LKBH § 107 – Bookning af vikar tider

2. semester eksamensprojekt (Datamatiker)

Et billede, der indeholder skærmbillede, Rektangel, design, kvadratisk

Automatisk genereret beskrivelse

**J**on Lundby Nielsen (09-08-1985)

*GitHub username - JonLundby*

**M**agnus Carstensen (19-09-1992)

*GitHub username - Elvasfar*

**M**arkus Ingerslev Olsen (20-04-2001)

*GitHub username - MarkusIngeslev*

**P**alle Jensen (02-06-1987)

*GitHub username - PalleGregersJensen*

GitHub frontend repository:

<https://github.com/MarkusIngerslev/LKBH-semester-projekt-frontend>

GitHub backend repository:

<https://github.com/MarkusIngerslev/LKBH-semester-projekt-backend>

Deployet frontend:

(github-pages-link)

Backend REST-API

(asure-eller-database-link)

Gruppe – JMMP

Dat23V2

Kea – Københavns Erhvervsakademi

Afleveret 15-12-2023

**Indhold**

[**Indledning inkl. problemformulering** 2](#_Toc151660721)

[**Diagrammer** 3](#_Toc151660722)

[Usecases 3](#_Toc151660723)

[Klasse-diagram 7](#_Toc151660724)

[ER-diagram 7](#_Toc151660725)

[Substitutes Table 8](#_Toc151660726)

[Shifts Table 8](#_Toc151660727)

[ShiftInterest Table 8](#_Toc151660728)

[**Analyser** 9](#_Toc151660729)

[Risikoanalyse 9](#_Toc151660730)

[Risk-management strategi 9](#_Toc151660731)

[Identificer risici 9](#_Toc151660732)

[Sandsynlighedsvurdering af risici 9](#_Toc151660733)

[Risikoovervågning 10](#_Toc151660734)

[Risikofaktorernes konsekvens 11](#_Toc151660735)

[Risikoanalysetabel 12](#_Toc151660736)

[Aktionsplan 12](#_Toc151660737)

[Udvidet risikotabel 13](#_Toc151660738)

[Feasibility study 14](#_Toc151660739)

[Teknisk gennemførlighed 14](#_Toc151660740)

[Økonomisk gennemførlighed 14](#_Toc151660741)

[Lovmæssig gennemførlighed 15](#_Toc151660742)

[Operationel gennemførlighed 15](#_Toc151660743)

[Planlagt gennemførlighed 15](#_Toc151660744)

[Politisk gennemførlighed 16](#_Toc151660745)

[Beslutning 16](#_Toc151660746)

[Teknologier 17](#_Toc151660747)

[Beskrivelse af kode 18](#_Toc151660748)

# **Indledning inkl. problemformulering**

Firmaet LKbh beskæftiger sig med at tilbyde et botilbud under servicelovens §107 samt ADL-træning (almindelig daglig levevis). Deres målgruppe er voksne i alderen 18-50 år, der har diagnoser inden for ASF (Autisme Spektrum Forstyrrelser), OCD og angst.

I dette projekt samarbejder vi med LKbh for at skabe en ny løsning – et bookingsystem for firmaets vikarer. I øjeblikket sendes der en SMS-besked ud til alle vikarer for at høre, om de har mulighed for at tage en ledig vagt. Vi vil implementere en løsning, hvor en ansvarlig driftsmedarbejder i stedet kan gå ind på en hjemmeside og indtaste de vagter, der i øjeblikket er ledige. Dette giver de forskellige vikarer i firmaet mulighed for at byde på de forskellige ledige vagter efter deres ønsker. Den driftsansvarlige har derefter mulighed for at godkende de forskellige vagter i overensstemmelse med vikarernes ønsker.

Formålet med denne løsning er at gøre det nemmere for både den driftsansvarlige og de forskellige vikarer at få et overblik over deres vagter samt identificere eventuelle tidspunkter, hvor der mangler dækning. Vi forventer også, at dette vil reducere den tid, der bruges på at planlægge vagter.

# **Diagrammer**

## Usecases

I projektet bruger vi usecases til at skabe os både et overblik, men også til at få en ide om hvad det er kunden godt kunne tænke at vores produkt gjorde. Nedenfor har vi et diagram for hvilke muligheder der er på hjemmesiden. Et billede, der indeholder tekst, skærmbillede, diagram, cirkel

Automatisk genereret beskrivelse

Vi har efterfølgende lavet en Use Cases (breif) for de forskellige tilfælde:

|  |  |
| --- | --- |
| Use case titel / ID | Admin får vist liste af vikarer / UC001 |
| fremgangsmetode | 1. Admin logger ind på websiden. 2. Systemet verificerer admin. 3. Systemet fetcher data med vikarer og genererer liste 4. Systemet viser liste med vikarer. 5. slut. |

|  |  |
| --- | --- |
| Use case titel / ID | Admin opretter ny vikar / UC002 |
| fremgangsmetode | 1. Admin navigerer til visning af vikar liste. 2. Admin trykker på opret ny vikar. 3. Systemet viser opret ny vikar formular. 4. Admin udfylder formular og trykker opret/bekræft. 5. Systemet dobbeltjekker bekræftigelse. 6. Hvis formular er valid:   systemet opretter ny vikar i database.   1. ellers:   system viser fejl besked.   1. Slut. |
| Use case titel / ID | Admin får vist specifik vikar / UC003  (AFLYST) |
| Steps involved | 1. Admin navigerer til vikar liste. 2. Admin vælger/trykker på specifik vikar. 3. System shows detail view.   (AFLYST) |
| Use case titel / ID | Admin sletter specifik vikar / UC004 |
| fremgangsmetode | 1. Admin navigerer til specifik vikar. 2. Admin trykker slet 3. Systemet spørger om bekræftigelse 4. Hvis admin bekræfter:  * Systemet sletter specifik vikar (baseret på id)  1. ellers:  * Systemet viser fejlbesked  1. Slut. |
| Use case titel / ID | Admin opdaterer specifik vikar / UC005 |
| fremgangsmetodess | 1. Admin navigerer til specifik vikar og trykker opdater. 2. Systemet viser opdaterings formular med detaljer fra den valgte vikar. 3. Admin laver ændringer i formular og trykker opdater. 4. Systemet spørger om bekræftigelse. 5. Hvis: formular er valid  * Systemet skriver den opdaterede vikar til databasen.  1. ellers:  * systemet viser fejlbesked.  1. End. |
| Use case titel / ID | Admin får visning af liste med vagter / UC006 |
| Fremgangsmetode | 1. Admin logger ind på websiden. 2. Systemet verificere admins log ind. 3. Systemet fetcher data for vagter og genererer liste 4. Systemet viser liste med vagter. 5. slut. |
| Use case titel / ID | Admin opretter ny vagt / UC007 |
| fremgangsmetode | 1. Admin navigerer til visning af vagt liste. 2. Admin trykker på opret ny vagt. 3. Systemet viser opret ny vagt formular. 4. Admin udfylder formular og trykker opret/bekræft. 5. Systemet dobbeltjekker bekræftigelse. 6. Hvis formular er valid:   systemet opretter ny vagt i database.   1. ellers:   system viser fejl besked.   1. Slut. |
| Use case titel / ID | Admin opdaterer vagt / UC008 |
| fremgangsmetode | 1. Admin navigerer til specifik vagt og trykker opdater. 2. Systemet viser opdaterings formular med detaljer på den valgte vagt. 3. Admin ændre formular og trykker opdater. 4. Systemet spørger om bekræftigelse. 5. Hvis: formular er valid  * systemet gemmer opdateret vagt I databasen.  1. else:  * systemet viser fejlbesked.  1. Slut. |
| Use case titel / ID | Admin sletter vagt / UC009 |
| fremgangsmetode | 1. Admin navigerer til specifik vagt og trykker slet. 2. Systemet spørger om bekræftigelse. 3. Hvis: admin bekræfter.  * Systemet sletter vagt.  1. Ellers:  * systemet viser fejlbesked.  1. Slut. |
| Use case titel / ID | Vikar får visning af vagter/ UC010 |
| fremgangsmetode | 1. Vikar logger ind på websiden. 2. Systemet verificere vikar. 3. Systemet fetcher data med vagter og genererer liste 4. Systemet viser liste med vagter. 5. Slut. |
| Use case titel / ID | Vikar får sortere + filtrere af vagter / UC011 + UC012 |
| fremgangsmetode | 1. Vikar navigerer til visning af liste med vagter. 2. Vikar vælger sortering eller filtrerings muligheder. 3. system viser listen på ny i sorteret og/eller filtreret tilstand 4. Slut. |
| Use case titel / ID | Vikar byder på ledig vagt / UC013 |
| fremgangsmetode | 1. Vikar navigerer til specifik vagt og trykker ”byd på vagt”. 2. System opdaterer vagten i vagt databasen. 3. Slut. |
| Use case titel / ID | Vikar fjerner bud på vagt/ UC014 |
| fremgangsmetode | 1. Vikar navigerer til specifik vagt og trykker fjern bud. 2. systemet opdaterer vagten i vagt databasen. 3. Slut. |
| Use case titel / ID | Vikar får visning af egen info/ UC015 |
| fremgangsmetode | 1. Vikar navigerer til ”min side” 2. system finder specifik vikar baseret på hvilket ID brugeren(vikaren) er logget ind med. 3. system viser vikarens egne kontakt informationer. |
| Use case titel / ID | Vikar redigerer brugernavn + password / UC016 |
| fremgangsmetode | 1. Vikar navigerer til ”min side” og trykker rediger brugernavn + password. 2. systemet viser formular. 3. Vikar laver ønskede ændringer og trykker ok. 4. systemet spørger om bekræftelse. 5. Vikar bekræfter. 6. System gemmer opdatering i vikar database. 7. Slut. |

## Klasse-diagram

## ER-diagram

I denne database har vi 3 tabeller. ”Substitutes”, der indeholder information om hver vikar i databasen med hver deres unikke ID. En tabel ”Shifts”, der indeholder praktiske informationer om alle vagter såsom dato, starttidspunkt, sluttidspunkt, medarbejderen, der har vagten og et unikt ID til hver vagt. ”Medarbejder” i tabellen er en foreign key fra ”Substitutes”- tabellen, hvor man dermed knytter en unik medarbejder fra Substitutes-tabellen til en unik vagt.  
Derudover har ”Shifts”-tabellen en boolean værdi ”ShiftIsTaken”, der beskriver, hvorvidt en vagt er taget eller ej.

Den tredje tabel ”ShiftInterest”, er en såkaldt ”junction”-tabel, der indeholder par af ”ShiftID” og ”EmployeeID”. Denne tabel registrerer vikarer, der har udtrykt interesse for forskellige vagter.

Tabellerne og opsætning af data i databasen skal opfylde 3. normalform og vi vil nu beskrive, hvorfor vores tabeller gør lige det.

### Substitutes Table

Alle kolonner afhænger af hele primærnøglen (EmployeeID).

Kolonnerne er direkte afhængige af EmployeeID, så dette opfylder 1NF og 2NF.

### Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, design Automatisk genereret beskrivelseShifts Table

Alle kolonner afhænger af hele primærnøglen (ShiftID).

EmployeeID er en fremmednøgle og afhænger kun af ShiftID, og ShiftID er en entydig identifikator for rækkerne i denne tabel. Dette opfylder 1NF og 2NF.

### ShiftInterest Table

Alle kolonner afhænger af hele primærnøglen (ShiftID, EmployeeID).

Både ShiftID og EmployeeID er nødvendige for at identificere rækkerne i denne tabel, og begge kolonner er simple (ikke-sammensatte) nøgler. Dette opfylder 1NF og 2NF.

Derfor opfylder de tre tabeller 1NF og 2NF. For at bekræfte, om de opfylder 3NF, skal vi se på transitive afhængigheder, hvor ingen ikke-nøgleattribut af en tabel er transitivt afhængig af nogen supernøgle. I vores tabeller er der ikke nogen transitive afhængigheder, og alle attributter afhænger direkte af hele primærnøglen.

Derfor overholder de tre tabeller 3. normalform.

De 3 tabeller og deres relationer til hinaden kan ses illustreret til højre.

# **Analyser**

## Risikoanalyse

Vi har udarbejdet denne risikoanalyse for at identificere og lave en handlingsplan for risici forbundet med udviklingen af den løsning vi udvikler for ”LKbh”. Vi har primært haft fokus på de risici, der vedrører vores arbejde og udarbejdelse af løsningen.

### Risk-management strategi

Vi benytter os af en proaktiv risikostrategi, hvor risici bliver identificeret inden projektets start og der bliver udarbejdet en aktionsplan for hver enkel risiko, således, at alle trusler for vores projekt er kalkuleret for og medregnet i vores projektplanlægning.

### Identificer risici

Her er en liste over de risici vi mener er relevante for vores projekt:

1. Et medlem vælger at forlade gruppen.

* Dette er en personalerisiko, som anses som en risiko, da med en mand mindre i et team på 4 personer, så skal de tilbageværende arbejde væsentligt mere effektivt, for at nå i mål med projektet.

1. Tidsplanen skrider.

* Dette er en estimeringsrisiko, da vores estimering af opgaver/ userstories omfang ikke altid kan forventes at være præcise og i sidste ende kan resultere i, at vi ikke bliver færdig med vores projekt eller med alle de funktionaliteter som vi havde sat os for og aftalt med vores kunde.

1. Krav fra kunde kan ikke imødekommes.

- Dette er en produktrisiko, hvor det produkt som kunden forventer, at vi som team leverer, ikke kan leveres. Det er derfor en risiko ift. det færdige produkt og dermed en produktrisiko.   
Hvis ikke vi kan levere produktet, om det så grunder i manglende tid eller kodeevner, så må det kunne konkluderes, at vi ikke har fuldført opgaven.

1. Fejlestimering af opgavers / userstories tid til fuldførelse.

* Dette er en estimeringsrisiko, hvor vores manglende erfaring som programmører også medfører en manglende evne til at estimere brugen af tid til diverse opgaver.

1. Kunde ændrer i krav eller kommer med nye krav undervejs i projektet.

* Dette er en produktrisiko, hvor produktet som kunden ønsker ændres. En ændring af produktet kan medføre, at vi som team udfører en masse spildt arbejde og bruger tid og ressourcer på noget som ikke skal bruges. I tilfældet med nye krav, der vil omfanget af produktet stige som vi ikke tidsmæssigt har estimeret for.

### Sandsynlighedsvurdering af risici

Vi vil nu vurdere, hvor stor sandsynligheden er for, at de forskellige risici indtræffer. Det er self. relevant, at have denne sandsynlighedsovervejelse in mente, når man som team skal vurdere, hvilke risikofaktorer man skal have fokus på.  
Vi vil benytte os af denne model til at vurdere sandsynligheden af en risikofaktors skulle indtræffe i procent:

* Meget lav (<10%) værdien 1
* Lav (10%-25%) værdien 2
* Moderat (25%-50% værdien 3
* Høj (50%-75%) værdien 4
* Meget høj (>75%) værdien 5

1. Vi vurderer, at der, i vores tilfælde, er en lav risiko for, at en af vores gruppemedlemmer forlader os undervejs. Når projektet først er startet, er det svært at komme ind i en ny gruppe. Derudover skal man med en ny gruppe næsten starte forfra, så jo længere vi kommer i projektet, des mindre bliver risikoen for, at det sker.   
     
   Vi har givet denne risikofaktor en sandsynlighedsværdi på 2.
2. At tidsplanen skrider for os, er en reel trussel. Vi har inden projektets start ikke 100% fastlagt vores kundes krav og vi ved heller ikke med sikkerhed, at kravene kan løses og hvor lang tid det vil tage at udvikle. Denne usikkerhed gør, at sandsynligheden for at denne risiko indtræffer er relativt høj.

Vi har givet denne risikofaktor en sandsynlighedsværdi på 4.

1. Vi har modtaget nogle krav fra vores kunde og selvom vi før har arbejdet med lignende løsninger og delelementer er velkendte for os, så er det vores største projekt til dato og med funktionaliteter som vi ikke har prøvet at implementere før.   
   Vi vurderer derfor, at der er en moderat risiko for, at vi ikke kommer i mål.

Vi har givet denne risikofaktor en sandsynlighedsværdi på 3.

1. Som risikofaktor 2, så er det en trussel, at tidsplanen skrider og vi er nye i faget som programmører og ved derfor ikke altid, hvor lang tid forskellige opgaver tager. Det er på grund af dette en stor trussel, at tidsplanen ikke bliver overholdt.  
     
   Vi har givet denne risikofaktor en sandsynlighedsværdi på 4.
2. At kunden i dette projekt skulle ændre i kravene eller komme med nye krav i løbet af projektet, vurderer vi ikke til at være særlig sandsynligt.   
   Vi forventer, at kunden kunne tænke sig ændringer i design eller mindre justeringer, men ændringer i de primære krav er ikke forventeligt.

Denne risikofaktor tildeler vi en sandsynlighedsværdi på 1.

### Risikoovervågning

Det er vigtigt, at alle risikofaktorer bliver overvåget løbende i den forstand, at vi som team i vores proces skal vurdere om risici har ændret sig. Er der kommet nye til? Er der eksisterende risikofaktorer, som ikke er aktuelle længere og kan fjernes fra vores risikoanalyse?   
Derfor bør alle risikofaktorer ved afslutningen af hvert sprint genovervejes ift. deres sandsynlighed, aktionsplan og hvorvidt det er en reel risiko længere.

### Risikofaktorernes konsekvens

Det er også vigtigt at vurdere, hvor stor en betydning eller konsekvens de forskellige risikofaktorer har for projektet. Hvis det er meget sandsynligt, at noget indtræffer, men at det ingen betydning har for arbejdsprocessen, det færdige produkt eller andre elementer af projektet, så er det ikke relevant at tage højde for.  
Vi har taget udgangspunkt i denne model til at vurdere de forskellige risikofaktorers konsekvens:

* Ubetydelig værdien 1
* Tålelig værdien 3
* Alvorlig værdien 7
* Katastrofal værdien 10

1. Hvis et gruppemedlem forsvinder, er det i princippet 25% af vores arbejdskraft, der forsvinder, hvilket må siges at være en væsentlig andel. Derfor må det derfor også betragtes, at det har en stor konsekvens for projektet og teamet, hvis dette indtræffer.

Vi har givet denne risikofaktor en konsekvensværdi på 7.

1. At tidsplanen måtte skride, er en reel trussel for os og med den tydeligt mærkbare konsekvens, at vi ikke bliver færdige med projektet. Konsekvensen er endnu højere, da vi ikke ligesom virksomheder har mulighed for at sætte ekstra arbejdskraft på vores projekt fra andre projekter eller ved at hyre konsulenter.

Vi har givet denne risikofaktor en konsekvensværdi på 7.

1. At kravene ikke er opfyldt, ville under normale omstændigheder være fatalt, men i dette projekt er det ikke et krav, at vi når i mål med alt. Det ville være at foretrække, at vi blev færdig med alle elementer, men så længe, at vi har anvendt de grundlæggende principper fra undervisningen og er blevet færdig med de primære krav fra kunden, så kan vores projekt stadigvæk godkendes.

Vi har derfor givet denne risikofaktor en konsekvensværdi på 3.

1. Forklaringen på konsekvensen for denne risikofaktor er tilsvarende for risikofaktor 2.

Vi har givet denne risikofaktor en konsekvensværdi på 7.

1. At kravene skulle ændres undervejs, har en stor konsekvens for om vi når i mål med projektet og påvirker også det arbejde som er udført og det som skal udføres.

Vi har givet denne risikofaktor en konsekvensværdi på 7.

### Risikoanalysetabel

Vi sætter nu værdierne for konsekvens og sandsynlighed for de respektive risikofaktorer ind i en tabel og som tommelfingerregel, så hvis produktet af de 2 er over 9, så er det en risikofaktor, der er værd at holde øje med og have en aktionsplan for.

|  |  |  |  |
| --- | --- | --- | --- |
| Risikoanalysetabel | | | |
| Risikoindex | Sandsynlighed | Konsekvens | Produkt |
| 1 | 2 | 7 | 14 |
| 2 | 4 | 7 | 28 |
| 3 | 3 | 3 | 9 |
| 4 | 4 | 7 | 28 |
| 5 | 1 | 7 | 7 |

### Aktionsplan

1. Den bedste måde at være sikker på, at folk bliver i et projekt, er at lave en kontrakt, der binder de enkelte til at blive i teamet til projektets afslutning.   
   En anden tilgang kunne være at sørge for, at alle teammedlemmer trives, føler at de bliver hørt og samtidig også kan omgås socialt. Derfor er det vigtigt at skabe et sundt arbejdsmiljø for ens team, hvis man ønsker et produktivt og tilfreds team.
2. Ved at have korte sprints eller i hvert fald daglige SCRUM-møder om vores udvikling eller mangel på samme, kan man fange, denne problematik og afhjælpe det i opløbet ved at omstrukturere arbejdskraft eller bruge flere timer.
3. Fordel arbejdsopgaver løbende og fokuser på de reelle krav.
4. Se risikofaktor 2.
5. For at undgå, at kundens krav ændres vil vi holde et uddybende møde med vores kunde omkring de UserStories som kunden har og beskrive kravene illustrativt med Wireframes, overfor kunden, for at sikre, at vi har den samme forståelse af, hvad kunden ønsker, både funktionelt og designmæssigt.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Udvidet Risikotabel | | | | | | |
| Risikoindex | sandsynlighed | Konsekvens | Produkt | Præventive tiltag | Løsningsforslag | Ansvar |
| 1 | 2 | 7 | 14 | Se aktionsplan | | Marcus |
| 2 | 4 | 7 | 28 | Jon |
| 3 | 3 | 3 | 9 | Magnus |
| 4 | 4 | 7 | 28 | Palle |
| 5 | 1 | 7 | 7 | Magnus |

### Udvidet risikotabel

Hele den foregående risikoanalyse bygger op til denne udvidede risikotabel, hvor alt, hvad vi indtil nu har fundet ud af, kan plottes ind i tabellen. Den eneste forskel er, at aktionsplanen bliver delt op i ”præventive tiltag” og ”Løsningsforslag” og derudover bliver ansvaret for den givende risikofaktor tildelt til nogen i teamet.

Vi vil her blot refererer til aktionsplanen, men tanken er at beskrive, hvad man kan gøre for at undgå, at en risikofaktor indtræffer og hvad man så kan gøre, når først lokummet brænder.

## Feasibility study

Mening med denne feasibility report er at vi vil gennemse de forskellige områder af projektet. Vi vil undersøge hvilke svagheder og udfordringer der kunne være ved et muligt projekt, men også hvilke styrker der ville kunne være.

Dette projekt har som formål at lave en hjemmeside med et skema, hvor en leder vil kunne oprette forskellige ledige vagter. Det er så muligt for forskellige medarbejder at mælde sig på de ledige tider. Inden projektet, så blev planlægningen gjort gennem sms beskeder, som hver medarbejder så kunne svare tilbage på, om de var ledige til at kunne tage en vagt. Mening med projektet er så at frigøre en stor mængde planlægnings tid, der så ville kunne bruges andetsteds.

### Teknisk gennemførlighed

I denne del af vores feasibility study analyseres de tekniske udfordringer i vores projekt.

Vi arbejder med programmeringssprog og -koncepter, som vi er bekendte med. Vi benytter Javascript, som vi kender forholdsvis godt og har arbejdet en del med, så er vi på forholdsvis sikker grund i forhold til at kunne komme i mål med vores projekt. Risiciene for ikke at kunne lave projektet vil være større ved implementering af programmeringssprog og koncepter, som vi ikke er bekendt med eller har så meget erfaring med. Der kan være delelementer i projektet, som vi ikke har prøvet at lave før.

Det projekt, som vi skal lave, er ikke nyskabende i forhold til udvikling af ny teknologi/programmeringskoncepter. Der vil derfor være en del hjælp at hente til at lave diverse delelementer. Vi er fire personer i projektgruppen, så der er forholdsvis meget arbejdskraft til rådighed i projektperioden.

Vores webapplikation kræver ikke, at brugere har store it-kundskaber eller særligt hardware. Appen vil fungere som mange andre standardhjemmesider. Der kræves blot internetforbindelse.

Det kan være værd at notere sig, at et feasibility study og en SCRUM-proces ikke nødvendigvis går hånd i hånd. Hvis man skal vurdere, hvorvidt et projekt kan lade sig gøre, kræver det også grundlæggende viden om projektet. I en SCRUM-proces er de endelige krav ikke kendt på forhånd, og det er derfor svært at vurdere, hvorvidt et sådant projekt kan lade sig gøre/er hensigtsmæssigt at føre ud i livet.

### Økonomisk gennemførlighed

Når det kommer til den økonomiske del af projektet, er der mange forskellige ting, der skal tages hensyn til. Før projektejeren kan få sit produkt, skal han først betale for udviklingen. Dette kan være i form af løn til udviklere og eventuelle materialer, som de kan få brug for under udviklingen. Efterfølgende vil der også være en pris for at holde produktet vedligeholdt/videreudviklet, hvor projektejeren også skal betale for eventuelle serveromkostninger samt løn til dem, der skal køre produktet.

I dette projekt er det os, der har kontaktet produkteejeren for at kunne lave vores projekt. Det er derfor os som udviklere, der gerne vil skabe et produkt, der kan være til gavn for produkteejeren, og vi vil derfor udføre dette projekt ulønnet.

Efter at projektet er færdiggjort, vil der kun være få ting, der kræver vedligeholdelse. Produkteejeren vil være nødt til at købe domænet, hvor hjemmesiden kommer til at køre. Det vil også være nødvendigt at have en person til at holde styr på databasen, hvor forskellige medarbejders personoplysninger er placeret, samt hvor tidligere vagter er gemt. Dette kan dog løses med en eller to personer.

|  |  |
| --- | --- |
| Cost | Benefit |
| ***Målbar***   * Server * Vedligeholdelse * Løn   ***Ikke målbar***   * Omstillings periode | ***Målbar***   * Ingen udviklingsomkostninger * Hurtigere planlægningstid   ***Ikke målbar***   * Bedre oversigt over tider |

### Lovmæssig gennemførlighed

I dette projekt vil der blive arbejder og behandlet forskellige personoplysninger. Men eftersom disse oplysninger allerede er blevet givet til arbejders, giver i forvejen, så vil der ikke være nogen brud på GDPR. I tilfælde af at en medarbejder skulle fratræde, så har virksomhed/arbejdes giver pligt til hurtigst muligt at få fjernet fratrådte medarbejders oplysninger.

### Operationel gennemførlighed

Det operationelle aspekt af et feasibility study omhandler bl.a., hvorvidt det foreslåede projekt løser virksomhedens problemer og understøtter virksomhedens forretningsstrategi.

I øjeblikket ringes der rundt til vikarer for at dækket ledige vagter. Med vores løsning vil dette ikke længere være nødvendigt i samme omfang. Dette projekt skulle derfor gerne lette arbejdsbyrden for planlæggerne på bostedet og gøre det mere overskueligt og gennemskueligt for vikarer at booke nye vagter og se tidligere vagter. Virksomheden har en hjemmeside i forvejen (<https://lkbh.dk/>), så det kunne være oplagt at gøre bookingsystemet for vikarer til en del af deres egen hjemmeside.

### Planlagt gennemførlighed

Deadline er sat til D. 15. december 2023. Denne deadline er ikke til forhandling da det er et eksamensprojekt. Det giver os i alt 4 arbejdsuger til at færdiggøre det endelige produkt. Vores gruppe består af 4 udviklere ud af 4 maksimalt tilladte, altså er vores arbejdskraft på fuld styrke. Vi bruger scrum som arbejdsproces og planlægger vores opgaver derefter med et tilhørende scrumboard og burndown chart. Disse værktøjer giver os et løbende indblik i vores tidshorisont og hjælper os med at forudsige eventuelle forsinkelser i god tid. Gennem vores uddannelse har vi allerede stiftet bekendtskab med lignende opgaver og er derfor nogenlunde beredt på omfanget af denne eksamens opgave. Ydermere har vi mulighed for at få vejledning gennem vores undervisere hvis der skulle opstå komplikationer undervejs.

Den første uge bruger vi på et ”sprint 0” hvor vi vil udvikle vores indledende analyser, diagrammer og datastruktur. I sprint 1 vil vi fokusere på at få backend og frontend til at fungere. Der skal gerne kunne fetche data og udføres CRUD operationer via lokal database i datagrip og postman. Sprint 2 vil omhandle implementering af bruger-funktioner som sortering og filtrering samt opsætning af database på Microsoft Azure. Det sidste sprint 3 er tænkt som buffer til forventet opsamling samt finpudsning. Eventuelle risici for ekstraordinære komplikationer vil fremgå af vores risikoanalyse.

### Politisk gennemførlighed

I dette projekt vil der ikke være politiske udfordringer, som der skal tages udgangspunkt i eller hensyn til

### Beslutning

# Teknologier

Hvilke teknologier vi har brugt/ valgt – til blandt andet database, deplyment, etc. Og kort argumentation for hvorfor.

# Beskrivelse af kode