



# OnlineBooking

---

Marc Hielscher

[ Marc Hielscher ]

Ragnar Svensson Rodenbeck

[  ]

Jesper La Cour

[  ]



---

<b>Indledning</b>	<b>4</b>
Problemstilling	4
Problemformulering	4
Tilgangsvinkel	4
<b>Gruppe etablering</b>	<b>4</b>
Gruppemedlemmer	4
Generelle aftaler	5
Gruppekonztrakt	5
<b>Projektetablering</b>	<b>6</b>
Værktøjer	6
visual studio	6
Github	6
Azure DevOps og Azure Portal (deploy)	7
swagger / postman	7
Teknologi	7
ASP.Net Core	7
<b>Systemudviklingsmetode</b>	<b>7</b>
Fravalg	7
Projektstyring	8
Scrum	8
"Iteration 0"	8
<b>Kravspecifikation/systemanalyse</b>	<b>9</b>
FURPS+	9
Functionality	9
Usability	9
Reliability	9
Performance	10
Supportability	10
Plus	10
Begrænsninger i implementeringsmuligheder	10
Fysiske krav	10
Domain diagram	11
Use Cases	11
UC1 Se aktuelle bookinger	11
UC2 Administrere bookinger (CRUD)	12

---

---

UC3 Oprette ny booking	12
UC4 Godkende ny booking	12
UC5 ændre og annullere booking	13
UC6 Sætte vare som utilgængelig	13
System Sequence Diagram	14
<b>Prototyping</b>	<b>14</b>
Horisontale prototyper	14
Filtrering af booking	15
Booking for kunden	15
Vertikale prototyper	16
<b>Arkitektur</b>	<b>17</b>
<b>Test og deployment</b>	<b>18</b>
Exploratory testing	18
Kvalitetssikring	19
Deployment	20
<b>Sprints - projektstyring i praksis</b>	<b>21</b>
Ceremonier	21
<b>Produkt/løsning</b>	<b>21</b>
<b>Konklusion</b>	<b>22</b>
<b>Perspektivering</b>	<b>23</b>
<b>Litteraturliste</b>	<b>23</b>
<b>Bilag</b>	<b>25</b>
Azure sprint 3 oversigt	25
Azure burndown chart	26
Prototype - kunde booking	26
Post Event Kalender/Controller/Swagger	27
Get Event Swagger/Controller	29
Put Event Controller/Azure vindue	30

---

---

## Indledning

[Forfatter: Jesper]

### Problemstilling

Del-Pin er en udlejningsvirksomhed centraliseret omkring syd- og sønderjylland og med adskillige års erfaring i branchen. Det er en veldrevet virksomhed med en gennemtænkt organisatorisk setup. Afdelingen vi har kontakt til Vejle. Del-Pin har et nuværende system i form af Microsoft Navision, hvor i debitorkartotek, ordre og faktura samt ressourcer er lagret/administreret. Deres nuværende system for bookinger er i en fysisk kalender i hver enkelt afdeling.

### Problemformulering

Del-Pin har et ønske om at se ind i en digitalisering af deres fysiske kalender. Der ønskes en mere eller mindre en-til-en erstatning for den fysiske kalender, hvor medarbejderne kan administrerer eksisterende, oprette og slette bookinger. Der ønskes i tillæg et modul til Del-Pins kunder, hvor der kan oprettes nye bookinger. Systemet må gerne kunne integreres i deres nuværende Microsoft Navision.

### Tilgangsvinkel

Vi ser problemet som et ret kritisk punkt for virksomheden og ændringer/fejl i system kan potentielt være forretningskritiske. Sikkerhed i systemet, fejlhåndtering og test inden deploy vil derfor være en del af vores fokus.

## Gruppe etablering

### Gruppemedlemmer

Ragnar Svensson Rodenbeck	Ragn0709@edu.ucl.dk	26297964
Marc Hielscher	Marc3411@edu.ucl.dk	31260636
Jesper la Cour	Jesp06f2@edu.ucl.dk	52501090

---

---

## Generelle aftaler

De enkelte afsnit i rapporten er angivet med pågældende forfatter. I koden angives der også med dato for oprettelse. Evt. større rettelser i kode / afsnit angives også.

Hvis der ikke er indikeret en forfatter på et afsnit er afsnittet skrevet i fællesskab og hvert gruppemedlem kan ses som ansvarlig.

I tillæg vedlægges oversigt fra GitHub med commits / bidrag.

### Gruppekontrakt

Hvad?	Hvordan?
Forventninger for projekt og gruppearbejde	Vi i gruppen har snakket om det fælles ønske, at udarbejde en rigtig god opgave som vi alle vil være glade for, at sætte vores navn på. Vi ønsker at arbejde målrettet og fast på, at møde de krav der stilles i opgaven således at vi kan løse dem efter bedste evne.
Aftaler om møder i gruppen	Vi forholder os til SCRUM framework se punkt SCRUM under Procesvalg. <ul style="list-style-type: none"><li>- Sprint planning holdes fysisk hver mandag</li><li>- Daily stand up kan holdes virtuelt (discord)</li><li>- Programmering foregår som udgangspunkt fysisk</li></ul> Skulle man være forhindret i, at møde gives der besked til gruppen, så tidligt som muligt.
Løsning af uenighed i gruppen	Ved større uenigheder om design af programmet forholder vi os til designprincipper, anerkendte mønstre og flertallets magt i gruppen. Der argumenteres fagligt for sin sag. Ved mindre uenigheder om kode valg er det op til den ansvarlige for området at træffe beslutninger, så længe det lever

---

	op til “kvalitet i programmering” og med respekt for implementering i programmet.
Forventninger til tidsforbrug	I SCRUM planlægges der med en arbejdsindsat svarende til 40 timer ugentlig. (svarende til 5 dage x 3 gruppemedlemmer) 20 dage.
Kvalitet i programmering	Vi forpligter os til at overholde de aftaler i punktet Kvalitet i programmering (side #)
Konsekvens af manglende overholdelse af ovenstående aftaler	Vi er i gruppen enige om en politik om “three strikes and you’re out”. I tillæg til dette forventes der respons til gruppen på dagligt plan og mangel på dette vil medføre udtrædelse. Overordnet vil meget kunne tilgives, så længe der er kommunikation og udmelding i god tid.
Ved udtrædelse af gruppen	Det fælles fremstillede vil til enhver tid tilhøre gruppen. Ved udtrædelse får man udelukkende råderet over egenproduceret indhold. Gruppen råder fortsat over det indhold som det udtrådte medlem har produceret.

## Projektetablering

[forfatter: Jesper]

### Værktøjer

#### visual studio

Vi har benyttet os af Visual Studio 2019 / VS for Mac som udviklingsmiljø.

#### Github

Som versionsstyring har vi brugt Git og Github til at hoste vores repository. Vi har brugt Github desktop til at udføre de mest almindelige Git kommandoer og sørge for at vi hver især har haft en

---

opdateret lokal version.

### Azure DevOps og Azure Portal (deploy)

Til at tracke vores projekt løbende har vi brugt Azure og specielt boards der viser oversigt over sprints og backlog. Desuden har vi også deployed vores program / API'er som web services via Azure Portal.

### swagger / postman

Til at teste og interagere med vores API'er / endpoints har vi brugt ovenstående programmer. Swagger som startup på vores API projekter og Postman til dokumentering af tests.

## Teknologi

### ASP.Net Core

Det er naturligt for os at udvikle API'er og MVC application i C# / ASP.Net.

Til brugerfladen har vi valgt en løsning der implementere et JavaScript library fra FullCalendar.IO. Dette gør det muligt at integrere en kalender på vores client side af MVC.

## Systemudviklingsmetode

[forfatter: Jesper]

Vi har i gruppen gjort os overvejelser om, hvilken systemudviklingsmetode der vil være bedst egnet for det kommende projekt. Nogle af de tanker der har lagt bag vores beslutninger og valg har blandt andet været kendte krav fra start, teamets størrelse og udviklingserfaring.

## Fravalg

Eftersom vi fra start kender krav til programmet og i tillæg også selv har været med til at udforme kravene, så vil man kunne argumentere for valget af en predictive model. heriblandt vandfaldsmodellen som egentlig kunne være meget anvendelig til projektet, hvis det ikke var fordi vi ser os som et uerfaren team med en lav chance for at designe

---

---

et helt program korrekt fra start og samtidig i en arkitektur der for os er mere eller mindre uprøvet. Vi vil altså med en adaptiv model have bedre mulighed for retningsskifte både tidligt, men også længere inde i projektet.

## Projektstyring

Der har egentlig fra start ikke været den store uvished om, hvilken udviklingsmetode vi helst ville arbejde med og i vores overvejelser var det mere et spørgsmål om, hvorvidt vi kunne finde grunde nok til at vælge noget andet. Det korte svar på spørgsmålet var nej, så valget er faldet på Scrum.

## Scrum

For os giver Scrum mulighed for, at vi allerede fra første uge i projektet kan begynde udvikling af vores bookingsystem uden vi har den fulde forståelse for, hvordan systemets endelige udformning bliver. Altså en af de store grunde til at fravælge en predictive model. Samtidig får vi nogle ufravigelige ceremonier, som er med til at støtte op om projektet og sikre en sikker retning.

Men eftersom Scrum kun giver projektstyring, så har vi selvfølgelig også inddraget praksisser fra mere systemorienterede udviklingsmetoder. Herunder par programmering, use cases, domain diagram og flere.

Man kan selvfølgelig argumentere for, at vi skulle have valgt en model rettet mere direkte mod systemudvikling, men vi kan egentlig godt lide den frihed det giver, at kunne mixe og matche det bedste fra begge verdener.

## “Iteration 0”

Et diskuteret emne når man vælger Scrum er altid pre-sprint/Iteration 0. De agile udviklingsmetoder antager ofte at projektet starter uden en egentlig form for forberedelse, så man hurtigt kommer igennem første iteration og leverer resultater til kunden øjeblikkeligt.

Vi har arbejdet med en iteration 0 i vores projekt. Her har vi indsamlet krav, arbejdet med artefakter fra de forskellige systemudviklingsmodeller og planlagt forløbet.

---



---

Bagsiden af en iteration 0 er, at det sjældent skaber værdi for kunden og ofte er mere for vores skyld, end kundens. Derfor har det også været vigtigt for os, at det har fyldt tidsmæssigt så lidt som muligt. En for omfattende iteration 0 kan også fjerne det agile i Scrum, da man er i risiko for at arbejde efter en mere predictive model.

## Kravspecifikation/systemanalyse

[Forfatter: Marc]

### FURPS+

#### Functionality

Virksomheden ønsker et bookingsystem der indeholder oplysninger om ressourcerne

Programmet skal kunne oprette en booking, der skal på bookingen kunne læses oplysninger om den bookedde ressource. Programmet skal vise oplysninger om nummer på ressourcen, timetal, lokation. Oplysninger om leveringsdato, sted og kontakt telefonnummer. Lagerstatus på ressourcen. Derudover skal programmet tage højde for lejeperiode og tage højde for andre bookinger af samme typer. Yderligere skal programmet tage højde for booking over helligdage og beregning af forsikring og evt. leje frie dage.

Programmet vil være et flere brugere system, således at flere brugere kan bruge programmet på samme tid og der vil være roller for de forskellige brugere, såsom områder kun medarbejdere kan tilgå og områder kun administrator kan tilgå.

#### Usability

Programmet vil indeholde knapper der visuelt er nemme at se teksten på og nemme at trykke på, således anvendelse fra mobiltelefon bliver nemmere. Det skal være nemt at navigere rundt i programmet, og let anvendeligt.

#### Reliability

---

---

Programmet bliver en online web løsning der holder styr på alle bookinger, det er derfor et forretningskritisk system. Derfor er det vigtigt med minimal downtime. Dette vil opnås via en pålideligt hosting service der sikre optimal uptime og at der ikke går data tabt.

### Performance

Programmet skal være hurtig og effektiv for brugeren, dette vil opnås på samme måde som i reliability ved at have en god hosting service der sørger for hurtige svartider på webserveren. I kode delen vil der arbejdes efter clean code principperne.

### Supportability

Programmet laves så det er nemt at teste. Arkitekturen og databasen laves til at det er optimalt til at vedligeholde og udskifte dele af programmet.

### Plus

#### *Begrænsninger i implementeringsmuligheder*

Systemet vil bestå af api'er og dto'er som står for overførsel af data, disse vil i første omgang ikke være udviklet til at arbejde sammen med MS Nav, og vil skulle ændres for at MS Nav vil kunne implementeres.

#### *Fysiske krav*

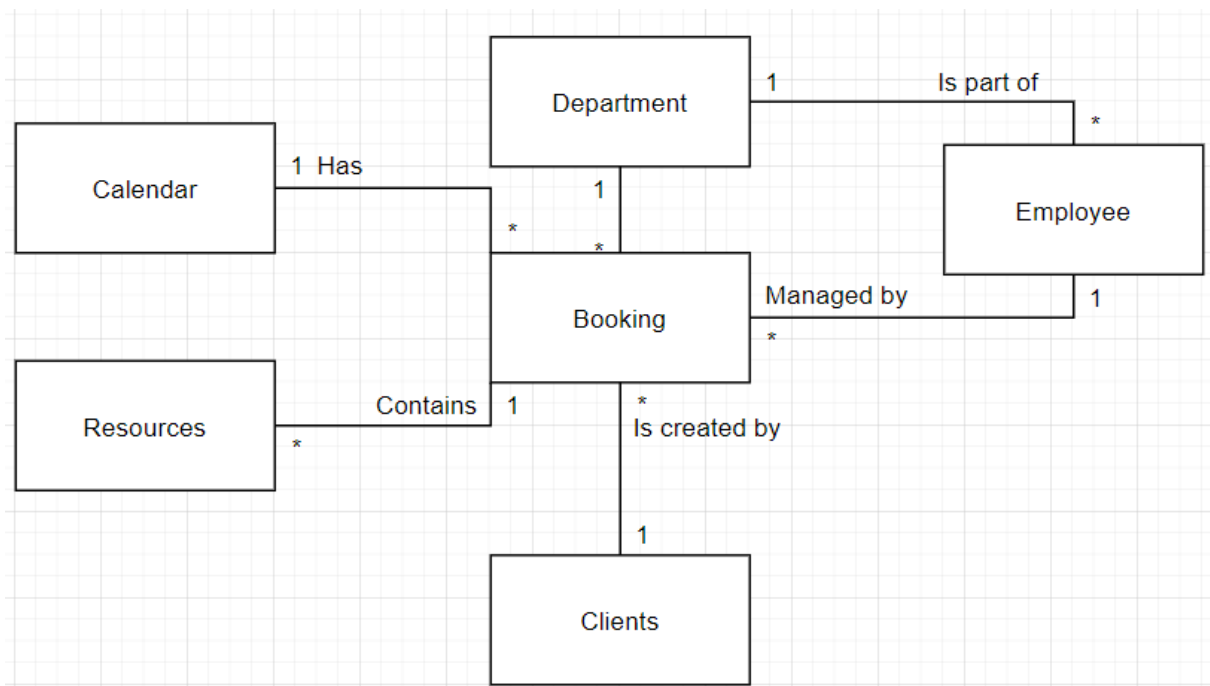
Der kræves at man har en computer eller tilsvarende i butikken med internetopkobling, hvor systemet skal tilgås da det vil være en webløsning.

---

---

## Domain diagram

[Forfatter: Ragnar]



## Use Cases

### UC1 Se aktuelle bookinger

**Actor:** Medarbejder

**Pre-conditions:**

Der er oprettede bookinger som kan vises og medarbejder er logget ind med medarbejderrettigheder.

**Main success scenario:**

1. Medarbejder logger ind
  2. Vælger herefter kalender som viser aktuelle dag og en oversigt med alle ressourcer
  3. Medarbejder filtrer på kategorier, lokation og lignende.
-

---

**Post condition:**

Der vises aktuelle bookinger og medarbejderen har mulighed for at redigere i dem.

**UC2 Administrere bookinger (CRUD)**

**Actor:** Medarbejder

**Pre-conditions:**

Medarbejderen er logget ind med medarbejderrettigheder.

**Main success scenario:**

1.
  - a. Booking der vil rettes vælges
  - b. Ny booking kan oprettes
2. (a) Det er muligt at ændre tidspunkt og dato eller fjerne bookingen
3. Ændringer skal bekræftes i popup

**Post condition:**

Rettelser er gennemført

**UC3 Oprette ny booking**

**Actor:** Kunde

**Pre-conditions:** Kunde er oprettet i online systemet og tilknyttet debitor. Kunden har fundet ønsket ressource i katalog (på hjemmeside)

**Main success scenario:**

1. Ønsket vare vælges i katalog
2. kunden logger ind / opretter sig som bruger
3. Periode for leje vælges, ønsket lokation for afhentning/levering udfyldes.
4. Kunden valideres (kreditgodkendes, tjekkes for spærring)
5. booking går til godkendelse hos medarbejder

**Post condition:**

Booking er til godkendelse hos medarbejder.

**UC4 Godkende ny booking**

---

---

**Actor:** Medarbejder

**Pre-conditions:** Kunde har oprettet booking online.

**Main success scenario:**

1. Medarbejder får notifikation om afventende booking.
2. booking sammenholdes med kalender (der undersøges for tilgængelighed) og afvises / godkendes.
3. (Medarbejder kan flytte booking til anden ressource)
4. Kunde får bekræftelse

**Post condition:**

#### UC5 ændre og annullere booking

**Actor:** Kunde

**Pre-conditions:** Kunde er oprettet i online systemet og tilknyttet debitor. Booking er oprettet.

**Main success scenario:**

**Post condition:**

#### UC6 Sætte vare som utilgængelig

**Actor:** Medarbejder

**Pre-conditions:** Ressource er oprettet i bookingsystem.

**Main success scenario:**

1. Menupunkt "gør ressource utilgængelig" vælges
2. Der udfyldes oplysninger (RessourceId, start-/sluttidspunkt, årsag).
3. Ressourcer "bookes" i kalenderen i den angivne periode.

**Post condition:**

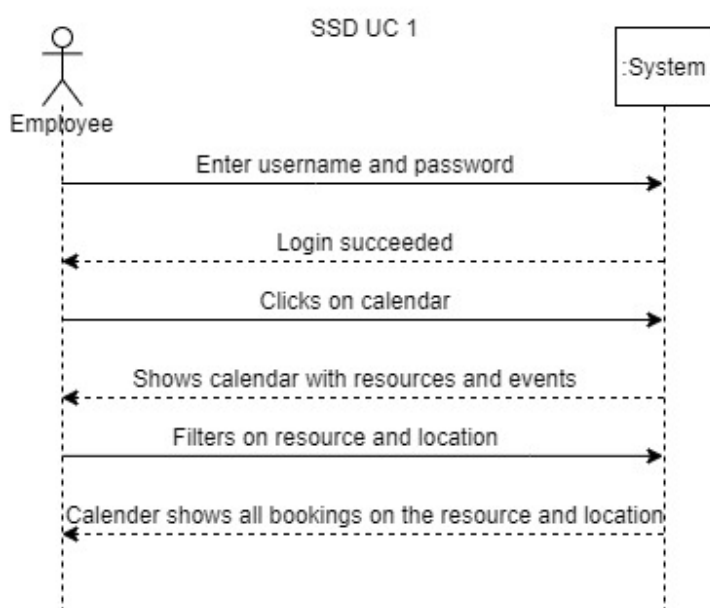
Varen gøres utilgængelig i en angiven periode

---

---

## System Sequence Diagram

[Forfatter: Marc ]



## Prototyping

*Forfatter: Jesper*

Vi har i opgaven medtaget prototyper af både typen horisontale (designs) og vertikale (funktioner) for bedre at kunne illustrere, forbedre og præcisere kravene. Det har også været en måde vi i de enkelte sprints har fastlagt retning for specielt designet.

De overvejelser vi har gjort os er dog også, hvorvidt behovet for prototyper er blevet mindre i takt med at funktioner løbende leveres for hvert sprint og det er blevet nemmere at ændre retning midtvejs i et projekt.

## Horisontale prototyper

[Forfatter: Jesper]

---

## Filtrering af booking

Vi har lavet en horisontal prototype der illustrere hvordan vi kunne have integreret en filtrering på vores ressourcer i kalenderen. Det er en prototype som nemt vil kunne ændres, men også kunne implementeres hvis det blev godkendt.

Filtrering (horizontal prototype)

Lokation

2. juni 2021

Resources	00	01	02	03	04	05	06	07	08	09	10	11	12
<input type="checkbox"/> Maskiner													
Gravko													

## Booking for kunden

Viser hvordan kunden vil kunne bruge eksisterende hjemmeside og vælge "book" af valgte udstyr. Kunden vil blive sendt videre til side, hvor der skal logges ind, vælges tidspunkt og indtastes kundeoplysninger<sup>1</sup>. Prototype kan tilgås via link<sup>2</sup>. Vertikal prototype vil kunne laves her.

<sup>1</sup> Se bilag: Prototype – kunde booking

<sup>2</sup> <https://delpinbookingv2.azurewebsites.net/Userevent/create>


UDLEJNINGSKATALOG
AFDELINGER
BLIV KUNDE

Udlejningskatalog / Bygningsmateriel / Donkraft / Maskindonkraft

Søg efter maskine
Hvad leder du efter?
Eller se A-Z Index >

### Maskindonkraft

**Holmatro - DK jack**  
[Læs mere >](#)



Løftekapacitet, maks.	10 t
Indsætningshøjde	20 mm
Løftehøjde, maks.	425 mm
Egenvægt	18,5 kg

Leje pr. dag\*  
**kr. 260,00** ekskl. moms  
kr. 325,00 inkl. moms  
[Book](#)

Kontakt	Afdelinger		Åbningstider		Information
Administration: Nordvesthavnsvej 60 6400 Sønderborg Telefon: 7448 8888 E-mail: post@delpin.dk	<b>Jylland</b> Sønderborg Rødekro Haderslev Kolding Vejle Esbjerg	<b>Fyn</b> Odense Assens	<b>Afdelinger</b> < 31.05.2021 - 06.06.2021 >  Mandag 6.30 - 16.30 Tirsdag 6.30 - 16.30	<b>Hovedkontor</b> < 31.05.2021 - 06.06.2021 >  Mandag 8.00 - 16:00 Tirsdag 8.00 - 16:00	Job Betingelser Bliv kunde Cookie Politik COVID-19 GDPR Levering & Afhentning

## Vertikale prototyper

[Forfatter: Ragnar]

I den vertikale prototype som vi bruger til at gå helt ned i dybden af programmet for at være sikker på at det virker. derfor lavede vi en vertikal prototype på nogle af vores funktioner. Første funktion som er blevet testet igennem i dybden var funktionen "Opret event" hvor der først logges ind på siden og forsiden vises, derefter åbnes kalenderen og der findes ønskede dato/tidspunkt og markeres med markøren, hvorledes opret ny event modal åbner og giver mulighed for at redigere de ønskede oplysninger. For denne specifikke funktion findes lavede vi senere en "QuickCreateBtn" som er en knap der giver mulighed for at åbne samme ny event modal, men med Tomme felter der så udfyldes. Da denne vertikale prototype blev lavet eksisterede den dog ikke, knappet er herefter blevet grundigt testet og virker.

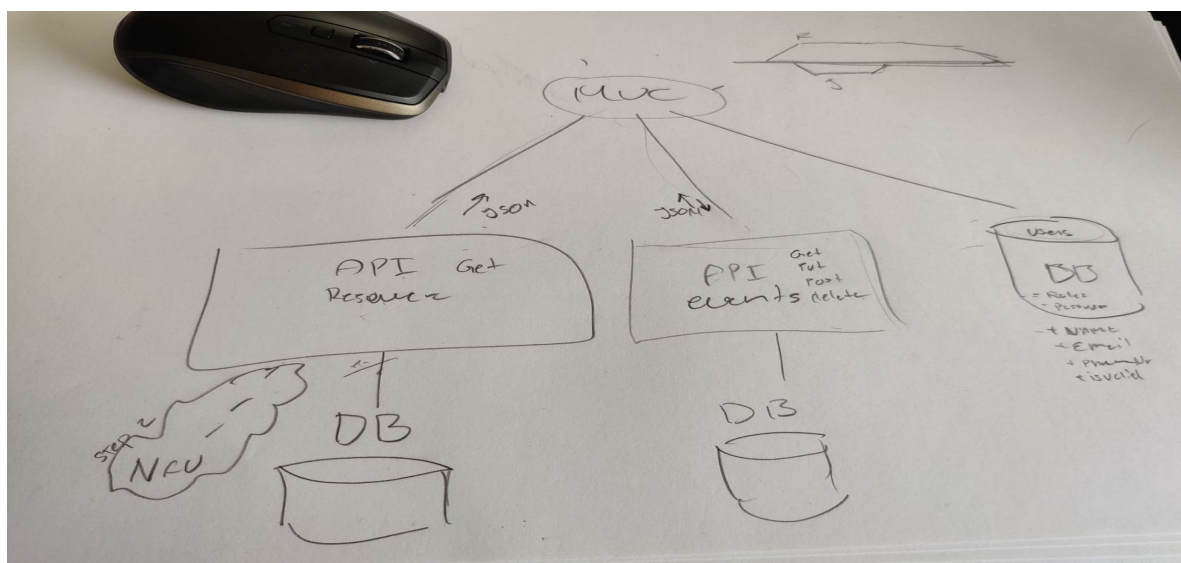


Efter der trykkes gem på modalen, vil informationerne blive gemt i event databasen via vores EventApi, denne funktion er testet via swagger<sup>3</sup>.

En anden vertikale prototype som blev lavet er lavet omkring funktionen GET til vores events. Efter der er logget ind og vedkommende er logget ind med admin eller employee trykkes der på kalender vinduet og kalenderen genereres, her vil der også forekomme en GET funktion som henter aktuelle events og displayer dem på den genererede kalender kaldt fra vores controller.<sup>4</sup> Alle events findes frem via vores EventAPI som er blevet testet i Swagger<sup>5</sup>

## Arkitektur

[forfatter: Ragnar]



Vores program består af en .Net MVC løsning, som står for alt UI kaldet "Delpinbooking", under Delpinbooking har vi en database på azure som håndterer kunder, såvel som ansatte. Delpinbooking snakker sammen med 2 API'er en til Ressourcer (ResourceAPI) og en som håndterer events (EventAPI). Disse to har hver især deres egen database som ligger på Azure.

<sup>3</sup> Se bilag 7, Swagger giver kode 200 (succes)

<sup>4</sup> Se bilag 9, Controller til vores HttpGet GetCalendarEvents

<sup>5</sup> Se bilag 8, swagger returnerer kode 200 (succes)

---

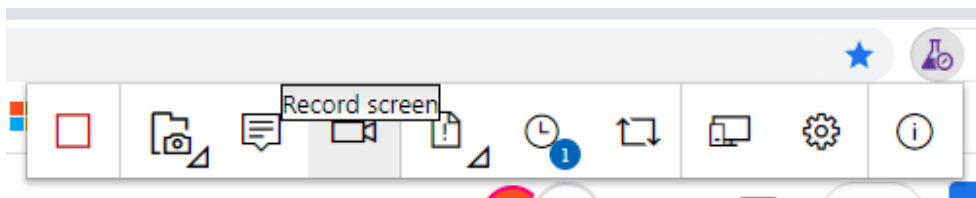
Ved at opbygge vores program med den lave kobling som det har, ender vi med at have meget nemt ved at udskifte eventuelt API'er eller tilføje andre API'er, et eksempel vil være at vi bruger en API til at vise alle adresser i Danmark Kaldet DAWA/dataforsyningen<sup>6</sup>. Andre gode grunde til at have denne løse kobling er at, databaserne til programmet også let kan skiftes ud, eller ændres i uden at det påvirker hoved løsningen. Der vil være rig mulighed med den valgte arkitektur at skifte databaserne til api'erne så man i fremtiden kan bruge vores løsning til.

## Test og deployment

### Exploratory testing

[forfatter: Ragnar]

Under vores sprints er der løbende blevet lavet exploratory testing, af programmet, ved hjælp af chrome udvidelsen "Test and feedback" som gav os direkte adgang til at submitte bugs, forslag og skærbilleder til azure, da der er direkte link derimellem.



---

<sup>6</sup> <https://dawadocs.dataforsyningen.dk/dok/api/adgangsadresse#autocomplete>

---

taskboardbacklogCapacityAnalyticsNEW WORK ITEMColumn Options

BUG 157

157 Same name when creating several events

Jesper Bjørn la Cour

1 comment

Add tag


StatgDone

AreaDelpinBooking

ReasonWork finished

IterationDelpinBooking\Sprint 2

Repro Steps



[View full image](#)

• Update input field with value test

TITLE

test

[View full image](#)

• Update input field with value test1

TITLE

test1

[View full image](#)

• Update input field with value test1bug

TITLE

test1bug

[View full image](#)

System Info

Browser - Name	Google Chrome 90
Browser - Language	da-DK
Browser - Height	1416
Browser - Width	2576
Browser - UserAgent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
OS - Name	Windows NT 10.0; Win64; x64
OS - Architecture	x86_64
OS - ProcessorModel	Intel(R) Core(TM) i5-9600KF CPU @ 3.70GHz
OS - NumberOfProcessors	6
Memory - Available	8036012032
Memory - Capacity	17097986048
Display - PixelsPerInchXaxis	96
Display - PixelsPerInchYaxis	96
Display - Height	1440
Display - Width	2560

Med vores agile fremgangsmåde scrum, har det været en stor fordel at teste samtidigt med vores sprints. Ved at bruge udvidelsen samt den undersøgende metode det er at bruge exploratory testing har vi også kunne finde fejl og mangler som vi har skrevet på til kommende sprints.

## Kvalitetssikring

[Forfatter: Marc]

Vi har brugt to forskellige code review metoder for at sikre god kode, vi har benyttet os meget af både par programmering og code review

---

gennem github. Dette har været med til at sikre at der har været flere øjne på koden og dermed bedre input til at få skrevet optimal kode.

Til GUI'en har vi benyttet os af exploratory testing som beskrevet ovenfor, dette er gjort med browser extensions for at finde bugs og sikre at UI'et er optimalt for brugeren at navigere.

Til vores hosting har vi valgt at lægge både databaser, api'er og klienten på azure, dette sikre en optimal uptime og hastighed ved at vælge en stor og pålidelig provider.

Vi har sikret programmet via lukkede API endpoints, authorization og authentication.

## Deployment

[forfatter: Jesper]

Den bedste strategi for en deployment af vores applikation har vi talt om kunne være en Gradual Cutover, hvor vi starter med at implementere system for én afdeling, mens de andre afdelinger fortsat arbejder via eksisterende arbejdsgange. Dette vil gøre virksomheden mindre sårbare over for kritiske fejl og eventuelle fejl vil i så fald også være begrænset til én lokation.

Et naturligt valg af afdeling kunne baseres på en evt. superbruger / lokation for product owner.

Af andre overvejelse indgår

- hardware

Hver medarbejder har eget login. Udviklet som webapplikation, så det er ikke nødvendig med egen pc samt specifikke krav til ydelse på hardware etc.

- Training

Vi har antaget at brugeren vil være med gennemsnitlig forståelse for IT. Der forventes kun kort træning i programmet, da det er udviklet med tanke på brugervenlighed. Træning af medarbejdere vil kunne gøres af en superbruger.

- Database / server

---

---

Deployment af applikation vil være på Azure Portals og vil derfor ikke stille krav til fysisk server/hardware for at deploye.

## Sprints - projektstyring i praksis

[forfatter: Jesper]

Som allerede angivet har vi valgt Azure DevOps til projektstyring og det har igen vist sig som uundværlig i relation til styring af vores backlog og tasks<sup>7</sup>.

Oversigten med tasks for selve Sprintet var med til at sikre arbejdsopgaver for alle i gruppen og at vi havde samme mål for Sprintet. Det fungerede samtidig som et sted, hvor vi hurtigt kunne indmelde bugs til senere løsning.

I løbet af ugen gav vores burndown chart et overblik over, om vi var med på planen eller måske skulle øge effektiviteten<sup>8</sup>

### Ceremonier

Vi har været meget opmærksomme på at få afholdt de ceremonier som er en del af Scrum.

Hver mandag har vi planlagt den pågældende uges Sprint (Use cases, backlog items og tasks). Det har fungeret således at hvert gruppemedlem har budt ind på de opgaver som man fandt interessant, men også udfordrende.

Dagligt har vi, uanset fysisk eller online fremmøde, haft en daily scrum / stand-up, hvor vi har revurderet planen, omfordelt eventuelle opgaver samt råbt højt hvis man var kørt fast i en task.

Hver fredag har vi haft et kort review og forberedt os til næste uge.

---

<sup>7</sup> Se bilag : Azure sprint 3 oversigt

<sup>8</sup> se bilag : Azure burndown chart

---

---

## Produkt/løsning

[Forfatter: Ragnar]

Ud fra problemstillingen og de krav vi er kommet frem til med kunden har vi fået et produkt som i bund og grund skal erstatte deres Maylandskalender, under udviklingen af program løsningen har vi sigtet efter at få lavet en løsning der lå tæt op af det som der allerede eksisterede. Simpelt nemt, troværdigt og til at finde ud af. (skærbilleder) (link til programmet?)

I løsningen er der forskellige roller som man kan administrere gennem logins, alle med forskellige sikkerhedsniveauer samt adgang. Admin har mulighed for at redigere i de forskellige roller kunder og ansatte har.

Kunder får ved oprettelse pr. automatik rollen "Customer" som kun har adgang til at booke ressourcer, rette i deres profil information og se deres egne bookinger.

Lige nu har vi lavet så at programmet er en løsning til en enkelt afdeling, at det skulle have flere afdelinger på var noget kunden ønskede men som skal laves videre på da andre funktioner er blevet højere prioriteret.

## Konklusion

[forfatter: Jesper]

Vi er med vores applikation kommet et skridt nærmere en digitalisering af den klassiske Mayland kalender. Vi ser selv nogle interessante løsninger på, hvordan man kan give brugeren samme overblik som deres nuværende "offline" kalender kan. Vi har lagt meget fokus på brugerfladen og synes at vi i den grad er lykkedes med et overskueligt og venligt design.

Et andet ønske var et system der kunne integreres med navision. Vi har på ingen måde lavet en integration, men i stedet har vi lavet en afkobling mellem vores MVC applikation og en web API. Skulle man ønske at integrere med navision, vil det kun være vores ResourceAPI der skal ændres i.

---

---

Sidste ønske var en booking-side for DelPins kunder. Vi har lavet en horizontal prototype på, hvordan vi forestiller os det kunne laves. Vores login/Identity system i programmet er forberedt på at håndtere kunder som en UserRole.

## Perspektivering

[Forfatter: Marc]

Siden starten af projektet har vi lagt en plan om hvordan vi ønskede at system skulle fremstå. Vi valgte efter nogle overvejelser at gå med FullCalendar til vores kalender. Den løsning viste sig at være en god løsning, FullCalendar har levet op til det vi håbede at få ud af den.

At arbejde med REST API'er har været spændende og udfordrende, især da vi har valgt at lave de 2 api'er og MVC'en i hver sine solutions dette gav os nogle problemer senere hen i forhold til docker, dette kunne være en ting som vi ville overveje at lave på en anden måde fremover.

Der var meget af implementationen i projektet som tog en del tid i forhold til hvad vi måske forventede da meget af det vi har arbejdet med er forholdsvis nyt for os, og derfor var der flere små problemløsninger som længere tid at løse end først antaget.

## Litteraturliste

Rod Stephens, *"Beginning Software Engineering"* .....

Craig Larman, *"Applying UML and Patterns"*, 3. udgave. ....

Andrew Lock, *"ASP.NET Core in Action"*. ....

ASP.Net Core opslag, *"<https://docs.microsoft.com/da-dk/>"* ..... Senest besøgt 02.07.21

---

\_\_\_\_\_

三

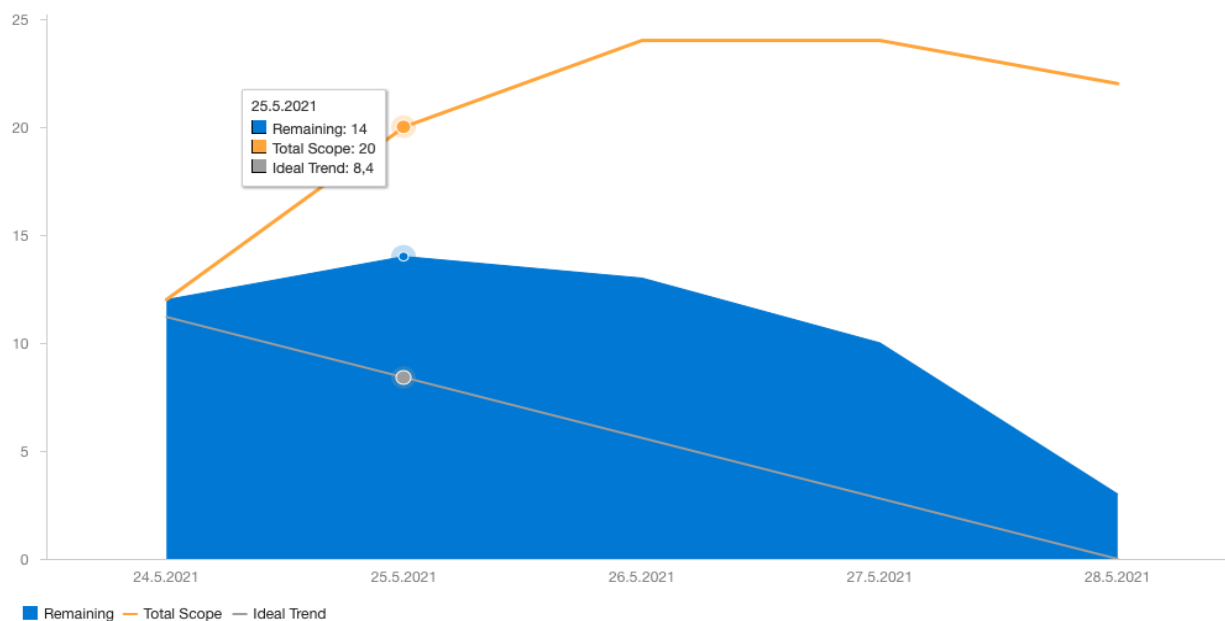




## Azure burndown chart

24.5.2021 - 28.5.2021

Completed 100%

Average  
burndown 2Tasks  
Remaining 0  
Total Scope  
Increase 9

## Prototype - kunde booking

Delpin Booking Rolle Manager Brugere Kalender

Hej Jesper@mail.dk! Log ud

### Booking

Dit navn

Jesper la Cour

Start

03.06.2021 08.30



End

03.06.2021 14.30



Adresse

Grønnedalen 2, 7100 Vejle

Leveres



Må sættes i indkørsel

Opret booking

[Vis mine bookinger](#)

## Post Event Kalender/Controller/Swagger



×

Title

Start

End

Save

Close

```
[HttpPost]
public ActionResult CreateEvent(Event e)
{
    client = new HttpClient();
    client.BaseAddress = new Uri(url);
    client.DefaultRequestHeaders.Accept.Clear();
    client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

    var postTask = client.PostAsJsonAsync<Event>($"Events", e);
    postTask.Wait();
    return Json(e.Id);
}
```

POST

/api/Events

Parameters

Try it out

No parameters

Request body

application/json

Example Value | Schema

```
{
  "id": 0,
  "resourceId": 0,
  "customerId": "string",
  "allby": true,
  "start": "string",
  "end": "string",
  "title": "string",
  "addressId": 0,
  "address": {
    "id": 0,
    "streetAddress": "string",
    "zipcodeId": 0,
    "zipcode": {
      "zipcodeId": 0,
      "city": "string",
      "addresses": [
        null
      ]
    }
  },
  "events": [
    null
  ]
}
```

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "resourceId": 0,
  "customerId": "string",
  "allby": true,
  "start": "string",
  "end": "string",
  "title": "string",
  "addressId": 0,
  "address": {
    "id": 0,
    "streetAddress": "string",
    "zipcodeId": 0,
    "zipcode": {
      "zipcodeId": 0,
      "city": "string",
      "addresses": [
        null
      ]
    }
  },
  "events": [
    null
  ]
}
```

## Get Event Swagger/Controller

### Events

GET /api/Events

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X GET "https://localhost:5001/api/Events" -H "accept: text/plain"
```

Request URL

https://localhost:5001/api/Events

Server response

Code

Details

200

Response body

```
{
  "id": 1,
  "resourceId": 0,
  "customerId": "1E962401-0818-429C-8717-4381178902F3",
  "allDay": false,
  "start": "2021-05-18T10:00:00+02:00",
  "end": "2021-05-18T16:00:00+02:00",
  "title": "Lara Jensen",
  "addressId": 1,
  "address": null
},
{
  "id": 2,
  "resourceId": 0,
  "customerId": null,
  "allDay": false,
  "start": "2021-05-18T10:45:00+02:00",
  "end": "2021-05-19T11:30:00+02:00",
  "title": "Peter Frandsen",
  "addressId": 1,
  "address": null
},
{
  "id": 0,
  "resourceId": 0,
  "customerId": null,
  "allDay": false,
  "start": "2021-05-18T05:00:00+02:00",
  "end": "2021-05-18T06:00:00+02:00",
  "title": "Lara Jensen",
  "addressId": 1,
  "address": null
}
]
```

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 05 May 2021 12:32:25 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header

Example Value | Schema

```
{
  "id": 0,
  "resourceId": 0,
  "customerId": "string",
  "allDay": true,
  "start": "string",
  "end": "string",
  "title": "string",
  "addressId": 0,
  "address": {
    "id": 0,
    "streetAddress": "string",
    "zipCodeId": 0,
    "zipCode": {
      "zipCodeId": 0,
      "city": "string",
      "addresses": [
        null
      ]
    },
    "events": [
      null
    ]
  }
}
```

```
27 [HttpGet]
28 public JsonResult GetCalendarEvents()
29 {
30     client = new HttpClient();
31     client.BaseAddress = new Uri(url);
32     client.DefaultRequestHeaders.Accept.Clear();
33     client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
34
35     HttpResponseMessage response = client.GetAsync("").Result;
36
37     if (response.IsSuccessStatusCode)
38     {
39         var data = response.Content.ReadAsAsync<IEnumerable<Event>>().Result;
40         return Json(data);
41     }
42     else
43         return null;
44 }
```

## Put Event Controller/Azure vindue

```
46 [HttpPut]
47 public ActionResult UpdateEvent(Event e)
48 {
49     client = new HttpClient();
50     client.BaseAddress = new Uri(url);
51     client.DefaultRequestHeaders.Accept.Clear();
52     client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
53
54     var postTask = client.PutAsJsonAsync<Event>($"Events/" + e.Id, e);
55     postTask.Wait();
56     return Json(e.Id);
57 }
```

TEST CASE 163

## 163 Edit testet

Ragnar Svensson Rodenbeck

0 comments

Add tag

State Design

Area DelpinBooking

Reason New

Iteration DelpinBooking\Sprint 2

## Steps



Steps	Action	Expected result	Attachments
1.	Click on 'Jens Madsen'.		actionLog-2021-05-20T12-... ✖
2.	Click on button 'Edit'.		actionLog-2021-05-20T12-... ✖
3.	Click on text box		actionLog-2021-05-20T12-... ✖
4.	Click on 'End'.		actionLog-2021-05-20T12-... ✖
5.	Click on text box		actionLog-2021-05-20T12-... ✖
6.	Click on 'Title Start End'.		actionLog-2021-05-20T12-... ✖
7.	Click on button 'Save'.		actionLog-2021-05-20T12-... ✖
8.	Click on button 'Confirm'.		actionLog-2021-05-20T12-... ✖

Click or type here to add a step