

Konklusion

Vi lavede vores egen case, som bestod af at digitalisere FlipTheBurgers ordresystem for at forbedre restaurantens effektivitet, optimere netværket og sikkerhed.

For at optimere nettet havde vi besluttet os for at skifte udbyder og opdele nettet i to netværk vha. VLAN. Ved at opdele det, har vi øget netværkets sikkerhed, men det er ikke fuldstændig sikret. Vi kunne forbedre dets sikkerhed med ACL (Access Control List). Udover det, ville jeg vurdere, at vi har sikret FlipTheBurger nok, til det forbrug restauranten har.

Vi har ikke lavet noget brugerstyring, men der kunne muligvis blive lavet nogle implementationer, ved de devices der er i butikken, for at sikre forretningen mod angreb indefra. For at øge sikkerheden yderligere, kunne man træne restaurantens medarbejdere i angreb gennem “social hacking”.

Da vi endelig begyndte på koden, udforskede vi mange nye teknikker og metoder til vores program. Fordi vi gjorde dette, lærte jeg en hel del om nye metoder jeg ikke havde prøvet før. Jeg mener selv, at jeg har en bredere forståelse for området, end jeg havde da jeg startede.