**Indholdsfortegnelse**

# Abstract

## Project

En lille beskrivelse af hvad der indgår i project forløben, dvs. Inception, elaboration og construction og generel forløb osv.

## Kommunikation brug af github

-omhandlende hvordan vi kommunikere. Hvordan vi snakker sammen som team og hvilke redskaber vi bruger til det.

# Indledning

Dette rapport er skrevet som en del af 2 semester eksamensrapport, og er lavet for virksomheden Café Diamond.

Café Diamond har beliggenhed i Køge. Det er en café, der er specialiseret i vandpibe rygning og ejeren har 10 års erfaring inden for branchen.

Ejeren af Caféen får sine vandpiber fra Egypten som er specielt fremstillet til ekseptionel kvalitet. Udover det importeres kvalitets tobak og kul også fra samme sted i Egypten.

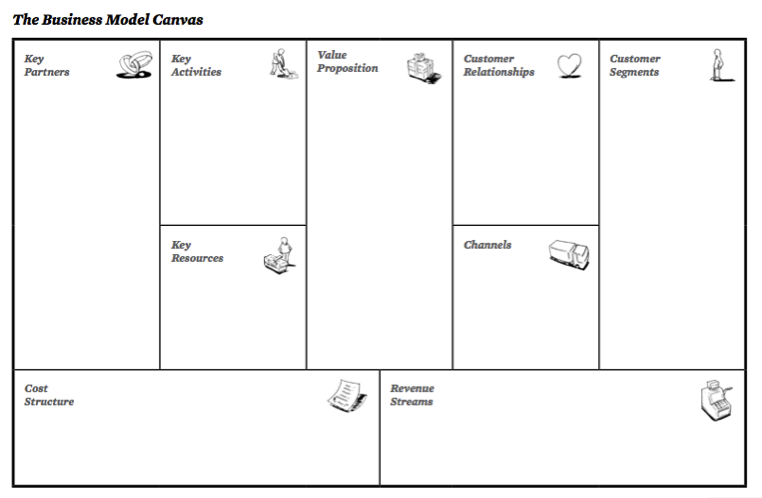
Caféen har åben i hverdagene fra kl.10 – 22, og om weekenden til kl.5 om morgen. Der er 5 medarbejder og ejeren skal ekspandere virksomheden i Københavns området.

Caféer er meget populære i Europa, især i Danmark. Men da det er et marked hvor der er hård konkurrence, kræver det at caféerne bliver nød til adskille sig fra hinanden.

Café Diamond håndtere denne voksende efterspørgsel ved at arrangere forskellige events, hvor man kan komme og se rival opgør i fodbold samt PPV boksekampe m.m.

Da der er kæmpe efterspørgsel, kræver det at man forudbestiller sine siddepladser. Da man kan risikere at caféen kan være helt booket.

På nuværende tidspunkt[[1]](#footnote-1) fungere alt booking ved hjælp af en papirblok og diverse papir lapper. Dette resultere i at medarbejderne kommer til booke et bord som allerede er booket, på den måde kan en gæst komme ind og alligevel ikke få plads siden alle bordene er optaget. Formålet er derfor at udvikle et system, der kan afhjælpe virksomhedens problem med at få overblik, samt håndtere disse bookinger. Sammentidlig hjælpe dem med lette de administrative opgaver.



# Business Model Canvas

Før vi kan hjælpe vores kunde, så skal vi kende dem. Ved brug af Business Canvas, kan vi kortlægge og beskrive det der holder virksomheden sammen.

Business Canvas model beskriver rationalet hvad der ligger bag en virksomhed, og hvordan den skaber og fastholder en værdi – både for sig selv og for kunden.

Før vi kunne lave denne model, så blev vi nød til at analysere vores kunde. Det gjord vi bl.a. ved at bruge følgende værktøjer; SWOT Analyse, Porters 5 forces & Porters Value Chain.

**·** Aldersgruppe: 16 - 60

**·** Gennemsnits alderen: 20-30år.

**·** Vandpiberygere

**· Facebook.**

**· Telefonisk kontakt.**

**· Kunderne bliver mødt personligt i caféen**

**· Langvarige kunder, da man primært har loyale/faste.**

**· Rolig og afslappende atmosfære**

**· Se forskellige PPV arrangementer**

**· Mellemøstlige oplevelse, samt et sted hvor du får en oplevelse.**

**· Markedsføring via Facebook**

**· Følge med i udviklingen og være en skridt foran konkurrenterne.**

**· Events**

**· Sælger vandpibe, drikkelse, mad, dessert.**

**· Vandpibe tobak leverandører**

**· Dagrosa Engros**

**· TDC Internet**

**· Viaplay/Digiturk ( TV**)

[[2]](#footnote-2)

**Faste priser:**

* **Salg af snacks, vandpibe, tilbehør, drikkevare m.m.**
* **Udleje af vandpibe.**

**Faste omkostninger (Husleje, lønninger, Indkøb af vare, ansatte)**

**Variable omkostninger (Arrangere og holde forskellige events)**

**Internationalt brand, så når ud til kunderne via hjemmeside, netværk, via og personligt fremmøde.**

**· Lokale, Lager og Cafe inventar, kvalitets tobak & vandpibe**

**· Engageret medarbejder**

**· Entertainment anlæg**

# Problemformulering

Problemstillingen handler om bookinghåndtering samt administrative opgaver. Der efterspørges en elektronisk løsning til virksomhedens forældet bookingsystem. Ud over det, en måde der kan differentiere virksomheden fra konkurrenterne.

Vi vil forsøge at afhjælpe til denne problemstilling ved at arbejde ud fra en problemformulering. Denne er bygget op omkring følgende hovedspørgsmål:

* Hvordan kan Unified Process (UP), C # -Programming og relationelle databaser anvendes til udvikling og gennemførelse af en lille enkelt bruger it-system?

For at forstå og løse den overordnede problemstilling, vil vi tage fat på følgende del spørgsmål:

* Kan vi lave et system der kan gøre bookinger nemmere og mere overskuelig?
* Hvordan kan vi lave et system, der vil give virksomheden konkurrencemæssige fordele?

# Projektetablering

Hvem er ’ The A-Team’:

Vi er en gruppe på 6 personer som har høj forventninger til os selv, Cemil, Rafi, Erkan, Ramal Daniel og krmanji. Vores formål er at lave en god Applikation til vores projektet. Vores projekt er en applikation til vandpibe ejere, det vil gør det nemmere for ejeren at få overblik over deres System, fordi man kan booke pladser fra vores App og sætte nogle penge ind på vandpibe rygers saldo som de kan bruge i Cafeen.

Vi vil gøre vores bedste for at lave at godt stykke arbejde, som kan tilfredsstille vores kunde. Projektet vil give nye muligheder for Vandpibe ejere. Vi regner med at få høje karakter og derfor vil vi knokle hårdt i dette projekt. Vi er et godt Team som arbejder rigtig godt sammen, vi er gode til at løse problemer og snakker rigtig godt sammen med hinanden.

**Ramal:** Jeg er 23 år, jeg og er i gang med at tag datamatiker uddannelse og jeg synes jeg er bedre til Software Design end Konstruktion.

**Cemil:** Jeg er 22 år jeg er i gang med at tage datamatiker uddannelsen. Jeg synes jeg godt kan finde ud af Software design, og synes jeg skal arbejde lidt mere med konstruktions delen.

**Erkan:** Jeg er 27 år, jeg bor i Solrød Strand og er på nuværende tidspunkt i gang med en datamatiker uddannelse på EASJ og er på 2. semester i øjeblikket. Det er ret spændene at arbejde med at software construction/design projekter hvor vi får et indblik i hvordan det fungere til dagligt i virkelighedens verden.

**Rafi:** Jeg er 27 år. Min fritid bruger jeg gerne sammen med min familie og venner. Jeg nyder gode stunder med min 2 årige gamle søn. Derudover er jeg meget glad for at motionere, da det virker afstressende og giver mig tid til at reflektere over tingene.

**Daniel:** Jeg er 21 år, jeg bor i roskilde og er under udannelse på EASJ som datamatiker, vil gerne på udveksling i kina eller usa. Ud fra egen synsvinkel syntes jeg ikke der er noget jeg har svært ved (endnu), dette er min fordel. Men jeg har meget svært ved at opretholde en normal døgn rytme og sover tit for længe, hvilket resultere i høje fraværs procenter.

**Krmanj**: Jeg er 22 år, født og opvokset i Danmark, hjemland Kurdistan, fritidshobbier fodbold og træning.

**Hvordan skal vores Team være?**

Vi skal have noget disciplin som skal overholdes. F.eks, kom til tiden , Man skal lave det som der blevet sagt. Man skal kunne sammen arbejde, snakke sammen og hjælpe hinanden. Vi skal arbejde sten hårdt og få et godt resultat ud af det.

## Risikoanalyse

Ud fra omtanke er der listet forskellige risici som kan have indflydelse på vores projekt. Da medlemmer af gruppen er nye i feltet kan disse risici være underdrevet i den forstand, at hvis det skulle være et reelt projekt og ikke en læringsproces, ville disse risici være vurderet anderledes. Nogle af disse noteringer er baseret på en mavefornemmelse frem for aktuel data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risici Nr.** | **Sandsynlighed**  **(1-10)** | **Beskrivelse:** | **Konsekvens** | **Impakt** | **Årsag** |
| **1** | **10** | **Manglende kompetence inden for programmering** | **Kvalitet**  **Færdiggørelse** | **M** | **Projektets medlemmer er under uddannelse og har ikke alle elementer under fuld mestring, da det er en læringsproces.** |

**Konsekvens***:* Projektets kvalitet og færdiggørelse er i fare på grund af dette. Da programmerings elementer skal læres først før de kan anvendes betyder det at opgaver og andre ting vil tage længere tid end hvad det realistisk ville tage, samt en forringet kvalitet.

**Mitigations plan:** For at afvige konsekvensen er alle projektets medlemmer med til at afhjælpe dette ved at læse op på tingene hjemme fra, så når vi skal arbejde er der en bedre forståelse for det der skal laves. Dette Resultere i mere produktive sprint perioder.

**Ansvar:** Alle projektets medlemmer.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risici Nr.** | **Sandsynlighed**  **(1-10)** | **Beskrivelse:** | **Konsekvens** | **Impakt** | **Årsag** |
| **2** | **1** | **Syg eller tilskadekommet medlem** | **Kvalitet**  **Færdiggørelse** | **C** | **N/A** |

**Konsekvens***:* Projektets kvalitet og færdiggørelse er i fare på grund af dette. Hvis et medlem bliver syg eller kommer til skade vil arbejdet for den syge/tilskadekommende skulle fordeles ud til resterende medlemmer af projektet. Hvilket resultere i et større pres og forringet kvalitet da opgaver skal laves hurtigere end før.

**Mitigations plan:** Der er ikke lagt nogen Mitigations plan for denne risici, grundet at risikoen ikke er særlig sandsynlig. At et medlem bliver så syg eller kommer så slemt til skade at medlemmet ikke kan hjælpe til hjemme fra. Men vi er heldigvis så mange i gruppen, at dette burde ikke blive et problem.

**Ansvar:** Resterende af projektets medlemmer som ikke bliver syg eller kommer til skade

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risici Nr** | **Sandsynlighed**  **(1-10)** | **Beskrivelse:** | **Konsekvens** | **Impakt** | **Årsag** |
| **3** | **8** | **Tidspres** | **Færdiggørelse**  **Kvalitet** | **H** | **Der kan forekomme tidspres I forbindelse med deadlines hvis opgaver og aftaler ikke bliver overholdt.** |

**Konsekvens***:* Projektets færdiggørelse og kvalitet vil komme I fare hvis dette skete. Da der skal arbejdes hurtigere og mindre grundigt omkring et givent element, vil det derfor give formindsket kvalitet til projektet.

**Mitigations plan:** For at afvige at projektet kommer I tidspres vil vi afhjælpe risikoen ved at holde SCRUM meetings, for at afklare hvor langt vi er og finde eventuelle problemer før det er for sent. Så vi kan fokusere på de ting der kan give problemer som resultere i tidspres senere.

**Ansvar:** Scrum Master.

## Vision Statement

**Problem Statement**

Ud fra vores kundes observationer, viser det sig at man mister potentielle kunder, da man fejlvurdere tidsestimering af bord reservationerne.

Udover ovenstående er det også et kæmpe administrativt tidsforbrug på at modtage bookninger telefonisk og via personlig henvendelse. Medarbejderne og ejeren bruger mange kræfter på at promote de diverse arrangementer som de ønsker. Da det ikke når helt ud til kunderne sikres det ikke at eventen vil blive det værd at arrangere, dette skyldes dårlig kunde relationer[[3]](#footnote-3).

Der er tendens i Café branchen, at der bliver overbooket bord. Dette medfører tab af omsætning og utilfredse kunder.

**Product Position Statement**

Vores Kunde er Cafe Diamond. 2 af deres hoved aktiviteter er således: at booke bord for kunder og arrangere events. Som beskrevet tidligere i ovenstående sektion er der visse ulemper ved den måde disse to aktiviteter bliver udført. For at løse deres ”problem”, vil vi lave et produkt der kan lette arbejdsbyrderne. Sammentidlig vil det medfører mere overskuelighed for dem og deres kunder.

Det produkt vi vil lave til vores kunde er et software program, og ligeledes en mobil applikation som er kompatibel og synkroniseret med software programmet.

Programmet som hedder ’ OVP ’, vil gøre arbejdsprocessen nemmere for vores kunde, og vil give bedre overblik over virksomhedens primær aktiviteter.

Når vi kigger på vores kundens branche, så kan vi se at der er ikke noget lignende produkt på markedet. Det er unikt og revolutionerende inden for branchen. Dette vil medfører at vores kundens virksomhed, vil differencer fra sine konkurrenter.

**Stakeholders beskrivelse**

**Stakeholder Summary**

|  |  |  |
| --- | --- | --- |
| **Navn** | **Beskrivelse** | **Ansvar** |
| Café Ejer (Ibrahim Sabri) | Selvstændig virksomhedsejer, som har 10 års erfaring i branchen. | * Give os de nødvendige data, som vi skal bruge i udviklingen af vores program. * Give os adgang til hans Café så vi kan teste vores program, ved at lave en Validation Test. * Vores kunde, og hans medarbejder laver en Acceptence test. |

**User Environment**

For at gennemfører dette projekt, har vi etableret en gruppe på 6 mænd, hvor arbejdes opgaverne vil blive fordelt via ens kompetencer og ambition.

Vi planlægger fra start vores arbejdsproces og arbejdsgang. Vi fordeler opgaverne og sætter dem systematisk op, og sætter dem i forskellige faser, og realistiske deadlines. Vi har sat vores milepæle op, ude fra vores endelige afleverings deadline.

Da vores kunde bruger Microsoft produkter, vil vi lave et program der kompatibelt med Windows styresystem, både pc og mobil.

**Product Overview**

**Needs and Features:**

**CD = Café Diamond. Bruger = Café Diamonds kunder.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Need** | **Priority** | **Features** | **Planned Release** |
| Log ind function (Admin) | **1** | CD kan logge ind på programmet. | **D.20/4-15** |
| Event, sortiment oversigt | **3** | CD kan redigere events og sortiment og pris. | **D.27/4-15** |
| Booking system | **1** | CD kan booke bord, og se bookinger via programmet. | **D.20/4-15** |
| Saldo & Konto funktion | **3** | CD kan fylde saldo, på sine kunders konto. | **D.27/4-15** |
| Kunde database | **1** | CD kan manualt administrere bruger, tilføje, redigere og slette. | **D.20/4-15** |

**Other Product Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Need** | **Priority** | **Features** | **Planned Release** |
| Se saldo funktion | **3** | Bruger kan gå ind på App og se Saldo og bookinger. | **D.10/5-15** |
| Log ind funktion | **1** | Bruger kan oprette sig, og logge sig ind og ud af App. | **D.1/5-15** |
| View funktion. | **2** | Kunden kan se forskellige events og sortiment. | **D.5/5-15** |

Dette er for vores Mobil applikation. Dette laver vi, hvis alt går som planlagt. Det er vores andet prioritet at lave mobil applikation. En ekstra ydelse/produkt vi vil lave, men *kun* hvis vi når vores overordnet deadline med vores hovedprojekt.

## Use-Case Model

Målet med use-cases er at holde beskrivelsen simpel og letforståelig. Man anvender typisk use-cases som det første skridt i en objektorienteret designproces. Denne metode definere systemets adfærdsmæssige sammenhæng, og definere kravene til systemets funktionalitet. I use-cases defineres systemets overordnede funktioner og de medvirkende aktører. Dvs, hvem bruger systemet, i hvilket scenarie, og hvad er deres mål? Man vælger denne beskrivelse, hvis der i processen er involveret kunder eller andre interessenter, som designet tager udgangspunkt i.

**Usecase Program:**

**Usecase 1 (Admin): Login for Admin**

* Kunden kan logge ind på program.

**Usecase 2 (Admin) Sortiment & Event**

* Kunden kan redigere events eller sortiment og pris.

**Usecase 3: Booking**

* Kunden kan booke bord, og se bookinger via programmet.

**Usecase 4: (Saldo)**

* Kunden kan fylde saldo

**Usecase 5 (Bruger Handling)**

* Kunden kan manualt administrere bruger, tilføje, redigere og slette.

**Usecase Mobil App:**

**Usecase app 1 (Er oprettet): Login**

* Bruger kan gå ind på App og se Saldo og bookinger.

**Usecase app 2: Fyld Saldo**

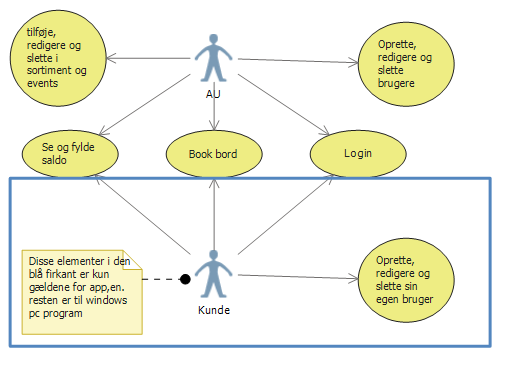
* Bruger kan gå ved Café og fylde saldo op.

**Usecase app3 (Er ikke oprettet): Bruger handling**

* Bruger kan oprette sig, og logge sig ind og ud af App.

**Usecase App3 (Kunde vil ikke oprette): Sortiment**

* Kunden kan se forskellige events og sortiment.
* Kunden kan logge ind på program.



**Brief Use-Cases:**

**I projektets opstartsfase har vi lavet en Brief use-case for at skabe et forkus omkring emne og mål. Brief use-casen har til formål at beskrive succes scenariet for systemet så overordnet og kortfattet som muligt.**

## Fully dressed usecase:

**Scope**: ‘Booking’

**Level:** User Goal

**Primary Actor:** AU[[4]](#footnote-4)

**Stakeholder and interests:**

* AU kan betjene kunderne hurtigere.
* AU har bedre oversigt over bookinger.
* Kunden sikrer sig en booking/plads.

**Forudsætning:**

* AU er logget ind.

**Success Guarantee**

* Booking er gemt og sendt til database. Der tilbagemeldt en succes, via en pop up box.
* Den gemte booking kan ses på bordplan View, via lyse rødt.

**Main Success Scenario (Or Basic Flow)**

1. Kunden ringer eller møder personligt op, for at booke eller reservere bord.
2. AU starter nyt booking via trykke på booking knap.
3. AU indtaster tid og Dato.
4. Systemet viser hvilke bord der er ledige.

Au gentager trin 3 indtil trin 4 er gældende.

1. AU vælger ledig bord og trykker på Udfører booking knappen.
2. Systemets view går tilbage til start view og succes pop up box dukker op.
3. Bookingen er gennemført, og kunden kan sætte sig på det booket bord på det givne tidspunkt.

**Special Requirements:**

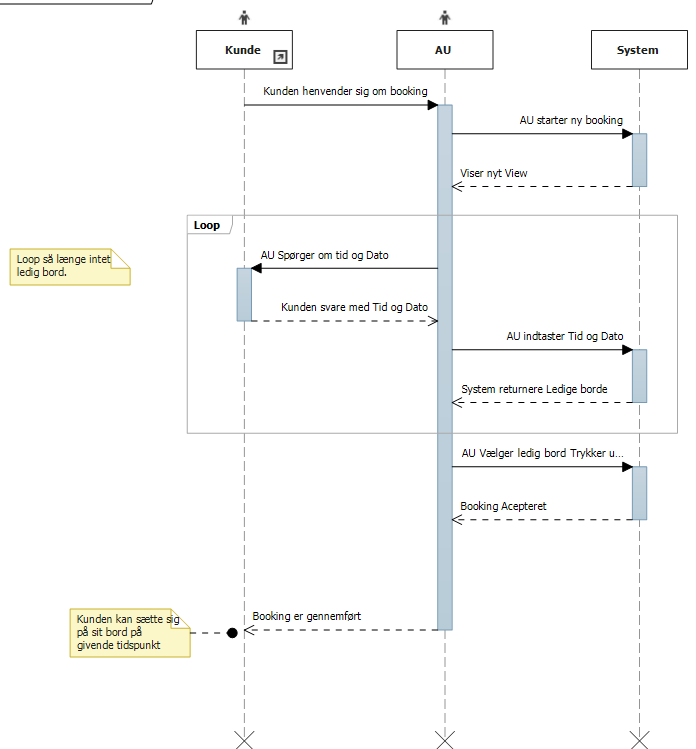
* Det skal være en Windows 8 computer.

**Frequency of Occurrence:**

* Der må være en forsinkelse på max 5 sekunder. Det skal være forudsætteligt.

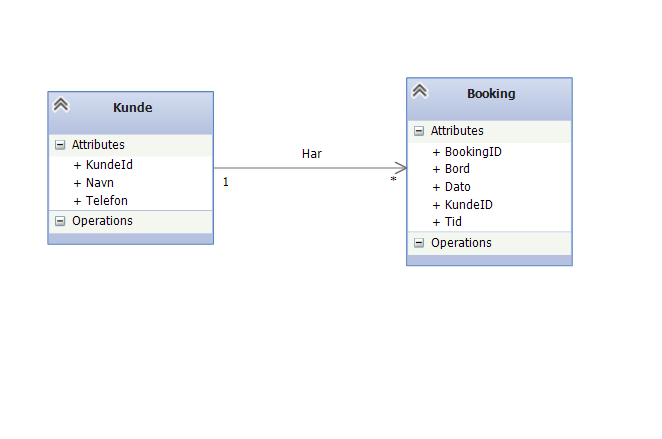
**Open issues:**

* Skal AU logge af når de er færdig med booke.
* Hvad vil der ske hvis computeren slukker under en igangværende booking.
* Hvad vil der ske, hvis man trykker tilbage under en igangværende booking.

Domain model & SSD

Dette er domæne modellen for første itteration af OVP. Denne model beskriver de forskellige attributer som anvendes til use-case omhandlende booking af bord. For nærmere forståelse for denne, se Full-adressed use case omhandlende booking af bord.

Dette er system sekvens diagrammet. For første itteration af OVP, dette diagram beskriver hvordan aktørere som kunden og AU(se glossary) i samarbejde med systemet OVP kan udføre use-casen omhandlende booking af bord. For nærmere forståelse for denne, se Full-adressed use case omhandlende booking af bord.



# Konklusion:

For at besvare vores problemformulering, var det nødvendigt at kende vores kunde. Vi startede med undersøge, hvilken ”slagmark” vi var i. Vi undersøgte vores kundens branche, analyseret virksomheden i alle nødvendige aspekter. Samtidigt holdte vi møder med ejeren af virksomheden, for at være sikker på at vi havde samme opfattelse af situationen og komme til enighed i vejen til løsningen.

Efter at have afsluttet denne rapport kan vi konstatere ud fra vores analyser, at Cafe Diamond er i et homogent marked, da forbrugerne ikke kan differencer deres produkter fra andre lignende produkter. Samtidigt kan vi se, at vores kunde er i et marked, hvor der er hård konkurrence. Der er en intens konkurrence iblandt de eksisterende konkurrenter. Det handler om at få-/holde på de loyale og faste kunder.

Ud fra SWOT analysen og Porters 5 Forces, står Cafe Diamond meget stabil. Især da de bl.a. bruger kvalitets produkter såsom importeret tobak og kul. Samtidigt når de holder Events, såsom fodbold aftener, PPV Boxing, så skaber det ifølge Porters Value Chain, konkurrencemæssige fordele.

Vores samlede konklusion ud fra vores observationer og analyser er, at vi kan give vores kunde en kæmpe konkurrencemæssige fordele ved opbygge et software program, der vil til dels lette arbejdsbyrderne, gøre booking mere overskuelig, og promote deres events til deres kunder.

Vores system vil være koblet op til en database, der vil effektivisere virksomhedens arbejdsgang. Med dette vil AU[[5]](#footnote-5) bruge mindre tid på administrative opgaver, samt vil AU med database systemet få bedre overblik over bord bestillingerne.

Når vi kigger på vores kundens branche, så kan vi se at der er ikke noget lignende produkt på markedet. Det er unikt og revolutionerende inden for branchen. Dette vil medfører, at vores kundens virksomhed vil udskille fra sine konkurrenter og give kæmpe konkurrencemæssige fordele.

Indledning til Eloboration

SCRUM/SCRUMBOARD

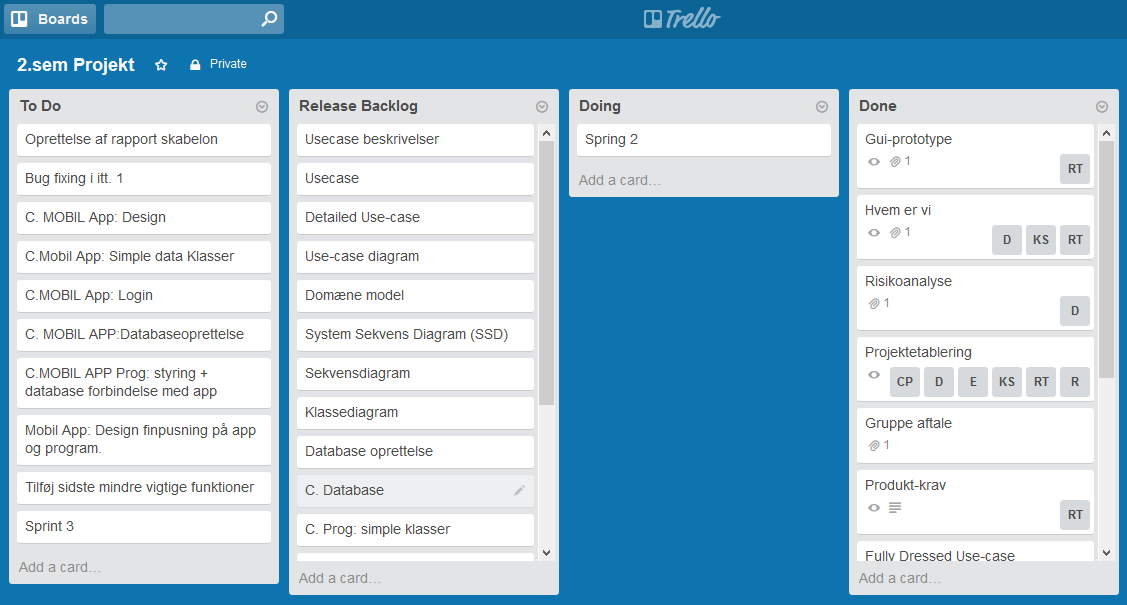
Vi bruger SCRUM, så vi nemmere kan styre vores projekt. Det giver os en struktur, og sikrer resultater, da vi arbejder med et projekt der kan ændres i forløbet.

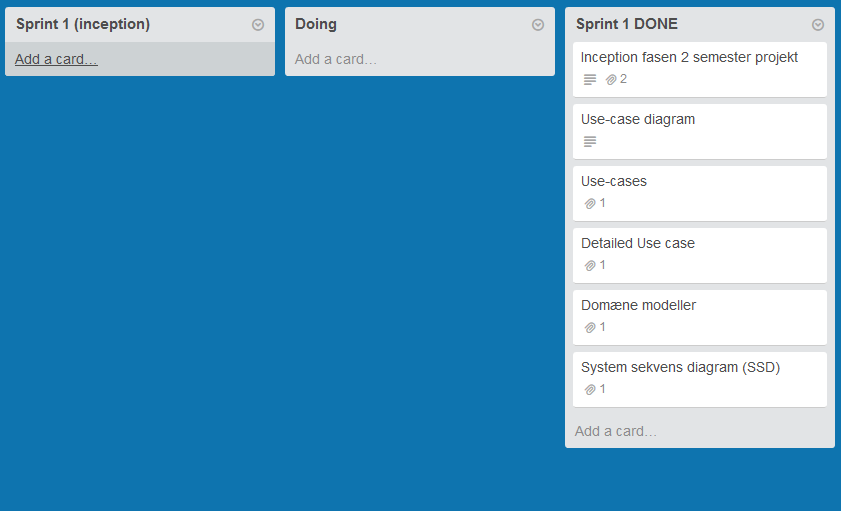
Sammentidlig giver det os bedre overblik på hele forløbet, og gøre det nemmere at delegere opgaverne.

Vi arbejder i Iterationer, som vi kalder for ”Sprints”, på den måde kan vi se hvor langt vi er i projektet, samt løbende opbygge viden om opgaven og dens løsning.

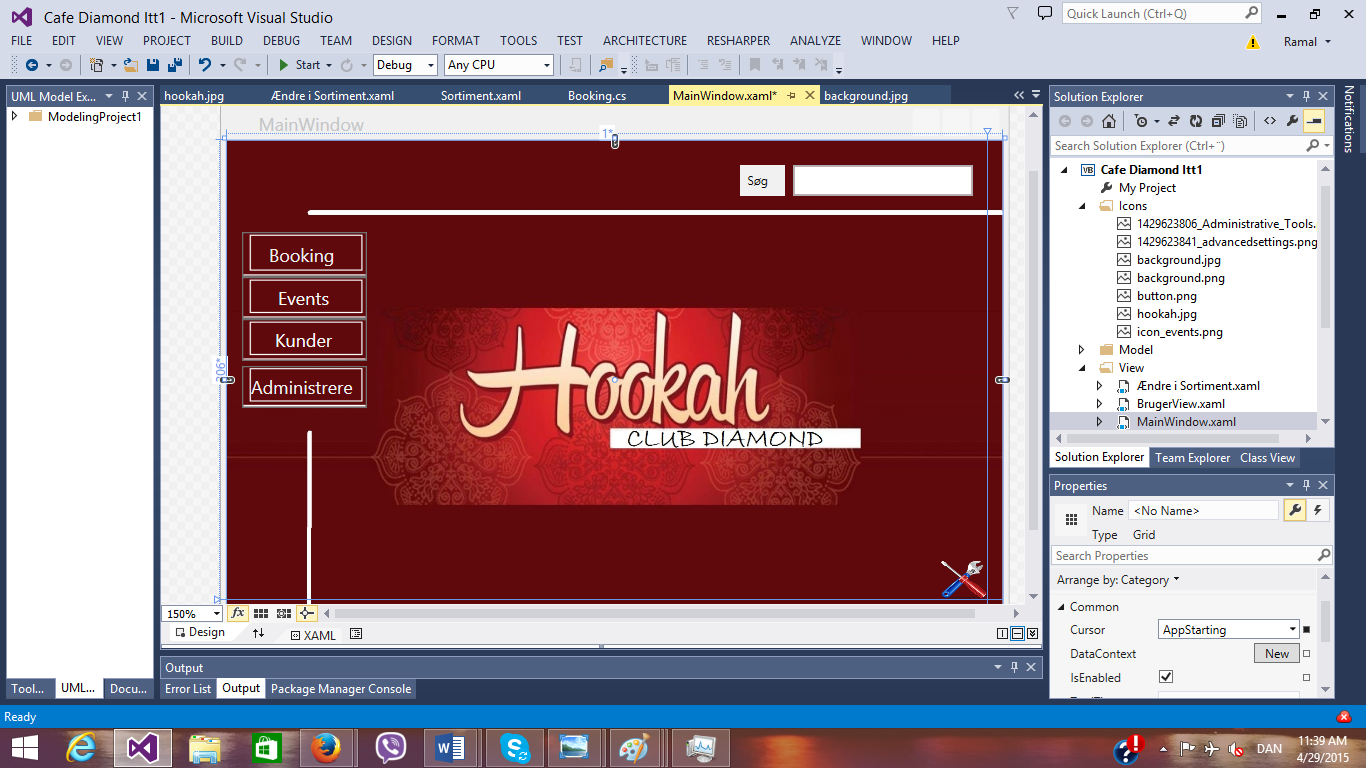
Vi startede med nedbryde projektet/opgaven i delelementer, altså lavede et ”Realese Backlog”, og derefter satte vi emnerne i hvert Sprint. Det vil sige, for hvert Sprint er der valgt de vigtigste delopgaver som vi vil løse. Dette metode gøre, at vi får en rigtig god følelse og gennemsigtighed i processen og forløbet. Da det er delt i delopgaver, betyder det at vi som gruppe levere kvalitetsløsninger på disse opgaver/Task.

For at vedligeholde denne smurt kørende proces, holder vores Team hver dag et 15 min. SCRUM møde. I mødet afklare vi om der er noget der forhindre Teamet i at arbejde optimalt, hvad der skal laves, generelt hvor vi er i processen. Vi skriver en logbog[[6]](#footnote-6) for hvert SCRUM møde. På den måde kan vi altid se hvad aftalerne imellem gruppen var, og det giver bedre forståelse når vi løbende kigger på vores Scrumboard.





# VIEW/GUI



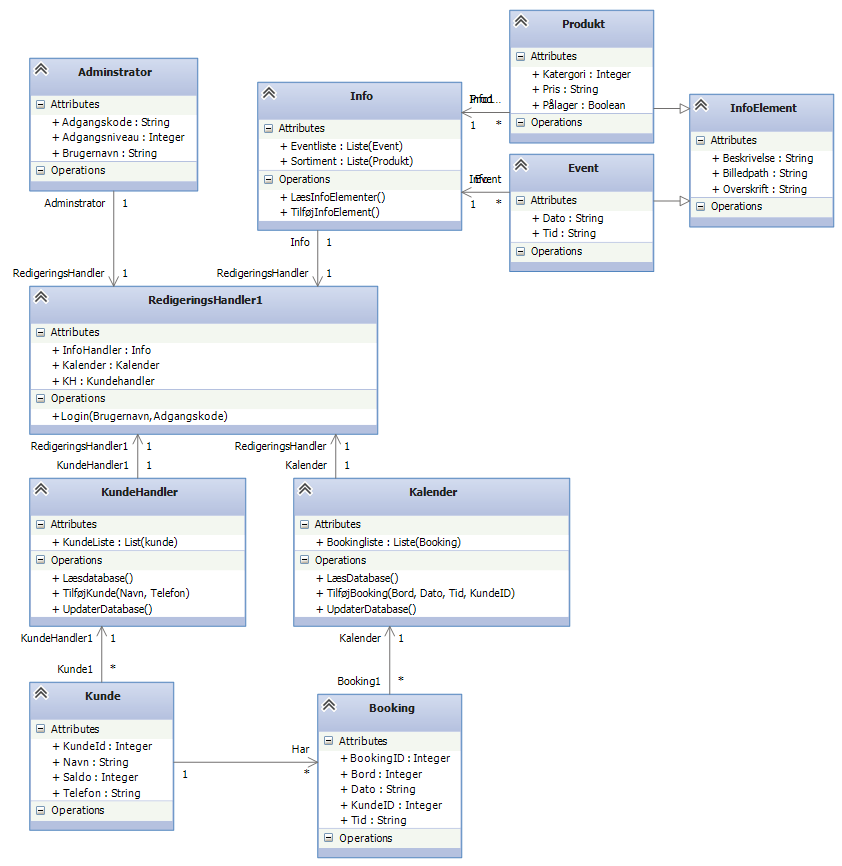
Vi har valgt denne design, da det passer til Caféens farver, og giver en mellemøstlig stemning som vores kunde ønskede. Vores kunde prioritere meget en tryg og god stemning og ønskede at farverne og design skulle afspejle cafeens indretning.

Vi har lavet baggrunden ud fra kundes ide, kundes valg af farver. Farverne gør sådan så det er nemt for kunden at se knapperne og overskueligt. Det er afslappede farver som giver kunden et godt overblik over programmet.

Vi vil have en knappe form, som vi synes var den mest overskuelige. Vi har taget de vigtigste/brugbare knapper som kunden bruger mest til at navigere rundt. Vi har valgt at sætte en søg funktion på forside så kan kunden hurtig finde sin Kunder og redigere - f.eks. fyld saldo, booke bord.

# Klasse diagram & MVVM

Klasse diagrammet bruges ofte til at strukturer programmets eller dele af programmets klasse indhold. Diagrammet laves ofte på baggrund af en domæne model, som beskriver attributter og hvilke elementer der indgår i klasser. Klasse diagrammet er mest brugt til programmøren som skal definere operationer/funktioner og hvilke tekniske typer disse attributter skal have. Som ses (nedenfor, til venstre, til højre, ovenfor), har vi fremstillet et klasse diagram som er dybt baseret på vores domæne model, der er minimale forskelle på denne og vores domæne model. De væsentlige forskelle er type definition, som eksempelvis: under kunde er der en attribut kaldet “navn”. Denne attribut er defineret som værende en string, da det er den mest fornuftige samt nemmeste type at anvende til denne slags attribut. Udover type definition, tilføjes også operationer som er de funktioner som den givne klasse indeholder og skal udføre. Et eksempel på dette på vores klasse diagram: Under Kalender klassen er der en operation kaldet “Opretbooking”. Denne operation er den funktion som Kalender klassen skal udføre når der skal oprettes en ny booking, som tilføjer en ny instance af klassen “booking” og indsætter den, samt tilføjer den til databasen ved at kalde på en operation I samme klasse(Kalender). Dette diagram åbner op for en langt nemmere arbejdsprocess når delen “models” på MVVM strukturen skal laves, da strukturen er gennemtænkt og elementer er tilføjet design mæssigt. Dvs. det bare skal kodes.

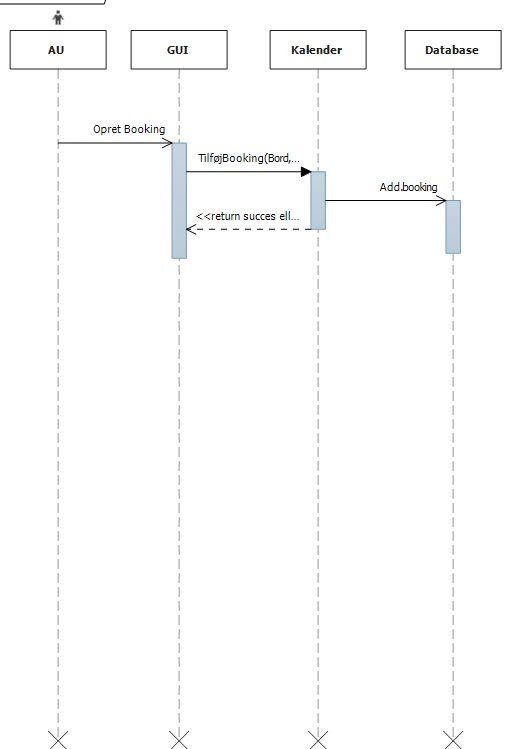
MVVM strukturen bruges til at opnå lav kobling, dette betyder at vi nemt kan ændre I viewet samt models uden at det har en stor betydning, da det ikke kræver meget at rette ting til efterfølgende. Vi har anvendt MVVM strukturen til netop dette formål. Der anvendes itterationsbaseret produktion I vores projekt. Det med, skal der laves flere itterationer hvor en lav kobling har stor fordel. Da der kan bygges videre og ændres på senere itterationer uden at ældre elementer bliver ugyldiggjort. På den måde kan vi nemt ændre tilføje og gøre ved I nyere itterationer uden arbejdsbyrden for ældrere elementer bliver særlig høj. Udover det, har denne struktur også den fordel at der er flere af vores gruppemedlemmer der kan arbejde på de delte ting uafhængigt, dette gør arbejdsprocessen hurtigere.

# Sekvens Diagram

Vi har brugt sekvens diagram til at få overblik over vores operationers kald. Det vil sige, fra hvilke elementer/Klasser som operationen kalder fra og hvor den kalder til, og hvilke data der kaldes med. Samt hvad der sker i operationen. Dette har vi gjort for at få bedre overblik over de operationer som der er i programmet, så det bliver nemmere senere at programmere.

Der er 4 emner, en aktør, GUI'en, Kalender og databasen. Aktøren er det emne der starter operationen, det kunne ligeså godt have været en anden operation som startede denne operation, men i dette tilfælde er det brugeren af programmet. Der kaldes til Kalender klassen for at udføre metoden OpretBooking.

Denne metode opretter et objekt af Booking og sender den til databasen, hvilket også ses på diagrammet, samt giver det svar tilbage til GUI'en, at bookingen er blevet oprettet og gemt i databasen.



# Konklusion af Eloboration

# Indledning Til Construction

I Inception og Eloboration fik udviklet en stabil arkitektur. Formålet med Construction fasen er at færdiggøre udviklingen af systemet på grundlag af det udarbejdet arkitektur. Det vil sige, at vi skifter fokus fra indsamling og opbygning af viden, til at begynde med produktion af koden, og fastholde vores arkitekturs integritet.

For at minimere risikoerne ved programmering, vil vi lave en grafisk repræsentation af vores data i vores database. Sammentidlig vil vi kigge på vores data tabeller og se på normalisering af dem.

I Construction fasen vil vi optimere kvaliteten af udviklingen via lave en god planlægning. Det vil sørge for at vi vil spare tid, minimere unødvendige omarbejde. Vi vil arbejde trinvist og der vil være en vis grad af parallelitet i vores arbejde, da der er komponenter der kan udvikles uafhængigt af hinanden. Da vi har udviklet en robust arkitektur i Eloboration, vil der ikke være nogle kompleksitet når programmet skal synkronisereres sammen. Det vil sørge for, at vi vil opnå en passende kvalitet, samt hurtigt opnå en brugbar beta version.

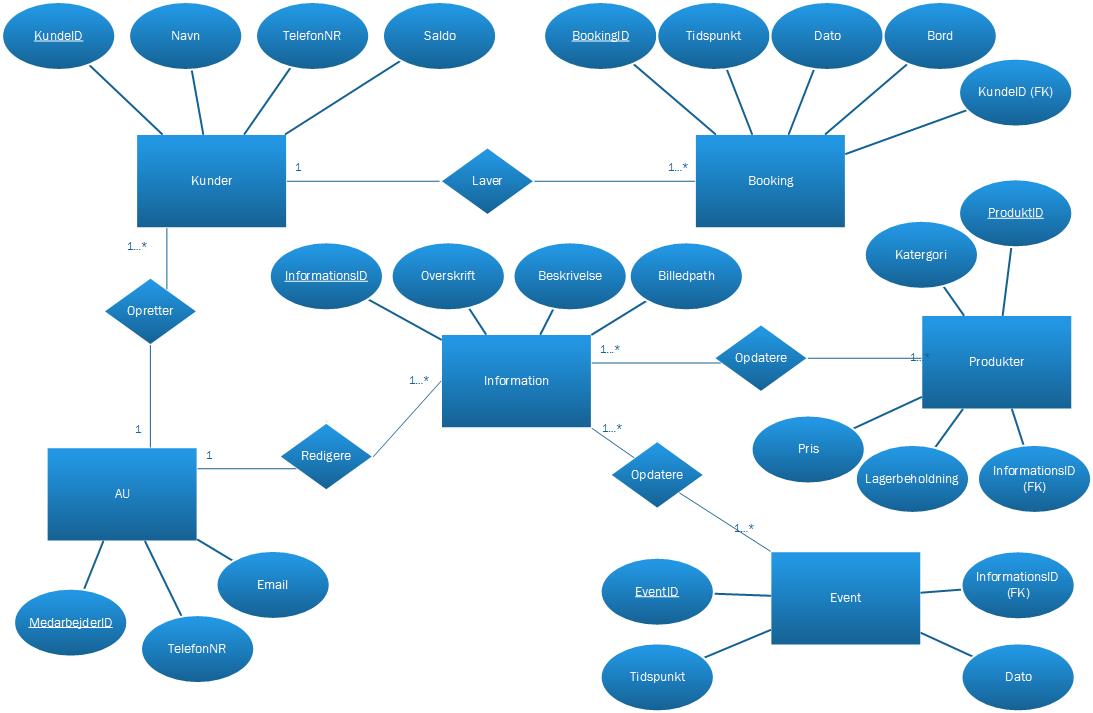
Vi har sat en review i vores planlægning, hvor vi har udarbejdet en reviewplan[[7]](#footnote-7), så vi kan få feedback for vores program. Det vil medfører færdiggørelse af analyse, design og afprøvning af funktionaliteten. Alt dette vil optimere kvaliteten, da vi kan arbejde og videreudvikle i nogle komponenter, som vil medfører en optimal program til vores kunde.

# ER-Diagram

Vores ER-Diagram viser en grafisk repræsentation af organisering af vores data i vores database. Den viser hvordan vores data tabeller hænger sammen imellem entiteterne og hvad deres attributter er.

Den viser hvordan AU[[8]](#footnote-8), kan oprette kunde i vores system, så kunden kan lave en booking.

Sammentidlig giver den en god forståelse for, hvad der sker, når AU ændre noget data/Info i programmet – hvori man kan se hvordan det information opdatere resten af systemet.



# Database

# Data tabeller & Normalisering

# Implementering af koder

# Implementering af koder

# Review

Inden vi kan indlevere produktet til vores kunde, så er det vigtigt for os, at det bliver lavet en review af vores program. Dvs., at vi får en anmeldelse af vores program på de kriterier som vi sætter op. Vi har sat en review gruppe (review G1) op, som vil ”bruge” vores program. Review G1 fik udleveret en ”Review formular[[9]](#footnote-9)”

Review G1 fik en kort gennemgang af programmet, og derefter fik udleveret en vejledning, som de fulgte og udfyldte. Resultaterne har vi vedhæftet som billag[[10]](#footnote-10). Vi ville gerne vide, hvilke elementer i vores program kan gøres bedre, og de brugerinput vi fik er følgende:

**Hvad kan gøres bedre?:**

|  |
| --- |
| Fejl i systemet når man opretter bruger  Fejl meddelelse: ”Der opstod en fejl” men ikke hvilken fejl.  Vi kunne ikke bruge symboler fx ”@”  I navne feltet kan man kun skrive fornavn.  Hvis der kun er navn, hvordan kender i så kunden?  Hvis man kan huske brugerens navn og hvordan den staves kan den godt findes i databasen  Designet kan godkendes  Der er småfejl der skal rettes |

Sammentidlig lavede vi en præsentation af vores program til en uafhængig review gruppe (review G2) som kunne komme med alle de spørgsmål som de kunne have, samt kommentere deres tanker om programmet. Vi fik konstruktivt feedback, angående navnet af vores program, samt designet som kunne forbedres.

Det feedback vi fik af vores review G1 & review G2 var meget brugbart. Det medførte at vi ændret de fejl der var i vores program, samt forbedret de elementer vi fik feedback på.

Efter forbedringerne og ændringerne snakkede vi med vores kunde. Efter gennemgangen af resultaterne af reviewformularen, samt feedbacksne fra reviewgrupperne, samt visning af vores program, var kunden respons positiv. Kunden var glad for de ændringer der var lavet, og syntes den var mere livlig og passer til hans café. Man kan sige at vi fandt en overset krav efter vores gennemgang af review.

# Fastsættelse af design

Man kan sige at skelettet til programmet er stadig det samme, men udseendemæssigt er kommet små ændringer. Bl.a. har vi fået ændret det navn der stod på programmet og fik lavet en tagline der ’omfavner’ essensen i caféen og programmet.

Før Gamle Design

EFTER – Nuværende Design

Vores kunde prioriteret en god og tryg stemning. Vi har valgt den grå nuance, da den afslappende og beroligende[[11]](#footnote-11), samt det grønne nuance som symbolisere ungdoat \*\*\*\* Resten skrives når view er færdig\*\*\*\*

# Konklusion af Construction

# Indledning til Transition

# Konklusion af Transition

# Bilag 1

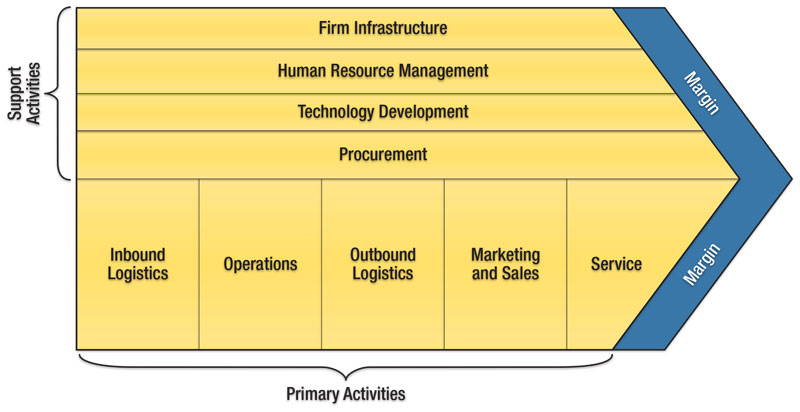
**SWOT ANALYSE**

# Bilag 2

**Porters 5 Forces**

**Truslen fra nye konkurrenter:**Truslen er relativ stor eftersom vandpibe cafe markedet ikke er mættet og flere får øjne op at det er en god branche med vækst i.   
  
**Leverandørens forhandlingskraft:**  
Forhandlingskraften hos leverandøren er ikke særlig stor eftersom der er stort udbud af vandpibe tobak og kul fra Mellemøsten.   
  
**Konkurrencen i branchen:**  
Konkurrencen i denne branche er rimelig stærk eftersom der er mange nye vandpibe cafeer som kan trække stamkunder og kunderne væk med billige priser osv.  
  
**Kundernes forhandlingskraft:**  
Kunderne har en stor forhandlingskraft eftersom det er den eneste indtjeneste som virksomheden laver.  
  
**Truslen fra substituerende produkter:**  
e-cigaretter og smøger er en stor truslen for vandpibe cafeer.

# Bilag 3

**Porters Value Chain**

**Firm infrastructure:**  
Ejer og ansatte.  
  
**Technology management:**  
Facebook side  
  
**Procurement:**  
Vandpibe, Vandpibe tobak.   
  
**Human ressource management:**  
Ansatte er en vigtig egenskab til virksomhedens succes.  
  
**Inbound logitistics:**  
God kvalitet af vandpibe tobak og kul  
  
**Operation:**  
vandpibe cafeen opererer i en butik  
  
**Service:**  
servicen er den vigtigste, da det skal gøre at kunderne kommer tilbage og vælger denne cafe fremfor andre vandpibe cafeer.  
  
**Marketing/sale:**  
ikke så meget marketing, eftersom det er en lille virksomhed. Bruger kun sociale og digitale medier for at fremme virksomheden.

# Bilag 4

**Glossary:**

**Admin** = Café Diamond ejer

**User** = Medarbejderne

**AU** = Admin & Medarbejder

**Kunde** = Café Diamond kunder

**PPV** = Pay-per-view (Et form for betalings tv)

# Billag 5

**Review af eBooking**

**Forudsætninger:**

Vi tester vores program på Windows 8.1 pro. Vi vil lægge vægt på brugervenligheden, og funktionaliteten på programmet.

**Hvad skal bedømmes:**

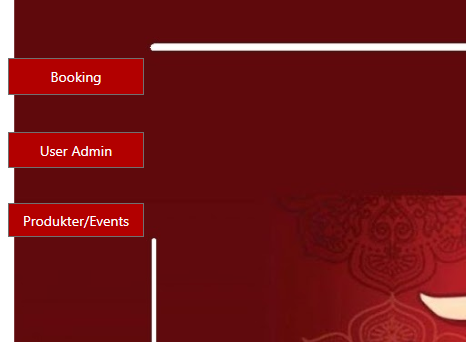
Der skal laves review på ’User Admin’ funktionen.

**Hvordan gør vi det:**

Review gruppen får udleveret en ’Review formular’, som de skal udfylde sammentidlig med de afprøver programmet. Review gruppen får en kort gennemgang af programmet af udvikler holdet, og derefter følger de følgende vejledning:

**Vejledning til hvordan User Admin funktionen fungere:**

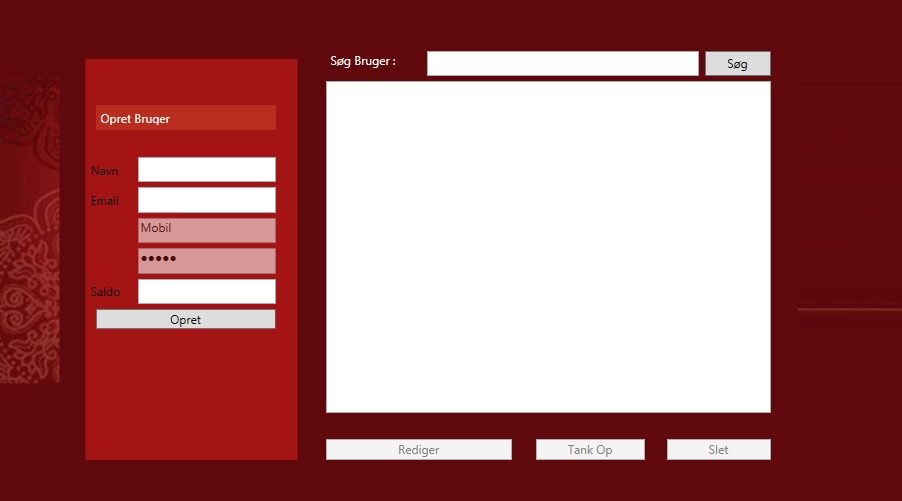
1)



For at komme ind på oprette bruger:

* Tryk på “User Admin” knappen

2)



For at oprette en bruger:

* Udfyld felterne og tryk på opret.

For at søge bruger:

* Skriv navn i søge feltet for at søge.

# Billag 6

**Review Formular:**

**Bedømmelse 1-5, hvor 5 er det højeste ”score”:**

**Kommentar felterne *skal* udfyldes.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Review** | **Beskrivelse** | **Bedømelse (1-5)** | **Komentar** |
| **R.1 – Design** | **Enkelt design?** | **5** | **Fint og overskueligt** |
| **R.2 – Usability** | **Overskueligt og nem at bruge?** | **5** | **Let forståeligt** |
| **R.3 - Efficiency** | **Hurtigt og effektivt og formår at løse opgaven** | **3** | **Når den fungere** |
| **R.4 - Correctness** | **Fungere den som den skal** | **2** | **Hvis der ikke er symboler** |
| **R.5 - Reliability** | **Fungere den hvergang, og kan brugen fortsættes** | **1** | **Se nedestående** |

**Hvad kan gøres bedre?:**

|  |
| --- |
| Fejl i systemet når man opretter bruger  Fejl meddelelse: ”Der opstod en fejl” men ikke hvilken fejl.  Vi kunne ikke bruge symboler fx ”@”  I navne feltet kan man kun skrive fornavn.  Hvis der kun er navn, hvordan kender i så kunden?  Hvis man kan huske brugerens navn og hvordan den staves kan den godt findes i databasen  Designet kan godkendes  Der er småfejl der skal rettes |

# Billag 7

1. 23-03-2015 [↑](#footnote-ref-1)
2. Bilag 1,2,3 [↑](#footnote-ref-2)
3. Se business canvas model under Customer relationship. [↑](#footnote-ref-3)
4. Glossary = Bilag 4 [↑](#footnote-ref-4)
5. Se glosery: AU: Ejeren eller/og Medarbejderen [↑](#footnote-ref-5)
6. Billag 6 Logbog [↑](#footnote-ref-6)
7. Billag 5 [↑](#footnote-ref-7)
8. AU = Admin & Medarbejder – Billag 4 [↑](#footnote-ref-8)
9. Bilag 5 – Review formular [↑](#footnote-ref-9)
10. Bilag 7 – Resultat af review [↑](#footnote-ref-10)
11. ”Bog om nuancer” [↑](#footnote-ref-11)