

# TheNext

## 3. SEMESTER EKSAMENSPROJEKT I SOFTWARE DESIGN DAT17-C

Casper Frost Andersen: 14-02-1990

Christian Strunge: 18-11-1992

Matthias Skou: 23-09-1995

Patrick Sirich: 05-08-1998

Vejleder: Cay Holmegaard Larsen

## INDHOLD

Introduktion .....	4
Problemformulering .....	5
Projektopstilling .....	6
Rollefordeling og ansvarsområder .....	6
Arbejds miljø .....	7
Coding standards .....	7
Database .....	8
Patterns .....	10
Udviklingsmodel .....	12
Methodology comparison .....	12
Analyse .....	14
Kunden .....	14
Risikoanalyse & risikoplan .....	15
Identifikation .....	15
RISIKOANALYSE .....	16
RISIKOPLAN .....	17
Kravspecifikation .....	19
Første user stories .....	20
Paper prototype .....	20
Endelige user stories .....	22
Metrics .....	23
Actual estimering .....	23
Burndown chart .....	23
Klassediagram .....	24
Sprint 1 .....	25
Sprint planning .....	25
Rådighedstimer .....	25
Product backlog .....	26
Sprint goal .....	27
Sprint backlog .....	27
US01 .....	28
US02 .....	29
Task descriptions .....	30
US01 .....	30
SD: Opret firma .....	31

SD: Login .....	32
US02 .....	34
End-tests .....	35
Burn down chart .....	36
Sprint review .....	37
Introduktion .....	37
Inkrementeringen .....	37
Demonstration & Feedback .....	38
Diskussion og projektering .....	38
Sprint retrospekt .....	39
Positivt .....	39
Negativt .....	39
Ændringer .....	40
Sprint 2 .....	41
Sprint planning .....	41
Rådighedstimer .....	41
Product backlog .....	42
Sprint goal .....	43
Sprint backlog .....	44
US03 .....	44
Us04 .....	45
Us05 .....	46
US06 .....	47
US07 .....	48
US08 .....	49
US09 .....	49
US10 .....	50
Task descriptions .....	51
US03 .....	51
US04 .....	52
US05 .....	53
US06 .....	54
US07 .....	54
US08 .....	55
US09 .....	56
US10 .....	56
End-tests .....	57

Burndown chart .....	58
Sprint review .....	59
Inkrementeringen.....	59
Demonstration & Feedback .....	60
Diskussion og Projektering .....	62
Sprint retrospekt.....	62
Tasks estimation vs. Tasks actual .....	64
Informal reviews.....	65
Formal reviews .....	65
SCRUM-board .....	66
Gantt-chart.....	67
Task networks.....	68
Glossary.....	69
Konklusion .....	71
Litteraturliste .....	72

## INTRODUKTION

*Af Casper*

I 2018 har miljøvenlighed, lavforbrug og genanvendelse af vores ressourcer aldrig været mere relevant. Flere og flere mennesker begynder at tage ansvar for at mindske ressourcespild, købe ting brugt, og generelt tænke over hvilket aftryk de efterlader på vores jord.

Supermarkeder har efterhånden også fået øjnene op for denne tiltagende forbrugertendens. De fleste butikker har indført en madspildspolitik, hvor madvarer der er ved at nå sidste salgsdato bliver sat markant ned i pris, hvilket både giver goodwill for virksomhederne og øget omsætning i og med, at varerne ikke bare bliver smidt ud.

En af de første til at stadfæste sig i Danmark var Too Good To Go, som er en applikation der tilbyder såkaldte 'lykkeposer', indeholdende mad, bagværk eller andre madvarer fra restauranter, som har mad til overs inden lukketid.-.

Det var blot et spørgsmål om tid, før der kom efterspørgsel efter en tjeneste der tilbød det samme som Too Good To Go, men for lokale supermarkeder, hvis datovarer var ved at udløbe. Your Local er en af de tjenester som tillader brugere at købe madvarer direkte fra butikkerne, i stedet for en samlet 'lykkepose'.

Vi vil i følgende projekt lave en platform for Your Local, som er drevet af The Next, en serviceudbyder vi vil redegøre for i rapportafsnittet om kunden.

## PROBLEMFORMULERING

*Af Casper & Christian*

Vi vil under dette projekt udvikle en applikation for The Next, som tager udgangspunkt i en af deres kunder: "Your Local".

Applikationen skal indeholde følgende funktionaliteter:

- Et firma skal kunne oprette en bruger og kunne:
  - Oprette, redigere og slette tilbud på datovarer tilhørende det firma.
  - Se en oversigt over egne aktive tilbud.
  - Se en statistik over gamle tilbud, der blandt andet inkluderer antallet af en pågældende vare, antallet af solgte varer og den samlede omsætning for dette tilbud.
- En kunde skal kunne oprette en bruger og kunne:
  - Se en liste over aktive tilbud i nærheden.
  - Købe et tilbud på listen, og få en bekræftelse når købet er godkendt.
  - Se en liste over købte tilbud.
- En administrator skal kunne:
  - Redigere og fjerne firmaer og kunder, på nær betalingsinformation.
  - Godkende nyoprettede firmaer (som vil være inaktive indtil da).

Fælles for alle ovenstående brugertyper er, at de vil kunne redigere sine egne kontakt- og betalingsoplysninger.

Vi vil benytte os af en task driven udviklingsmodel, der hovedsageligt benytter elementer fra SCRUM og XP. Derudover vil vi understøtte vores analyser med diverse diagrammer, blandt andet UML-modeller og andre relevante visualiseringer af data, der afbilder vores optimeringsproces.

## PROJEKTOPSÆTNING

### ROLLEFORDELING OG ANSVARSOMRÅDER

*Af Patrick*

PATRICK: PRODUCT OWNER

Patrick har påtaget sig rollen som product owner og kunde-fuldmægtig da han har haft kontakt til kunden som den eneste og har fået feedback fra kunden samt er den som forstår sig bedst på hvad kunden vil have. Han har derfor lavet og prioriteret user stories i samarbejde med kunden. Dette har hjulpet os meget i projektets udvikling da der har været minimal ventetid fra man kontakter kunden til de svarer, hvilket har gjort en stor forskel da vi i forvejen var tidspresede. Grundet Patrick's forståelse af kundens ønsker har det været nemmere at rådføre sig hos ham, hvilket har gjort udviklingen mere effektiv og præcis.

CASPER: RAPPORT-ANSVARLIG

Casper har påtaget sig rollen som rapport-ansvarlig, det har derfor været hans opgave at opretholde den overordnede struktur i rapporten og sørge for at layoutet i rapporten har været det samme, samt inddеле de forskellige emner i de korrekte undertitler. Hvis der har været tvivl om hvad der har manglet i rapporten, har det været Caspers ansvar at holde overblikket og derfor sige hvad der mangler at blive tilføjet. Når et afsnit er blevet skrevet færdigt, er det Caspers opgave i sidste ende at godkende afsnittet.

MATTHIAS: KODE- OG TASK-ANSVARLIG

Matthias har påtaget sig som rollen kode & task-ansvarlig, det har derfor været hans opgave at opretholde det store overblik i koden og sørge for at coding-standards bliver overholdt. Hvis de 3 andre skulle have spørgsmål eller tvivl i koden skal Matthias konsulteres først for at sørge for at det kode der bliver skrevet er sammenhængende med det andet kode der i forvejen står og derfor bedst muligt undgår bugs. Når en task var færdig og sendt til review, er det Matthias' opgave i sidste ende at godkende reviewet eller komme med eventuelle ændringer. Derudover har Matthias været ansvarlig for den interne prioritering af tasks.

CHRISTIAN: SCRUM-MASTER & PROJECT MANAGER

Christian har påtaget sig rollen som SCRUM-master hvilket har betydet at han har haft ansvaret for at alle SCRUMs discipliner bliver fulgt korrekt, og har der været tvivl om hvordan det fungerer har man kunnet konsultere Christian for at få afklaret sin tvivl. Han har også sørget for at der ikke har været nogen

distraheringer som har gjort udviklingen langsom. Det har også været Christian som har sørget for at diskussioner omkring estimering hurtigt når en konsensus så der ikke bliver spildt tid på dette.

Han har også påtaget sig rollen som Project-manager hvilket har betydet at han har haft ansvaret for at have det store overblik over alle medlemmer og har der været nogen indvendinger eller utilfredsheder har man kunnet konsultere med Christian for at få dette afklaret på en venlig vis for at sørge for at ingen bliver sure på hinanden. Han har som Project manager også sørget for at alles programmer var på samme version og har stået i spidsen for versionsstyring samt github opsætning.

## ARBEJDSMILJØ

---

### CODING STANDARDS

*Af Matthias*

#### Java:

- Der skal benyttes CamelCase igennem hele produktet på alle navne, klasser såvel som metoder og fields.
  - Klasser skal starte med stort begyndelsesbogstav, også kaldet pascal casing(Upper CamelCase).
  - Metoder og fields skal skrives med småt begyndelsesbogstav, f.eks. 'firstName'.
- Alle betegnelser og navne skal skrives på engelsk, både i database gennem JPA og i eventuelle kommentarer.
- Der må maximum være 100 tegn pr. linje, ellers skal linjen deles op i mindre dele.
- **Kommentarer:**
  - Kommentarer skrives på metoder(herunder Get- og PostMapping), hvor der forklares om metodens funktionalitet, samt hvilke(n) task og user story den har relevans til.
  - Syntaks for kommentarer skal laves som følger:
    - `"/` - Anvendes for enkelte kommentarer
    - `/**`  
*\* indsæt kommentar her*  
`*/` - Anvendes i forbindelse med længere kommentarer og metodebeskrivelser.
- Alt kommunikation med Spring skal foregå med `@GetMapping` og `@PostMapping`, i stedet for fulde REST API(Put og Delete)
- Alt data som skal sendes eller hentes gennem Spring, skal foregå gennem Model og ikke `@RequestParam`(medmindre andet aftales indbyrdes i gruppen).

#### Database:

- Databasenavne skal bestemmes igennem JPA og vil ikke være nødvendige at ændre, derfor er det vigtigt at Java-coding standards er blevet overholdt, så det ikke giver problemer i databasen senere hen.
- Database-opsætning foregår udelukkende gennem JPA, og hvis kolonner/tabeller eller andet skal tilføjes til databasen, skal dette kun foregå gennem JPA.
- Alle medlemmer arbejder med deres respektive login til database.
- `application.properties` hibernate skal sættes til update og må kun ændres med accept fra resterende gruppemedlemmer.



## HTML:

- Attribut-navne skal altid være med små bogstaver.
- Der skal ikke være mellemrum ved lighedstegn, f.eks: 'class="classExample"'
- Navne der bliver lagt op igennem model i Spring, skal have sigende navne og kun være med små bogstaver - f.eks: 'company'.
- Scripts skal placeres som det sidste indenfor <body>
- Link til stylesheets og andet, skal placeres i <head>
- Generelle kommentarer og lignende placeres øverst i filen, før <!DOCTYPE>
- Gentagende kode skal så vidt muligt, placeres i eksterne HTML-filer og kaldes med <div th:replace> i de respektive filer

---

## DATABASE

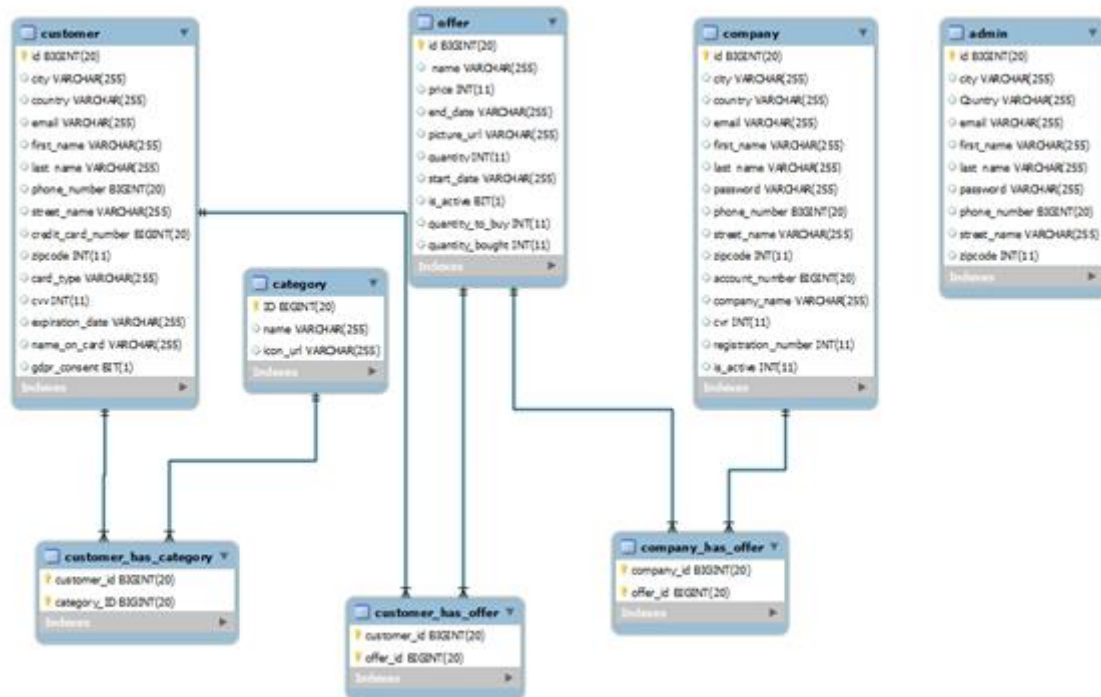
*Af Patrick*

Vores database er hostet på en privat server i København, og vi har fuld adgang til vores del af serveren og er derfor ikke afhængig af en anden person når det kommer til databasen. Den eneste risiko vi har ved denne server er strømnedbrud – dog er chancen for strømnedbrud meget minimal. Vi har valgt at have vores database på en online server for at sørge for at alle er på samme plan hele tiden og at vi alle arbejder med det samme data for at undgå forvirring.

ER-diagrammet ovenover er en afbildning af vores database og demonstrere den datastruktur vi har i vores program. Vi har i vores program benyttet os af en API kaldet JPA som hjælper med at sætte databasen op, hente og manipulere data fra databasen.

Vi har i starten af udviklingsfasen taget udgangspunkt i dette ER-diagram da vi skulle sætte datastrukturen op igennem JPA. Ved hver ændring i vores klasser har vi også holdt ER-diagrammet opdateret.

Brugen af JPA har gjort hele udviklingsprocessen meget nemmere da vi ikke har skullet tænke over at opdatere direkte (fx. via queries) i databasen gennem et DBMS såsom MYSQL Workbench når vi har haft ændringer, men kun i de klasser hvor den overordnede struktur er defineret. Ydermere, i kraft af at vi har brugt JPA, har det kun været nødvendigt at ændre i vores demo.application klasse når der skal opdateres i databasen.



1.1: ER-diagram over databasestrukturen

---

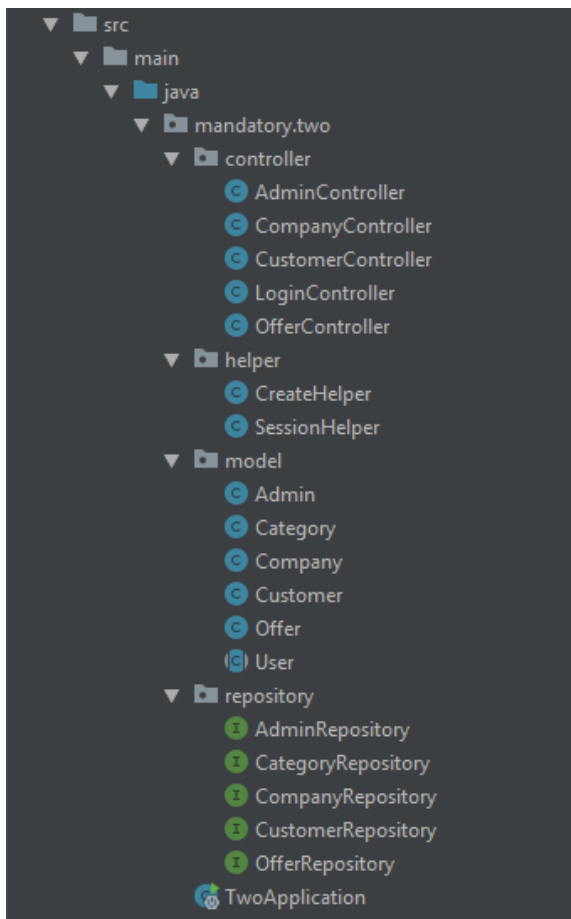
## PATTERNS

Af Matthias

Vi har i forlængelse af vores Coding-standards aftalt fra begyndelsen af projektet, at design-pattern MVC(Model - View - Controller) skulle anvendes. Det vil sige at vi har samlet alle vores Controller-klasser i en samlet 'Controller' package og at alle vores Models er samlet i en 'Model' package.

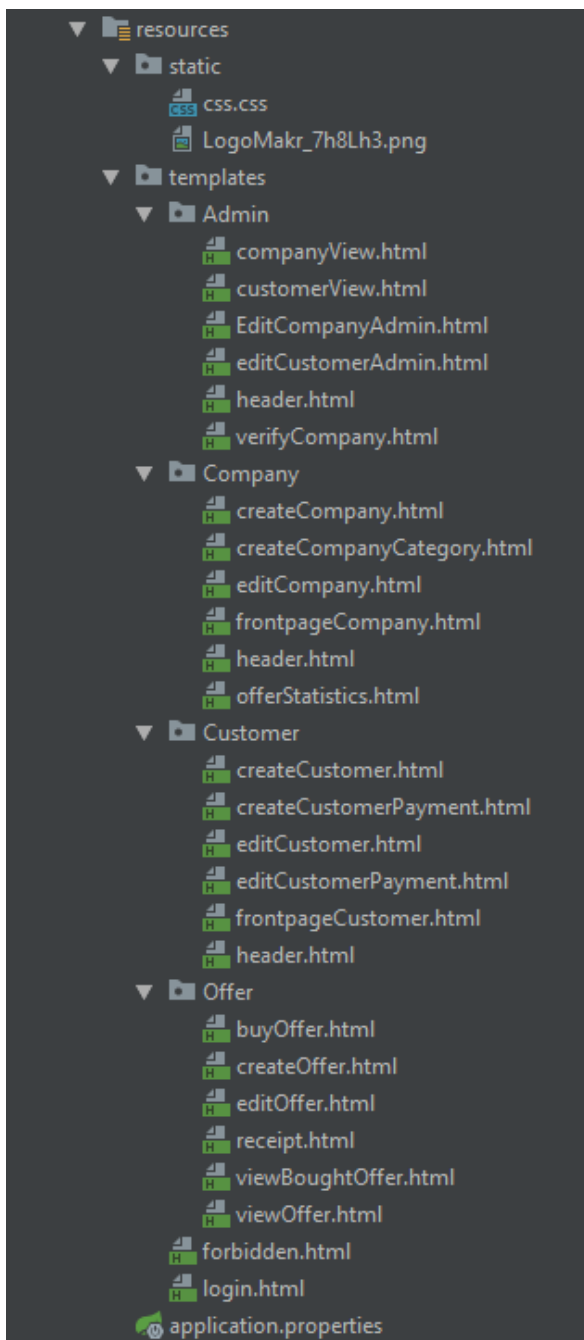
I forlængelse af dette, har vi valgt at oprette en 'Repository' package, til alle vores repositories der hjælper med JPA forbindelse. Til sidst har vi også en 'Helper' package, hvor vi har smidt klassen der hjælper med at håndtere sessioner(SessionHelper), samt en CreateHelper, der indeholder flere forskellige metoder som hjælper med at hente brugere fra databasen og tjekke om e-mails allerede eksisterer.

Vores 'Model' består af alle de java-klasser, som vi benytter os til at oprette objekter af. Vores 'Controller' består af alle Controller-klasserne, hvor vi skriver alle vores @GetMapping og @PostMappings - disse controllere står for alt kommunikation mellem HTTP og vores program.



1: packageinddelingen for programmets klasser

Derudover har vi samlet alle vores HTML-filer (vores views) under Template mappen, hvor de er blevet inddelt i følgende mapper: Admin, Company, Customer og Offer. Vores login- og 403-HTML befinder sig i /all. Alle vores HTML-filer i View er det som skal vises til brugeren på applikationen - disse filer viser alt den information vi sender med fra vores Controllere og fungerer som en gateway til at få bruger-input sendt med tilbage til controllere.



2: HTML-filerne inddelt under relevante tilhørsforhold

Det har været en stor fordel at benytte os af MVC-pattern, da det har givet et godt overblik i programstrukturen(separeret visning). Mappe-inddelingen gjorde det nemmere at navigere rundt i programmet når der skulle programmeres. Derudover var alle gruppens medlemmer indforstået med, hvad de forskellige packages indebærer(Model, view, controller), så man hurtigt kunne overskue hvor funktionaliteter var blevet tilføjet til programmet, af det andet programming-pair. Derudover hjalp MVC også til at man på forhånd vidste, hvor de forskellige funktionaliteter skulle tilføjes for at passe ind i MVC-strukturen.

En af de største fordele ved at benytte sig af MVC, var at man kunne arbejde to forskellige pair-programming par, på samme task eller sideløbende på forskellige tasks, uden at det ville skabe merge-konflikter senere

hen. Dette sparede os meget tid løbende da vi ikke var nødt til at arbejde i eksterne klasser/filer, men blot kunne arbejde i hver vores controller/view uden problemer.

## UDVIKLINGSMODEL

*Af Christian og Casper*

### METHODOLOGY COMPARISON

Vi vil i dette afsnit vise en tabel over de agile udviklingsmodeller vi har arbejdet med, hvor elementerne under hver model er forklaret og stillet op mod hinanden. Herefter vil vi redegøre for, hvilke elementer vi vælger fra hvilke modeller, og til sidst lave en opsamling over hvordan udviklingsmodellen vil se ud.

	XP	SCRUM
Udviklingstilgang	Interaktive inkrementeringer	Iterative inkrementeringer
Udviklingsperiode	1 – 6 uger	2 – 4 uger
Teamstørrelse	Mindre teams	Alle størrelser
Projektstørrelse	Mindre projekter	Alle projekter
Kommunikation	Stand-up meeting	Stand-up meeting
Fleksibilitet	Tasks kan byttes fra product backlog, så længe den ikke er startet	Tasks i sprint backloggen er fastlåst, indtil sprinten er færdig
Drive	Test-driven	Task-driven
Arbejdsgang	Pair programming	Individuelt
Arbejdstider	Kun on-site	Valgfrit

I tabellen ovenfor sammenligner vi to agile udviklingsmodeller: Scrum og XP. Begge modeller har meget til fælles, men indeholder også nogle distinktive forskelle. Vores projekt er relativt småt, og vi er et lille team – derfor ville det være nærliggende at vælge XP som udviklingsmodel ved første øjekast.

XP har dog sine restriktioner i forhold til arbejdstider, da den kun tillader arbejde i et specifikt tidsrum. Vi vil derfor benytte elementer fra begge udviklingsmodeller, som passer til vores arbejdsgang.

Vores udviklingsmodel	
Udviklingstilgang	Iterative inkrementeringer (SCRUM)
Udviklingsperiode	1-6 uger (XP)
Teamstørrelse	Mindre teams (XP)
Projektstørrelse	Mindre projekter (XP)
Kommunikation	Stand-up meeting (begge)
Fleksibilitet	Tasks i sprint backloggen er fastlåst, indtil sprinten er færdig (SCRUM)
Drive	Task driven (SCRUM)
Arbejdsgang	Pair programming (XP)
Arbejdstider	Valgfrit (SCRUM)

Baggrunden for vores udviklingsmodel lyder således:

- **Udviklingstilgang:** Vi har valgt at aflevere iterative inkrementeringer, i stedet for interaktive som brugt i XP, da vi vurderer at der ikke umiddelbart er behov for større ændringer under udviklingsperioden.
- **Udviklingsperiode:** Vi har pga. vores deadline været nødsaget til at læne os op af XP's korte udviklingsperiode, selvom en af vores sprints var kortere end en uge.
- **Teamstørrelse:** Begge udviklingsmodeller tillader et mindre team at anvende dem
- **Projektstørrelse:** Især XP henvender sig til mindre projekter, da kravene for projektet løbende kan tilføjes og fjernes/ændres
- **Kommunikation:** Et centralt omdrejningspunkt for begge modeller er stand-up mødet. Det er derfor en uomgængelig del af vores arbejdsgang.
- **Fleksibilitet:** Som nævnt i udviklingstilgangen regner vi ikke med, at der kommer radikale ændringer under udviklingsperioderne. Derfor synes vi det vil være en fordel for os at låse de tasks vi udvikler, så vi ikke behøver bekymre os om løbende ændringer.
- **Drive:** Vi vælger en task driven udviklingsmodel, da kunden har nogle konkrete krav til applikationen.
- **Arbejdsgang:** Pair programming er en metode vi har valgt at tage i brug, netop fordi det skaber en sikkerhed indenfor kvalitetssikring og effektivisering af udviklingen. Det nedsætter den overordnede rådighedstid med 50%.
- **Arbejdstider:** Vi følger ikke XP version af dette, da det i vores situation ikke ville være hensigtsmæssigt - blandt andet fordi der er medlemmer af gruppen der har nogle faste tider, de ikke er til rådighed. Da vi ikke har lavet et lignende projekt førhen, ville vi gerne åbne op for muligheden for at arbejde off-site, for at kunne imødekomme vores mål.

### KUNDEN

*Af Patrick*

TheNext er en SaaS og PaaS virksomhed som leverer finansielle produkter på et globalt plan og fokuserer på banking-økosystemer men vil dog i fremtiden inkludere finansielle services såsom kreditkort, betaling, finansiering etc. De vil i fremtiden også have muligheden for at tilbyde end-to-end services som UX/UI og helt hen til transaktioner, ledger management, compliance management og rapportering – alt sammen uafhængigt af de allerede eksisterende løsninger.

TheNext fokuserer på at forbedre kundeoplevelserne og at skabe værdi for slutkunden ved at benytte sig af de nye teknologiske muligheder og partner op med tredjeparter som kan hjælpe med at skabe den her værdi. TheNext forretningsmodel følger princippet: *"€ flows when customer value is created"*

TheNext's primære teknologi-partner er TCS som gør at de har adgang til et højt antal af patenterede og IP-beskyttede løsninger samt adgang til et stort netværk af kompetence.

TheNext er skabt af Jan Sirich som har arbejdet i Nordea i 27 år og sluttede i December 2017 for at starte TheNext ApS hvor han kan udnytte sin store forståelse for banking, teknologi, bankernes tankegang og deres kunder. Jan var b.l.a. i spidsen for at levere Netbank og mobil banke i hele norden. Denne teknologiske viden som Jan har, gør at han som kunde har en høj forventning til slut-produktet og ved hvordan udviklingsprocessen foregår. Dette betyder at vores forklarelse og termer er mere avancerede end hvad vi normalt ville bruge, når vi snakker med kunden. Dette har også gjort at startfasen har været nemmere end normalt da vores kunde ved præcis hvad han vil have fra starten af og i forvejen ved hvad der er muligt og umuligt, samt forstår sig inden for design af en applikation.

## RISIKOANALYSE & RISIKOPLAN

Af Patrick & Matthias

I dette afsnit vil vi identificere diverse risikomomenter hvor der er en risiko for at slutproduktet ikke lever op til forventningerne og krav. Derefter vil vi lave en risikoplan hvor vi tager de risiko som har et moment på over 10 og planlægge præventive tiltag samt løsningsforslag samt de ansvarlige for disse.

### IDENTIFIKATION

I nedenstående tabel vil vi identificere de risikomomenter som har en indvirkning på projektet slutprodukt.

Risikomomenter:	
Projektrisiko	<ul style="list-style-type: none"><li>• Misforståelse af krav til kunden</li><li>• Drastisk ændring af krav mod slutningen af projektet</li><li>• Mangel på lokale</li></ul>
Personalerisiko	<ul style="list-style-type: none"><li>• Længerevarende sygdom</li><li>• Gruppemedlem frafalder projektet</li><li>• Tidsoverskridelse i sprints</li></ul>
Kunderisiko	<ul style="list-style-type: none"><li>• Nedskæringer i budget</li><li>• Nedlukning af virksomhed</li></ul>
Ressourcerisiko	<ul style="list-style-type: none"><li>• Hardwarenedbrud</li><li>• Tab af data (overskrivning af data etc.)</li><li>• Fejl i version control</li><li>• Servernedbrud</li></ul>
Estimeringsrisiko	<ul style="list-style-type: none"><li>• Misforståelser i planlægning</li><li>• Større udfordringer i koden</li><li>• Tests viser store fejl</li><li>• Mindre afvigelser i tidsestimering</li><li>• Større afvigelser i tidsestimering</li><li>• Uforudsete sideløbende studierelaterede arbejdsopgaver</li></ul>
Produktrisiko	<ul style="list-style-type: none"><li>• Et produkt der ikke er brugervenligt nok</li><li>• Fejlkommunikation der resulterer i manglende/forkerte features</li></ul>



## RISIKOANALYSE

Efter identifikationen, vil vi analysere sandsynligheden for at risiciene hænder, hvor stor en konsekvens disse vil have på projektet, og ud fra disse udregne et produkt af, hvor stor en indvirkning de vil have på vores projekt:

Risikoanalyse			
Risikomoment	Sandsynlighed (skala: 1-5)	Konsekvens (1, 3, 7, 10)	Produkt (Sandsynlighed x Konsekvens)
Misforståelse af krav til kunden	3	7	21
Drastisk ændring af krav mod slutningen af projektet	1	7	7
Længerevarende sygdom hos gruppemedlem	1	10	10
Gruppemedlem frafalder projektet	1	10	10
Tidsoverskridelse i sprints	3	7	21
Nedskæringer i budget	1	7	7
Nedlukning af virksomhed	1	10	10
Hardwarenedbrud	2	3	6
Tab af data (overskrivning af data etc.)	2	10	20
Fejl i version control	2	7	14
Servervedbrud	1	7	7
Misforståelser i planlægning	2	7	14
Større udfordringer i kode	3	7	21
Tests viser store fejl	2	3	6
Mindre afvigelser i tidsestimering	5	1	5
Større afvigelser i tidsestimering	2	7	14
Sideløbende studierelaterede arbejdsopgaver	4	7	28
Manglende lokale	3	1	3

## RISIKOPLAN

Af Patrick & Matthias

Efter vi har fået identificeret vores risikomomenter og udregnet hvor stor en indvirkning de muligvis kan have på vores projekt, kommer vi her med en løsningsmodel for, hvordan vi kan undgå eller løse problemstillingerne. Vi har taget udgangspunkt i de risikomomenter der har et risikoprodukt på over 10, og derfor har en markant indvirkning i slutproduktet som gør det værd at analysere:

Risikomoment	Præventive tiltag	Ansvarlig	Løsningsforslag	Ansvarlig
Misforståelse af krav til kunden	Hav løbende dialog med kunden, og sørg for at stille uddybende spørgsmål	Project owner	Hvis der er opstået en misforståelse mellem kunden og os, vurderes det om funktionen skal implementeres eller forkastes.	Project owner og kunde
Gruppemedlem frafalder projektet	Klare retningslinjer og kommunikation i gruppen. Støt og hjælp hinanden.	Hele gruppen	Den resterende gruppe aftaler indbyrdes hvordan projektet fortløbende gribes an, og laver de nødvendige tiltag for at projektet kan færdiggøres.	Den resterende gruppe
Tidsoverskridelse i sprints	Vær realistisk omkring estimering, kend vores begrænsninger og arbejd systematisk mod målet.	Hele gruppen	Gruppen skal analysere fejlestimeringen, og om tiden kan fordeles bedre.	Hele gruppen
Tab af data (overskrivning af data etc.)	Sørg for at have backups på flere forskellige platforme: lokalt, GitHub, server	Hele gruppen	Alt afhængig af tabets omfang og tidspunkt i processen, vurderes det om arbejdet skal overføres til næste sprint.	Hele gruppen
Fejl i version control	Klare guidelines for hvordan vi holder filers version opdateret. Konsultere med teamet omkring baseline og version, inden at man laver større ændringer.	Den individuelle og hele gruppen.	Stop arbejde indtil versionsstyring er på plads igen. Backtrack for hvordan fejlen er opstået og arbejd sig igennem derfra.	Hele gruppen

Større afvigelser i tidsestimering	Benytte sig af flere forskellige estimeringsværktøjer. En dybdegående analyse af projektets omfang, for bedre forståelse af estimering fra starten af projektet. Sammenligne med tidligere projekter samt benytte sig af tidligere erfaringer.	Hele gruppen	Efter afsluttet sprint vil teamet foretage en reestimering af tasks og eventuelt justere projekteringen af projektet. Indlede en dialog med kunden under sprint review, om hvordan projektet fremadrettet skal tilpasses.	Hele gruppen og product owner
Sideløbende studierelaterede arbejdsopgaver	Planlægning af kendte opkommende/igangværende studieprojekter. Prioritering af disse studieprojekter.  Koordinere indbyrdes hvordan vi skal imødekomme disse sideløbende opgaver.	Hele gruppen	Hvis arbejdsopgaverne opstår midt i en sprint, kan vi forhøre os om udsættelse på opgaven, så det ikke påvirker vores planlagte sprint.	Hele gruppen
Misforståelser i planlægning	Konstant kommunikation samt klare regler omkring mødetid og dage. Der skal gives klar besked forinden, så resten af gruppen kan lave en plan for dagen.	Hele gruppen	Medlemmet der ikke kan arbejde den dag skal selv sørge for at medlemmets tasks for dagen er blevet overtaget af en anden eller at en anden plan er lagt forinden.	Individuelle gruppemedlemmer
Større udfordringer i kode	Læs op på stoffet hjemmefra, inden koden påbegyndes. Sørg for at have lavet acceptance test forinden programmeringens påbegyndelse	Hele gruppen	Søg hjælp hos hinanden eller hos andre grupper. Hvis løsningen ikke er opnåelig, diskuteres andre løsninger.	Hele gruppen

## KRAVSPECIFIKATION

*Af Patrick*

Vi har været i kontakt med virksomheden TheNext som gerne vil have udviklet denne case:

1. Firma logger på TheNext platformen
2. Firma indtaster deal, som de vil tilbyde til dét community TheNext har aftale med ifht. firmaet logon
3. TheNext sender via API deal til Your Local, som får lagt tilbuddet på Your Local (Your Local er samarbejdspartner på TheNext platformen)
4. TheNext pusher deal tilbud til community medlemmer, som har givet tilsagn (consent) og interesse for dén type tilbud, med link til Your Local tilbuddet
5. Your Local sender API notification til TheNext for hver instans af dealen der er blevet indløst hos en butik (€ flows when real value is created for customers)

I skal designe en TheNext Butiks-database (merchant DB), som bl.a. indeholder følgende data felter:

- Unikt ID
- Navn
- Adresse (incl. google link)
- Åbningstider
- Kontaktoplysninger
- Skal være skalérbart til at kunne indeholde 100.000 Unikke ID

I skal designe en TheNext Community-database, som bl.a. indeholder følgende data felter:

- Unikt ID
- Navn
- Consent data (GDPR-data krav)
- Øvrige adresse og kontakt oplysninger
- Kategorier som kunden vil modtage tilbud fra
- (Geografi radius som kunden vil modtage tilbud fra)
- Skal være skalérbart til at kunne indeholde 1.000.000 Unikke ID

Deal der indtastes af butik skal indeholde følgende data felter:

- Overskrift
- Kort beskrivende tekst
- Pris
- Start data/tid
- Slut data/tid
- Communities som de vil sende deal til (TheNext, sørger for at administrere hvilke de skal have adgang til)

Derudover skal i lave en brugergænseflade til Merchant og community.

Når man opretter sig som bruger skal man tilføje sine interesser og derved blive en del af et community som modtager specifikke tilbud fra butikker som kan levere disse tilbud.

---

## FØRSTE USER STORIES

*Af alle*

Efter kundens feedback af paper prototype har vi udviklet nogle user stories, som danner ramme for de funktionaliteter applikationen skal kunne. User stories er skrevet i en "som x skal jeg kunne y, så/for at z" form, for at gøre forståelsen mellem udvikler og kunde lettere. Disse user stories vil vi bryde op i tasks, som vil blive estimeret, og til sidst præsentere for kunden, som vil prioritere hvilke user stories der vægter højest:

User stories
Som kunde og firma skal jeg kunne logge ind, og have forskellige rettigheder.
Som kunde skal jeg kunne oprette en bruger, hvor mine interesser bliver gemt.
Som firma skal jeg kunne oprette en bruger hvor jeg kan oprette nye tilbud som vises til kunder.
Som kunde skal jeg kunne se en oversigt over tilbud i mit nærområde og sortere i disse ud fra kategorier.
Som kunde skal jeg kunne købe et tilbud.
Som kunde skal jeg modtage push-meddelelser om tilbud indenfor en geolokation på 5 km.
Som firma skal jeg kunne se en oversigt over hvor mange tilbud der er blevet købt.

---

## PAPER PROTOTYPE

*Af Patrick*

Som det første efter at have lavet user stories, udviklede vi en paper prototype, som blev vist til kunden. Kunden var positiv over vores paper prototype, og havde kun få ændringer, såsom at kategorier og persondata ved oprettelse af en ny bruger skal være på samme side, og ikke være 2 individuelle sider. Denne positive feedback gjorde, at vi havde et stærkt udgangspunkt i hvad kunden ønskede af funktionaliteter og layout, og derfor nemt kunne gå i gang med udviklingen da vi kom dertil. Kunden havde dog en enkelt bemærkning ift. det umiddelbare layout – han var ikke vild med hjertet, hvis funktion var at kunne favorisere bestemte virksomheder, men eftersom at det ikke havde en fundamental funktionalitetsmæssig indvirkning, havde det ikke en negativ effekt på vores fremtidige udvikling.

Billedet nedenunder er en collage af de forskellige sider vi lavede til vores paper prototype.



3: kollage over paper prototypen

---

## ENDELIGE USER STORIES

Af Casper

I forlængelse gennemgangen af paper prototypen, fik vi diskuteret vores første user stories og præciseret, hvilke funktionaliteter som kunden kunne tænke sig. De er blevet redegjort for nedenunder, i prioriteret rækkefølge:

Prioritet	User stories
1	Som firma skal jeg kunne oprette en bruger, så jeg kan logge ind senere med relevant indhold og rettigheder.
2	Som kunde skal jeg kunne oprette en bruger, hvor mine interesser bliver gemt, og logge ind, med relevante rettigheder, så jeg ikke behøver at finde de forskellige kategorier efter hvert login.
3	Som firma skal jeg kunne oprette nye tilbud, som skal vises til kunder, så kunderne har mulighed for at købe vores tilbud.
4	Som ansat hos Your Local skal jeg have administrative rettigheder over firmaer og kunder, så jeg kan godkende firmaer og redigere og slette brugerdata ift. GDPR.
5	Som kunde skal jeg kunne købe et tilbud, så jeg er sikker på at mit køb er blevet godkendt.
6	Som firma skal jeg kunne se en oversigt over hvor mange tilbud der er blevet købt, så jeg kan holde statistik over mine salg.
7	Som kunde skal jeg kunne se en oversigt over tilbud i mit nærområde, og sortere i disse ud fra kategorier, så jeg kun ser relevante tilbud ift. min placering.

## METRICS

*Af Matthias og Patrick*

---

### ACTUAL ESTIMERING

Vi har valgt at opsætte et skema over vores estimeringer vs. vores egentlig tidsforbrug. Vi har løbende registreret de antal minutter vi brugte på hver task og har sammenlignet det med vores estimerede værdier. Dette har hjulpet os med at få et overblik over hvor lang tid vi egentligt har brugt på hver task. Dette har været til stor gavn for os, da vi løbende er kommet ud for at skulle re-estimere vores tasks til næste sprint, hvor vi har sammenlignet tasks fra tidligere sprints for at komme med en mere præcis estimering.

---

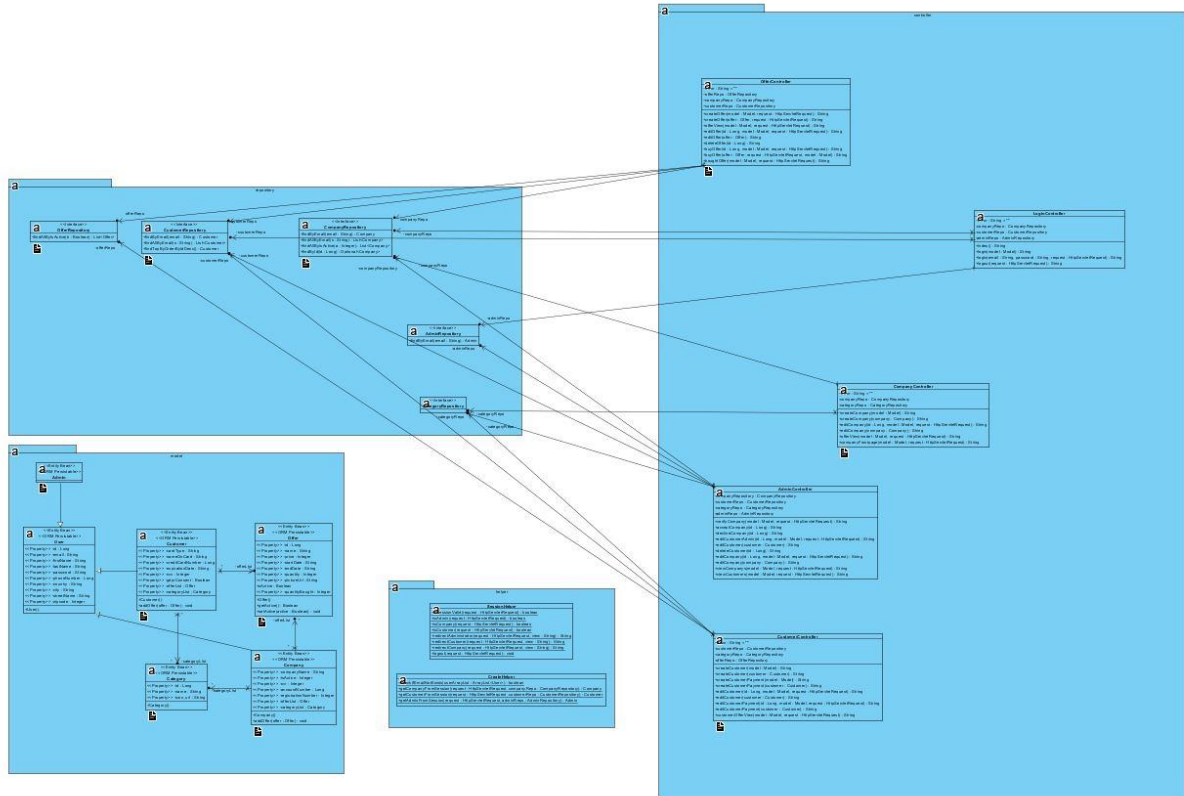
### BURNDOWN CHART

Vi har udover vores estimeringer vs. egentlig tidsforbrug, lavet Burndown Charts for hver sprint. Vi har udregnet en velocity inden start af sprint for at finde ud af hvor meget vi i gennemsnit skal gennemføre pr. dag for at sørge for, at vi bliver færdige med produktet til den aftalte tid. Efterfølgende har vi igennem vores sprint opdateret vores Burndown chart for at give et bedre overblik over hvor langt vi er i sprinten og hvorvidt om vi er forud eller bagud ifølge planen. Dette redskab har været en god komplementering til vores actual estimering, da vi kan se på begge metrics og danne et samlet billede af, hvor præcise vores estimeringer har været. Dette er en vigtig måling, som kan give et bedre udgangspunkt for estimeringer i fremtiden.



## KLASSEDIAGRAM

Af Patrick



### 4. Klassediagram

Vi har i vores projekt fulgt et MVC pattern hvilket afspejles i vores klassediagram hvor vi har inddelt de forskellige klasser i de respektive packages de hører til. Vores klassediagram giver en god visualisering af vores programs arkitektur og hjælper os derfor med at forstå hvilke klasser der arbejder sammen. Vi ved at vores kunde har planer om at fortsætte med projektet efter deadline og dette klassediagram vil derfor hjælpe fremtidige udviklere med at forstå hvilke tanker vi har gjort os i programmets arkitektur.

## SPRINT 1

### SPRINT PLANNING

*Af Christian, Matthias og Casper*

### RÅDIGHEDSTIMER

Under denne sprint, har holdet sat følgende timer til rådighed:

NAVN	DAG 1	DAG 2	DAG 3	DAG 4	DAG 5	SAMLET
Matthias	2	2	4	6	4	18
Casper	2	2	4	0	4	12
Christian	2	2	4	6	4	18
Patrick	0	2	0	0	4	6
I ALT	6	8	12	12	16	54

Eftersom vi benytter os af pair-programming, vil det samlede antal rådighedstimer blive delt med to – vi skal altså tage højde for, at vi har 27 arbejdstimer til denne sprint, som vi kan planlægge efter.

## PRODUCT BACKLOG

Vi har følgende user stories i vores product backlog, i prioriteret rækkefølge:

User story ID	User stories	Estimerede timer
US01	Som firma skal jeg kunne oprette en bruger, så jeg kan logge ind senere med relevant indhold og rettigheder.	15
US02	Som kunde skal jeg kunne oprette en bruger, hvor mine interesser bliver gemt, og logge ind, med relevante rettigheder, så jeg ikke behøver at finde de forskellige kategorier efter hvert login.	10
US03	Som firma skal jeg kunne oprette nye tilbud, som skal vises til kunder, så kunderne har mulighed for at købe vores tilbud.	7
US04	Som ansat hos Your Local skal jeg have administrative rettigheder over firmaer og kunder, så jeg kan godkende firmaer og redigere og slette brugerdata ift. GDPR.	11
US05	Som kunde skal jeg kunne købe et tilbud, så jeg er sikker på at mit køb er blevet godkendt.	5
US06	Som firma skal jeg kunne se en oversigt over hvor mange tilbud der er blevet købt, så jeg kan holde statistik over mine salg.	2
US07	Som kunde skal jeg kunne se en oversigt over tilbud i mit nærområde, og sortere i disse ud fra kategorier, så jeg kun ser relevante tilbud ift. min placering.	11

Som vi har fastslået har vi 27 timers arbejde til rådighed. Derfor vil vi tage US01 og US02 med i vores sprint backlog i vores første sprint, som samlet set er anslået til 25 timer.

---

## SPRINT GOAL

Målet for vores første sprint vil fokusere på to hovedpunkter:

1. Vi kan aflevere et produkt, som gør det muligt for firmaer at oprette en bruger på applikationen, hvor informationerne bliver korrekt gemt i databasen. Vi vil lave en grafisk brugergrænseflade, hvor der hhv. er en underside til firmadata og en til konto-oplysninger. Når et firma opretter sig kan de derefter tilknytte forskellige kategorier til deres bruger. Derudover vil der være mulighed for at redigere disse brugeres stamdata. Når et firma har oprettet en bruger, vil de få et login som er tilknyttet, med de korrekte administrative rettigheder.
2. Vi vil derudover aflevere et produkt, som gør det muligt for kunder at oprette en bruger, hvor kundens interesser bliver gemt til database. Kunden får et login når de opretter en bruger og ud fra dette login får de specifikke rettigheder, som er forskellige fra et firma-login. Der vil også være grafiske sider tilknyttet til oprettelse af en kunde – herunder side til betalings-information, kunde-information (GDPR samtykke i mente) og en underside til deres foretrukne kategorier. Derudover vil det også være muligt for en kunde at redigere sine informationer og præferencer.

## SPRINT BACKLOG

*Af Alle*

Vores tasks er de opgaver, som vi vil arbejde ud fra i et sprint. Disse tasks vil blive estimeret ud fra to planlægningsteknikker: *three point value estimating* og *planning poker*.

Vi vil for hver sprint redegøre for de user stories og tasks, vi har med i vores sprint backlog, som set herunder. Tabellerne indeholder user story navn, med tilhørende tasks i samme kolonne. Derudover har vi begge vores estimeringer for tasken, samt estimeringsproduktet.

## US01

ID: US01	User Story: Som firma skal jeg kunne oprette en bruger, så jeg kan logge ind senere med relevant indhold og rettigheder.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)
US01.1	Firma-, bruger- og tilbudstabel oprettes i databasen.	1	2	1,5
US01.2	Underside til firmadata inklusiv kontooplysninger.	2	2	2
US01.4	Opret gem-funktion for firma til databasen.	3	1	2
US01.5	Tilknytning af kategorier til en firma.	2	2	2
US01.6	Funktion til redigering af stamdata.	4	1	2,5
US01.7	Login-oplysninger gemmes i database.	2	1	1,5
US01.8	Login-funktion der adskiller de forskellige brugere via login.	3	4	3,5
SAMLET:	Det samlede timeantal for estimeringerne	17	13	15

## US02

ID: US02	User Story: Som kunde skal jeg kunne oprette en bruger, hvor mine interesser bliver gemt, og logge ind, med relevante rettigheder, så jeg ikke behøver at finde de forskellige kategorier efter hvert login.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)
US02.1	Kundetabel oprettes i database.	1	1	1
US02.3	Underside til kundeinformation med kategorier.	1	4	2,5
US02.4	Underside til betalingsinformationer.	2	1	1,5
US02.5	Gem-funktion til kunde	3	1	2
US02.6	Funktion til redigering af stamdata.	4	2	3
SAMLET:	Det samlede timeantal for estimeringerne	11	9	10

## TASK DESCRIPTIONS

Af Matthias & Christian

US01

ID	TASK	BESKRIVELSE
<b>US01.1</b>	Firma-, bruger- og tilbudstabel oprettes i databasen.	Alle tabeller skal være oprettet korrekt, og der skal være forbindelse imellem tilbudstabel og companytabel, der skal være kolonner til: ID, e-mail, fornavn, efternavn, password, telefonnummer, konto samt adresseoplysninger og konto-oplysninger.
<b>US01.2</b>	Underside til firmadata inklusiv kontooplysninger.	Man skal kunne indtaste firmadata og dette skal gemmes til DB via en knap, herunder: ID, e-mail, fornavn, efternavn, telefonnummer, firmanavn, CVR og adresseoplysninger.
<b>US01.3</b>	Opret gem-funktion for firma til databasen.	Man skal kunne indtaste kontooplysninger på undersiden for oprettelse af et firma, som skal gemmes til databasen via gem-knap, med følgende informationer: account number og registration number.
<b>US01.5</b>	Tilknytning af kategorier til en firma.	Et firma skal kunne tilknytte en til flere kategorier til deres firma, efter de har udfyldt firmainformationer. Disse kategorier skal gemmes korrekt i en ManyToMany tabel indeholdende id på category og id på firma. Et firma skal også kunne vælge ikke at tilknytte kategorier.
<b>US01.6</b>	Funktion til redigering af stamdata.	<p>Man skal kunne redigere et firmas informationer, hvor de redigerede inputs bliver gemt korrekt i databasen. Email skal ikke være muligt at ændre(da det er bruger-login).</p> <p>Det skal ikke være muligt at indtaste ugyldige inputs, de skal følge anvendte patterns og titles magen til dem der gør sig gældende, når man opretter et firma.</p>
<b>US01.7</b>	Login-oplysninger gemmes i database.	Når et firma opretter sig, skal der på samme underside som de andre informationer, være mulighed for at man indtaster et ønsket password. Dette password skal sammen med e-mail gemmes korrekt i databasen, og vil efterfølgende være det login som de benytter på applikationen. Når et firma gemmes, er det vigtigt at de ikke kan benytte en e-mail som allerede eksisterer for et andet firma.
<b>US01.8</b>	Login-funktion der adskiller de forskellige brugere via login.	<p>Der skal laves en login-side, hvor en bruger af system kan indtaste e-mail og password og logge ind. Disse login-informationer skal stemme overens med informationer i databasen.</p> <p>Hvis login-oplysningerne er forkerte, skal brugeren ikke logges ind og der skal komme en fejlmeddelelse. Når en bruger er logget ind, skal system kunne differentiere hvorvidt om det er et firma eller ej.</p>

## SD: OPRET FIRMA

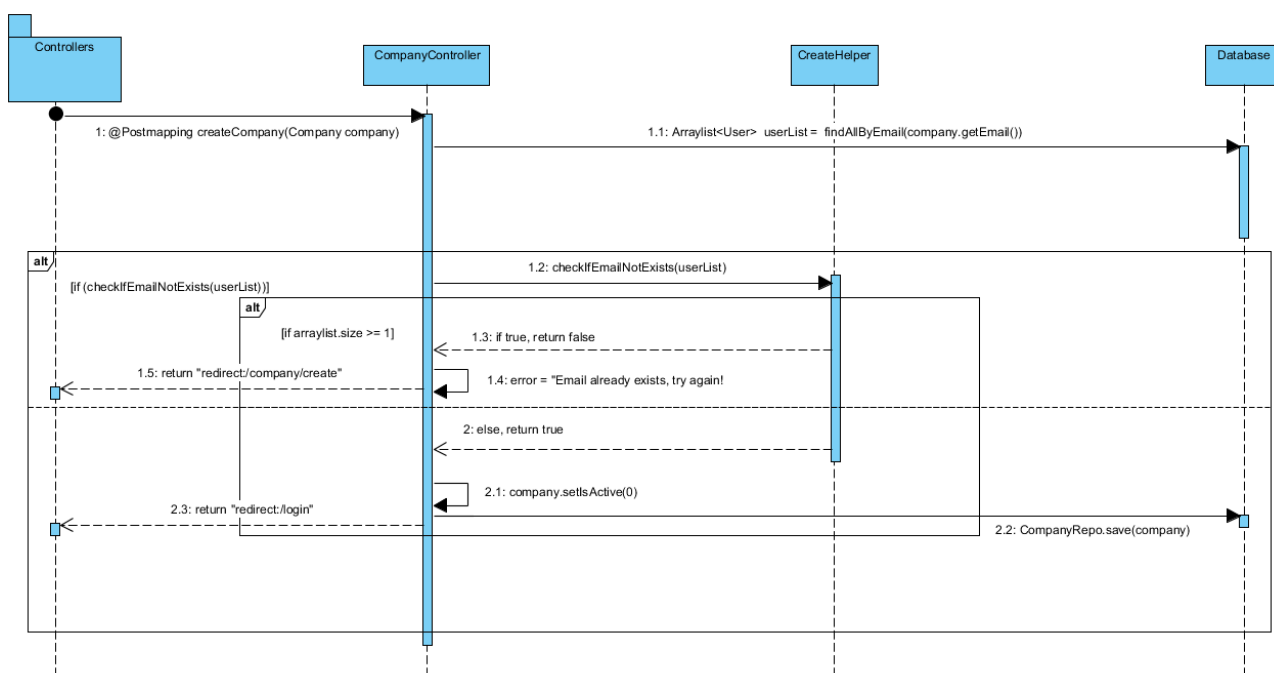
Af Matthias, Christian & Patrick

Dette sekvensdiagram dækker over US01.4, som gør sig gældende når der er blevet oprettet et nyt firma, som skal gemmes i databasen. Først henter vi alle firmaer med en email magen til det nye firmas email. Hvis listen er tom, betyder det at emailen ikke eksisterer i forevejen og vi kan gemme firmaet i databasen, samt sætte dets status til at være inaktiv, indtil en administrator godkender firmaet. Hvis arraylisten har et eller flere firmaer, så er den indtastede email duplikeret og der skal sendes en fejlbesked til brugeren. Denne fejl besked er 1.4 'error', som er en String der er defineret tidligere i CompanyController-klassen.

I vores message 1.1, har vi valgt ikke at tilføje en reply message fra databasen, da vi skriver userList = findAllByEmail – derved er det indforstået, at objektet er blevet fyldt i samme message.

Den venstre 'lifeline' er en package der dækker over flere forskellige Controllers, hvor der skal sendes en Get, afhængigt af hvilket udfald tasken har. I det inderste if-statement, smides data over til en klasse 'CreateHelper', som checker om arraylisten er tom eller ej.

Vi har valgt at lave et sekvensdiagram på denne task, da vi både skal kunne oprette firmaer, kunder og tilbud. Ved at lave et sekvensdiagram for, hvordan vi opretter et firma, har vi automatisk dækket over de resterende 'opret' tasks. Derudover stod vi med det dilemma, at vi var i tvivl om hvordan vi kunne sørge for, at der ikke blev oprettet et firma, med en tilsvarende email til et eksisterende firma. Sekvensdiagrammet ledte til diskussion og gav i sidste ende et klart overblik over, hvordan vi skulle håndtere denne udfordring.



5: SD-diagram for 'opret firma' funktionen.



---

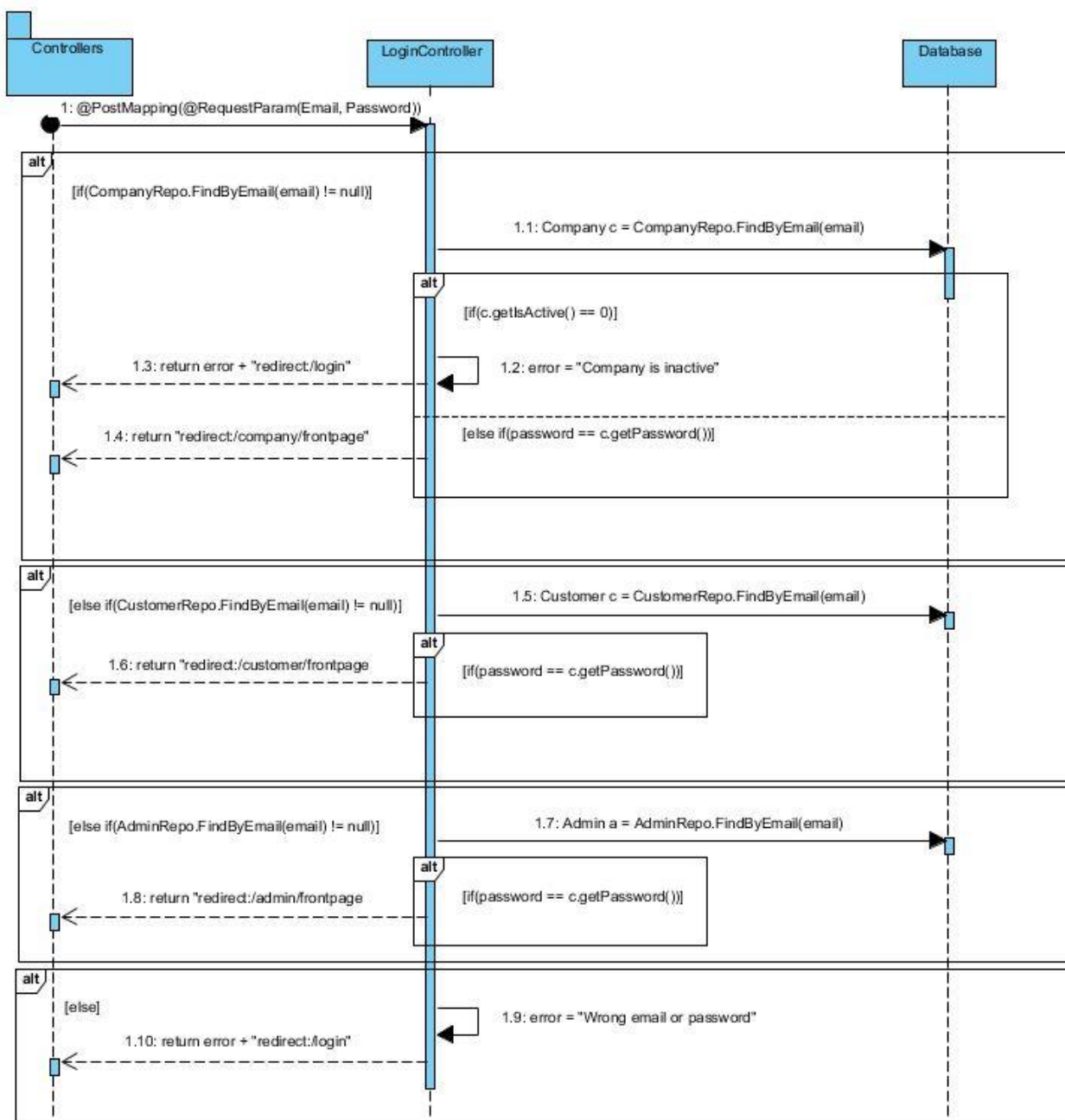
## SD: LOGIN

*Af Matthias*

Herunder ses vores sekvensdiagram for Login-funktionen. Vores 'lifeline' til venstre, skal repræsentere en samlet package, over alle vores controllere. Da vi i løbet af dette sekvens-diagram, returnerer 5 forskellige gange, valgte vi at samle alle controllers i én package for nemmere at kunne holde overblik. Derudover er der 2 steder i diagrammet nævnt 'error'. Dette er en String som er defineret tidligere i LoginControlleren - formålet med 'error' er at beskeden kan smides op på HTML'en igennem thymeleaf, så brugeren af applikationen kan se, hvad fejlen er. Derudover har vi valgt at lave en ny 'alt'-boks for hver if / if-else, i stedet for at samle det til én stor med flere operands. Dette har vi gjort for læsbarhedens-skyld, så man lettere kan finde de forskellige inddelinger.

I vores message 1.1, har vi valgt ikke at tilføje en reply message fra databasen, da vi skriver `Company c = findByEmail` – derved er det indforstået, at objektet er blevet fyldt i samme message.

Sekvensdiagrammet viser hvad der sker, når man forsøger at logge ind med email og kodeord. Først checkes der efter bruger-type(kunde, firma, admin) og derefter checkes der om kodeordet stemmer overens med det eksisterende kodeord i databasen. Hvis det er et firma som forsøger at logge ind, så skal der også checkes om dette firma er aktivt eller inaktivt. Hvis email eller kodeord er ugyldigt, skal dette vises på HTML'en og brugeren skal



5: SD-diagram for 'login' funktionen.

## US02

Af Matthias

ID	TASK	BESKRIVELSE
<b>US02.1</b>	Kundetabel oprettes i database.	Der skal oprettes tabel til kunde i database med de korrekte informationer: city, country, email, firstname, lastname, password, phonenumber, streetname, zipcode og GDPR consent, cardtype, creditcardnumber, cvv, expirationDate, nameOnCard, samt en junction-tabel til category indholdende kategori id og kunde id.
<b>US02.3</b>	Underside til kundeinformation med kategorier.	<p>En underside skal laves med input fields til kundeinformation. Følgende felter skal inkluderes: city, country, email, firstname, lastname, password, phonenumber, streetname, zipcode og GDPR consent checkbox.</p> <p>Disse felter skal garderes mod fejl 40(brugerfejl), således at der kun kan indtastes korrekte oplysninger der følger vores patterns og titles der viser til brugeren, hvilket pattern der skal følges.</p> <p>Derudover skal der laves checkboxes til alle eksisterende kategorier, så kunden kan vælge hvilke kategorier personen har interesse for.</p> <p>Alle disse felter skal gemmes korrekt i databasen, såfremt at kundens input er gyldigt.</p>
<b>US02.4</b>	Underside til betalingsinformationer	Der skal oprettes en ny underside som kunden bliver redirected til efter de har udfyldt alle brugerinformationer og kategorier. Denne underside skal indeholde cardtype, creditCardNumber, cvv, expirationDate og nameOnCard input fields med korrekte patterns og titles så en bruger ikke kan indtaste ugyldige informationer. Disse informationer skal gemmes til databasen i samme kunde-række som er blevet gemt i US02.3.
<b>US02.5</b>	Gem-funktion for kunde.	Der skal implementeres en gem-funktion, der sørger for at informationerne i US02.3 og US02.4 bliver gemt korrekt og at kunden bliver redirected til korrekte undersider efterfølgende.
<b>US02.6</b>	Funktion til redigering af stamdata.	<p>Man skal kunne redigere en kundes informationer, hvor de redigerede inputs bliver gemt korrekt i databasen. Email skal ikke være muligt at ændre(da det er bruger-login). Der skal både laves en underside for redigering af kundens stamdata og en for deres betalingsoplysninger. Begge disse undersider skal være udfyldt med de eksisterende data, når undersiden vises til at starte med.</p> <p>Det skal ikke være muligt at indtaste ugyldige inputs, felterne skal følge anvendte patterns og titles fra opret firma.</p>

## END-TESTS

Af alle

Vi har valgt at benytte en end-test form, som har en følger denne algoritme:

- **HVIS:** hvad skal der ske for, at testen kan igangsættes?
- **NÅR:** kriteriet for testens udfald.
- **SÅ:** testens output.
- **ID:** er skrevet i formatet "USERSTORY-ID.TASK-ID.TEST-ID".

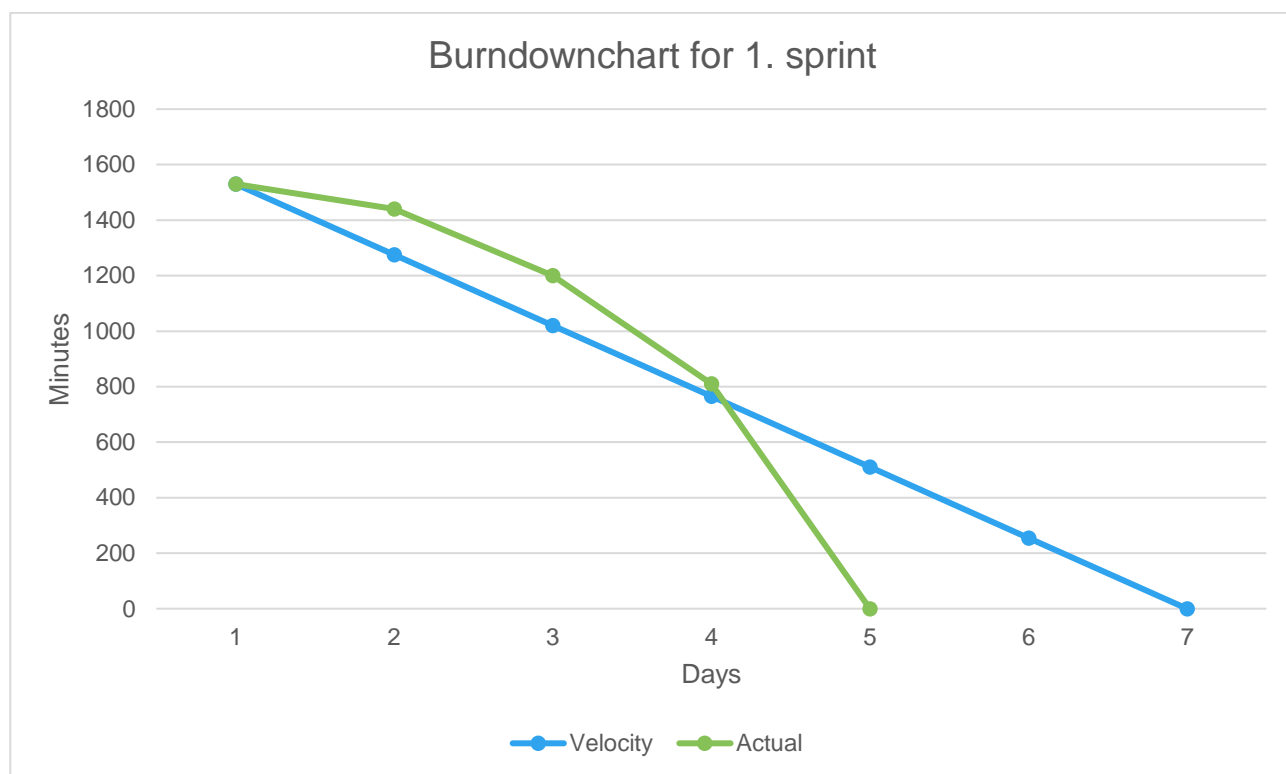
Vi har taget udgangspunkt i en række tasks fra vores US01, for at demonstrere hvordan tests skal udføres:

ID	HVIS	NÅR	SÅ
US01.5.1	Et firma opretter sig og ønsker at tilføje en kategori	Et firma har valgt en ønsket kategori	Bliver den enkelte kategori gemt korrekt i databasen, i en junction med kategori id og firma id
US01.5.2	Et firma opretter sig og ønsker at tilføje flere kategorier	Et firma har valgt flere kategorier	Bliver kategorierne gemt korrekt i databasen i en junction, med en række for hver valgt kategori
US01.5.3	Et firma opretter sig og ønsker ikke at tilføje en kategori	Et firma har ikke valgt nogle kategori(er)	Alt resterende data gemmes korrekt i databasen, men intet data gemmes i junction med kategori id og firma id
US01.6.1	Et firma ønsker at redigere deres brugeroplysninger	En bruger bliver redirected til undersiden for at redigere	Vil alle eksisterende informationer være udfyldt i redigerings-templat
US01.6.2	Et firma har ændret deres ønskede informationer og trykker gem	De udfyldte informationer er gyldige	Overskrives eksisterende data i databasen korrekt
US01.6.3	Et firma har ændret deres ønskede informationer og trykker gem	De udfyldte informationer er ugyldige	Vises der en fejlmeddelelse med det ugyldige data og ændringerne gemmes ikke.
US01.4.1	Et firma opretter en konto	Informationerne er gyldige	Opret bruger i database
US01.4.2	Et firma opretter en konto	Informationerne er ugyldige	Gemmes data ikke, og fejlmeddelelse vises
US01.7.1	Et firma opretter en konto	E-mail allerede findes i database	Gemmes data ikke, og fejlmeddelelse om duplikering vises
US01.8.1	En bruger logger ind på applikationen med et gyldigt login	Login-typen er firma	Programmet genkender login-typen og gemmer det korrekt i en session

<b>US01.8.2</b>	En bruger logger ind på applikationen	Login-typen er ikke et firma	Programmet genkender ikke login-typen og viser en fejlmeddelelse
<b>US01.8.3</b>	En bruger forsøger at logge ind på applikationen	Login-informationerne er ugyldige	Programmet viser meddelelse 'Email or password is incorrect' og brugeren bliver ikke logget ind

## BURN DOWN CHART

Af Casper



6: burn down chart for sprint 1

Vores første burn down chart afspejler en tendens, der bærer præg af opstartsproblemer og et sideløbende projekt. Vi havde sat vores sprint til 5 dage, da vi kalkulerede at der ville være nogle uforudsete arbejdsopgaver, der ikke ville have med dette projekt at gøre, som skulle afsluttes.

Fra dag 1 havde vi allerede opsætningsproblemer med vores IDE og versionsstyring, som gjorde at vi ikke kunne udnytte de timer vi havde sat til rådighed, hvilket forårsagede at vi allerede kom bagud ift. vores ideallinje. Da der var ved at være styr på det, holdt vi vores estimering fra dag 2 til 3, og fra dag 3 til 4 fik vi indhentet det resterende.

Fra dag 4 til 5 valgte vi at samle vores timer, som ellers var delt ud til det andet projekt, til denne sprint. Årsagen til dette var, at de funktionaliteter vi skal implementere i det næste sprint er altafgørende for de, der skal udvikles i næste sprint.

## SPRINT REVIEW

*Af Alle*

Eftersom vi følger en SCRUM-inspireret udviklingsmodel, er sprint review og sprint retrospekt essentielle dele af vores proces. I reviewet vil udviklerne og kunden, eller en repræsentant, holde et uformelt møde, der fokuserer på den inkrementering, som er blevet leveret som produkt af sprintet, hvor især feedback, kommentarer og ændringer er i højsædet.

Patrick vil udover at være product owner i vores projekt desuden også repræsentere kunden, og dermed have dennes fulde opbakning ift. produktets design, funktionalitet og andre relevante nøglepunkter.

Vi følger en disposition over vores sprint review der indeholder følgende:

1. En introduktion til det første Sprint review (punkt 1 vil ikke blive fulgt i fremtidige reviews med mindre andre stakeholders vil være til stede). Introduktionen vil understrege reviewets formål og de deltagende parter roller.
2. Et oplæg fra product owneren der fortæller, hvilke Product Backlog produkter der er blevet færdiggjort (og eventuelle user stories der ikke blev).
3. Udviklerne forklarer kortfattet hvilke problemer der blev mødt, og hvordan de blev løst. Fokus vil være på problemer, som kunden kunne have interesse i, og ikke tekniske komplikationer.
4. En demonstration af inkrementeringen, hvor kunden stiller spørgsmål og giver feedback sideløbende.
5. En diskussion af Product Backloggen og et sandsynligt færdiggørelsestidspunkt for projektet.
6. Alle deltagere planlægger næste skridt, så udviklerne har noget at gå ud fra i deres sprint planning.
7. Mødet hæves.

---

## INTRODUKTION

Formålet med sprint reviews er et uformelt møde, og ikke et statusmøde; alle involverede parter skal have en indgangsvinkel til reviewet som værende en udveksling af idéer og en konstruktiv dialog. Det er derfor vigtigt at understrege, at udviklerne ikke står og skal sælge det produkt de har udviklet til kunden – selvom de naturligvis skal stå på mål for deres tanker og handlinger under sprinten, og produktet deraf. Uenigheder under et sprint review udviser tegn på miskommunikation mellem parterne.

Ligeledes skal kunden eller repræsentanten ikke føle at det, der bliver demonstreret, er et færdigt produkt, som der skal bedømmes som helhed. Kundens job er at kommunikere deres behov så klart som muligt til udviklerne, og komme med indskud, ændringer og idéer til, hvordan de forestiller sig deres kundebase tænker.

Product owneren vil under et review stå for det bindende led mellem kunden og udviklingsteamet. Han vil blandt andet introducere hvilket arbejde der er blevet udført, præsentere oversigten over product backloggen efter teamets demonstration, og sætte en realistisk slutdato for projektet.

---

## INKREMENTERINGEN

Under dette sprint har vi fået færdiggjort to User Stories fra Product Backloggen: US1 og US2. Kundens vigtigste prioriteter for produktet var, at give både firmaer og kunder mulighed for at oprette sig i systemet. Vi har lavet det mest basale skelet for at kunne oprette sig som virksomhed, med tilknyttede kontooplysninger, og et småt udvalg af kategorier som firmaet kan tilknytte sig. Ligeledes kan en kunde nu oprette sig, vælge hvilke kategorier vedkomne er interesseret i, og samtidig give samtykke til at få sine private data behandlet.

---

## DEMONSTRATION & FEEDBACK

Efter demonstrationen af iterationen havde kunden følgende kommentarer:

1. Layout på siderne var som første udkast okay, men i fremtiden vil han gerne have en mere strømlinet og overskuelig side, som også inkluderer Your Locals farveskema.
2. GDPR-samtykket var en god tilføjelse, men han vil have ændret det til 'vilkår og betingelser', som linker til en pop-up med en Lorem Ipsum.
3. I forlængelse af punkt 1 vil kunden gerne have en navigationsbar i toppen, så man kan navigere rundt når man er logget ind. Kunden vil også gerne have en funktion i navigationsbaren, hvor man kan gå ind og redigere sine oplysninger, oppe i højre hjørne ved siden af en 'log ud' knap.
  - a. Ud fra dette krav oprettede vi en ny User Story, da vi blev enige om, at kravet ikke hørte til under nogen eksisterende User Story, og i øvrigt også var for stor til at være en enkeltstående task.
4. Kunden vil gerne have mulighed for at kunne oprette sig som kunde eller firma fra login-siden.

Vi har altså fået tre nye User Stories:

USER STORIES
Som bruger skal jeg have en tilpasset forside med navigeringsbar afhængig af min brugertype, med en kontoredigeringsfunktion, så jeg nemt kan navigere rundt på siden.
Som kunde og firma skal jeg kunne godkende Your Locals vilkår og betingelser, så mine personlige data kan blive behandlet og så jeg kan oprette en bruger.
Som ikkeeksisterende bruger skal jeg kunne oprette en bruger fra login-siden.

---

## DISKUSSION OG PROJEKTERING

Vi har, efter tilføjelsen af de nye user stories, nu 10 i vores product backlog.

Inden vi har estimeret de nye user stories, vurderer product owner at vi vil kunne levere alle resterende user stories i vores product backlog til næste sprint, med undtagelse af user story 7.

US07 er stadig en feature som vi er i tvivl om hvordan skal gribes an, da det kræver implementering af en geolokation. Derfor ville det give bedst mening at fokusere på at få de mest væsentlige funktionaliteter på plads i næste sprint.

Vi forventer til næste sprint en klar stigning af feedback fra kunden, i og med at vi til næste inkrementering vil indeholde så mange nye funktionaliteter.

## SPRINT RETROSPEKT

Af Alle

### POSITIVT

1. God kommunikation mellem udviklerne: vi har løbende alle haft fingeren på pulsen ift. de andre udviklere, og hvad de arbejder med. Der er efter hvert færdigudviklet artefakt blevet lavet informal reviews fra de resterende udviklere, hvilket har affødt løbende feedback og fælles forståelse for processen. Der har desuden også været en rigtig konstruktiv dialog, hvor parterne kom med deres udlæg, fulgt af en rationel diskussion, og ultimativt en fælles enighed om den videre fremgang. Følgevirkningen af dette har været, at alle udviklerne har haft lyst til at spørge hinanden til råds og sparre omkring mulige løsninger og opsætninger.
2. En struktureret arbejdsfremgang: Vi har under sprinten været rigtig gode til at tilpasse os den arbejdsdag haft i løbet af sprinten, qua vores prædefinerede rollefordeling. Alle i teamet har haft et klart ansvarsområde, og der har derfor ikke været nogen slack i forhold til at videreudvikle artefakter under processen.  
Desuden har vi jf. vores risikoplan haft en klar fremgangsmåde, hvis der skulle opstå problemer - hvis der f.eks. ikke har været et ledigt lokale på skolen, har vi haft en backup location vi kunne sætte os og fortsætte arbejdet.
3. God morale: til trods for en igangværende sprint, et sideløbende projekt og diverse fremlæggelser og undervisning på samme tid, har teamet lagt en markant indsats i at nå vores deadlines. Vi har alle medvirket til at løfte i flok, støttet hinanden, og haft en gensidig entusiasme for at nå vores mål. Vi er som team enige om, at vi klart arbejder bedst når vi er samlet, kontra over skype el. lign.

### NEGATIVT

1. Vi havde en del opsætningsproblemer, som besværliggjorde vores udvikling. Det var først efter leveringen af inkrementeringen, at vi fik løst vores opsætningsproblemer. Vi havde blandt andet problemer med:
  - Git-styringen: når vi pushede til GitHub havde vi to forskellige mapper med næsten identiske navne, som hver især indeholdt halvdelen vores program. Dette besværliggjorde vores arbejdsdag, da vi måtte ind i de forskellige mapper og lede efter de filer vi skulle bruge. Vi diskuterede at lave rollback til en tidligere version, men blev enige om at der ville forekomme for meget tab af data. Vi valgte derfor at klon repositoryet lokalt, og opsætte et nyt repository der indeholdt de gamle filer samlet i en enkelt mappe. Dette var en nødvendighed da vi først fandt fejlen så sent i projektet.
  - IDE-opsætningen grundet fejlagtige paths til JRE og SDK. Vi havde dette problem de første to dage af sprinten, men fik løst det efter ca. 2 timers research.
2. Eftersom vi igangsatte et projekt sideløbende, som blev frigivet for sent, måtte vi tilpasse os efter omstændighederne. Ideelt skulle vi arbejde på projektet efter vores kontrakts retningslinjer - mellem 8 - 14 hver dag - men grundet det andet projekts deadline måtte vi opdele vores tid. Derfor måtte vi arbejde på det andet projekt i dagstimerne på skolen, og tage de nødvendige timer til sprinten efter skoletid, for at kunne overholde vores estimeringer, hvilket resulterede i nogle meget lange arbejdsdage.



3. Vi havde ikke den opfattelse af, at stand-up meetings gav nogen synderlig værdi i vores tilfælde, da vi er så lille en gruppe som arbejder tæt op ad hinanden, og rådfører os med hinanden. Ydermere har vi kun et fåtal af gangene været fuldtallige til stand-up meetings pga. sygdom eller arbejde, så vi følte ikke at vi fik den fulde værdi ud af lige netop dette aspekt af Scrum.
4. Baseline har været besværlig at overholde fuldstændigt, eftersom gruppen er lille og der løbende er blevet holdt mange informal reviews. Dette har resulteret i nogle frustrationer blandt gruppen, da der har været en mangel på forståelse af logikken bag versionerne. Der er desuden ikke blevet holdt ordentligt hånd i hanke med versionstyringen, hvilket har skabt unødigt forvirring. Manglende definition på den enkelte artefakt til baselinen, har desuden også haft indflydelse på forvirringen.

---

## ÆNDRINGER

1. I og med vi har afsluttet vores sideløbende projekt, vil vi kunne opstille et mere struktureret arbejdsmiljø. Vi vil i øvrigt digitalisere vores SCRUM-board, så vi hele tiden har hånd i hanke med hvilke tasks der er i vores sprint backlog, og i de øvrige faser.
2. Vi fastsætter vores arbejdslokaler for hele sprinten, så der ikke opstår uklarhed omkring mødested og -tidspunkt. Vi vil være på skolen tirsdag og fredag, for at få så meget ud af vejledningen som muligt. Onsdag og fredags mødes vi privat, og torsdag fortsætter vi arbejdet på skolen efter endt undervisning.
3. Vi har klargjort, at vi vil benytte pair-programming afhængig af tasks, i stedet for et låst par under hele sprinten. Vi mener at dette er mere hensigtsmæssigt, da vi derfor kan benytte hinandens kompetencer bedre til de specifikke opgaver.

## SPRINT 2

*Af Casper og Christian*

I sidste sprint fik vi implementeret begge vores user stories, som er to vitale funktioner: at kunne oprette sig som firma og som kunde, hvilket er præmissen for at kunne benytte sig af Your Locals tjenester. Vi har nu fået dannet grundlag for at kunne videreudvikle de tilhørende funktionaliteter, der er skal implementeres i dette sprint.

## SPRINT PLANNING

*Af Alle*

### RÅDIGHEDSTIMER

Under denne sprint, har holdet sat følgende timer til rådighed:

NAVN	DAG 1	DAG 2	DAG 3	SAMLET
Matthias	7	5	5	17
Casper	5	5	5	15
Christian	7	5	5	17
Patrick	4	4,5	5	13,5
I ALT	23	19,5	20	62,5

## PRODUCT BACKLOG

Vi har efter sidste sprint re-estimeret vores user stories og tasks, som vises i user story tabellerne - dels fordi vi har fået tilføjet tre nye user stories, og dels fordi vi er blevet en del klogere på vores tempo. Herunder er vores product backlog med de nye estimeringer:

User story ID	User stories	Estimerede timer
US03	Som firma skal jeg kunne oprette nye tilbud, som skal vises til kunder, så kunderne har mulighed for at købe vores tilbud.	5,5
US04	Som ansat hos Your Local skal jeg have administrative rettigheder over firmaer og kunder, så jeg kan godkende firmaer og redigere og slette brugerdata ift. GDPR.	8
US05	Som kunde skal jeg kunne købe et tilbud, så jeg er sikker på at mit køb er blevet godkendt.	8,25
US06	Som firma skal jeg kunne se en oversigt over hvor mange tilbud der er blevet købt, så jeg kan holde statistik over mine salg.	2,25
US07	Som kunde skal jeg kunne se en oversigt over tilbud i mit nærområde, og sortere i disse ud fra kategorier, så jeg kun ser relevante tilbud ift. min placering.	11,75
US08	som bruger skal jeg have en tilpasset forside med navigeringsbar afhængig af min brugertype, med en kontoredigeringsfunktion, så jeg nemt kan navigere rundt på siden.	5,75
US09	som kunde og firma skal jeg kunne godkende Your Locals vilkår og betingelser, så mine personlige data kan blive behandlet og så jeg kan oprette en bruger.	1
US10	som ikke-eksisterende bruger skal jeg kunne oprette en bruger fra login-siden, så jeg kan benytte mig af Your Locals tjenester som kunde eller firma.	1

---

## SPRINT GOAL

I følgende sprint vil vi derfor kunne implementere følgende funktioner:

- Vi giver virksomhederne mulighed for at oprette tilbud, såfremt de har oprettet sig som bruger. Medarbejdere under virksomheden vil fremover kunne logge ind på Your Local, og oprette tilbud på varer der er ved at nå sidste udløbsdato.
- Medarbejdere hos TheNext vil have administrative rettigheder over både kunder og virksomheder, hvilket indebærer at kunne redigere kontaktoplysninger for begge – dog ikke betalingsinformationer. Derudover vil administratoren have mulighed for at slette en kunde eller bruger. Der vil også blive implementeret en funktionalitet som, når en virksomhed oprettes, giver administratoren mulighed for enten at godkende eller afvise denne, af sikkerhedsmæssige årsager.
- Kunder der logger ind hos Your Local vil have mulighed for at se en oversigt over udbudte tilbud fra butikker. Eftersom man som kunde allerede har angivet sine betalingsoplysninger, vil man let kunne vælge et tilbud og antal, og dertil få en kvittering for sit køb.
- Virksomhederne vil kunne se en oversigt over alle deres udbudte tilbud, nuværende og ældre, så der kan blive holdt statistik over salgene. Dette vil især være hensigtsmæssigt ift. prissætning, og om der er generel udløbstendens blandt butikkens fødevarer.
- Alle brugere vil kunne navigere mellem deres tilgængelige valgmuligheder via en navigationsbar, hvilket gør brugeroplevelsen for brugerne mere intuitivt og fuldendt.
- Produktet vil gøre brugeren opmærksom på, at vedkommendes personlige data vil blive behandlet i overensstemmelse med den nuværende lovgivning, jf. den nye persondataforordning.
- Man vil nu nemt kunne oprette sig som bruger fra login-siden via knapper, hvor brugeren vælger at oprette sig som virksomhed eller kunde.

## SPRINT BACKLOG

Af Alle

Vi har efter at have konsulteret product owner fået aftalt hvilke user stories vi skal have med i vores sprint backlog. De står herunder i prioriteret rækkefølge:

### US03

ID: US03	User Story: Som firma skal jeg kunne oprette nye tilbud, som skal vises til kunder, så kunderne har mulighed for at købe vores tilbud.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)	Planning poker 2.0	Three point value 2.0	Estimeret produkt 2.0
US03.1	Underside til oprettelse af tilbud, med tilknytning af billede.	2	4	3	4	3	3,5
US03.2	Redigerings- og slettefunktion af eksisterende tilbud	4	4	4	1	1	1
US03.3	Underside med oversigt over aktive tilbud	X	X	X	1	1	1
<b>SAMLET:</b>	Det samlede timeantal for estimeringerne	6	8	7	6	5	5,5

## US04

ID: US04	User Story: som ansat hos Your Local skal jeg have administrative rettigheder over firmaer og kunder, så jeg kan godkende firmaer og redigere og slette brugerdata ift. GDPR.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)	Planning poker 2.0	Three point value 2.0	Estimeret produkt 2.0
US04.1	Godkendelsesfunktion til firmaoprettelse.	3	8	5,5	1	1,5	1,25
US04.2	Redigering af information for både kunder og firmaer, med tilhørende undersider der henter brugere.	3	4	3,5	4	3,5	3,75
US04.3	Slettefunktion der fjerner en kunde eller et firma.	3	1	2	1	1	1
US04.4	Underside til godkendelse af nyoprettede firmaer.	X	X	X	2	2	2
<b>SAMLET:</b>	Det samlede timeantal for estimeringerne	9	13	11	8	8	8

## US05

ID: US05	User Story: som kunde skal jeg kunne købe et tilbud og få vist en kvittering, så jeg er sikker på at mit køb er godkendt.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)	Planning poker 2.0	Three point value 2.0	Estimeret produkt 2.0
US05.1	Funktion der ændrer lagerantallet for det pågældende tilbud, med tilhørende restriktioner.	1	1	1	2	2	2
US05.2	Funktion der kan bekræfte køb og vise en Lorem Ipsum kvittering	2	2	2	2	1,5	1,75
US05.3	Underside for køb af tilbud.	2	2	2	2	2,5	2,25
US05.4	Underside der viser købte tilbud, med antal.	X	X	X	2	2,5	2,25
<b>SAMLET:</b>	Det samlede timeantal for estimeringerne	5	5	5	8	8,5	8,25

## US06

ID: US06	User Story: som firma skal jeg kunne se en oversigt over hvor mange tilbud der er blevet købt, så jeg kan holde statistik over mine salg.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)	Planning poker 2.0	Three point value 2.0	Estimeret produkt 2.0
US06.1	Funktion der henter salgsinformation fra databasen, med tilhørende underside.	1	1	1	2	2,5	2,25
SAMLET:	Det samlede timeantal for estimeringerne	1	1	1	2	2,5	2,25



## US07

ID: US07	User Story: som kunde skal jeg kunne se en oversigt over tilbud i mit nærområde, og sortere i disse ud fra kategorier, så jeg kun ser relevante tilbud ift. min placering.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)	Planning poker 2.0	Three point value 2.0	Estimeret produkt 2.0
US07.1	Som kunde skal jeg kunne finde tilbud indenfor en geolokation på 5 km.	7	8	7,5	8	12	10
US07.2	Underside til at søge på tilbud på postnummer.	6	1	3,5	1	2,5	1,75
SAMLET:	Det samlede timeantal for estimeringerne	13	9	11	9	14,5	11,75

## US08

ID: US08	User Story: som bruger skal jeg have en tilpasset forside med navigeringsbar afhængig af min brugertype, med en kontoredigeringsfunktion, så jeg nemt kan navigere rundt på siden.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)	Planning poker 2.0	Three point value 2.0	Estimeret produkt 2.0
US08.1	Opret brugerspecifikke navigeringsbarer på alle undersider tilknyttet brugeren.	X	X	X	2	2,5	2,25
US08.2	Opret logo og tilknyt navigeringsbaren.	X	X	X	1	1	1
US08.3	Opret brugerspecifikke forsider.	X	X	X	2	2,5	2,25
<b>SAMLET :</b>	Det samlede timeantal for estimeringerne	X	X	X	5	6,5	5,75

## US09

ID: US09	User Story: som kunde og firma skal jeg kunne godkende Your Locals vilkår og betingelser, så mine personlige data kan blive behandlet og så jeg kan oprette en bruger.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)	Planning poker 2.0	Three point value 2.0	Estimeret produkt 2.0
US09.1	Opret modal for vilkår og betingelser der indeholder Lorem Ipsum	X	X	X	1	1	1
<b>SAMLET:</b>	Det samlede timeantal for estimeringerne	X	X	X	1	1	1

## US10

ID: US10	User Story: som ikke-eksisterende bruger skal jeg kunne oprette en bruger fra login-siden.	Three point value (timer)	Planning poker (timer)	Estimeret produkt ((TPV+PP)/2)	Planning poker 2.0	Three point value 2.0	Estimeret produkt 2.0
US10.1	Opret to knapper der redirecter til hhv. opret kunde og virksomhed.	X	X	X	1	1	1
SAMLET:	Det samlede timeantal for estimeringerne	X	X	X	1	1	1

## TASK DESCRIPTIONS

Af Matthias

### US03

ID	TASK	BESKRIVELSE
<b>US03.1</b>	Underside til oprettelse af tilbud, med tilknytning af billede.	<p>Der skal laves en underside til oprettelse af tilbud. Denne underside skal indeholde følgende input felter: navn på vare, pris, start dato, slut dato, kvantitet og et felt hvor man kan indsætte URL til et billede. Input-felterne skal overholde specifikke patterns, så det ikke er muligt at indtaste ugyldigt input – eksempelvis skal kvantitet ikke kunne være negativt. Derudover skal session for firma overføres, så det vides hvilket firma tilbuddet skal oprettes for.</p> <p>Når brugeren trykker gem, skal informationerne gemmes korrekt i databasen, såfremt de er gyldige. Hvis de er ugyldige, skal det i inputfelterne angives, hvordan input skal skrives før det er gyldigt.</p> <p>Derudover skal der være en 'Reset' knap, som fjerner alt indtastet information og en 'Cancel' knap der redirecter brugeren tilbage.</p>
<b>US03.2</b>	Redigerings- og slettefunktion af eksisterende tilbud	<p>Der skal laves en underside, hvor man kan redigere et enkelt eksisterende, aktivt tilbud. Der skal være samme input-felter som i US03.1 og samme anvendte patterns og titles. Når en bruger redirectes til denne side, skal alle de eksisterende informationer om tilbuddet, allerede være udfyldt.</p> <p>Når brugeren trykker gem, skal informationerne gemmes korrekt i databasen, såfremt de er gyldige.</p> <p>Dertil skal der ligeledes være en 'Default' knap, som genindsætter informationerne, som tilbuddet oprindeligt indeholdte. Der skal også tilføjes en 'Cancel' knap, som redirecter brugeren tilbage.</p> <p>På oversigten fra US03.3, skal der implementeres en 'Delete' knap(slettefunktion), som fjerner et tilbud. Når et tilbud 'slettes', bliver det sat som inaktivt i databasen, så det stadig er muligt at hente salgsstatistikker på tilbuddet.</p>
<b>US03.3</b>	Underside med oversigt over aktive tilbud	<p>Der skal implementeres en underside til applikationen, hvor et firma kan se en oversigt over deres aktive tilbud (et tilbud er inaktivt hvis det er slettet, eller kvantiteten er 0). Fra denne side skal 'Edit' funktionen (se US03.2) vises ud fra hvert tilbud – samme gør sig gældende for 'Delete'-funktionen(se US03.2). Desuden skal der fremgå en 'Opret tilbud' knap på siden, som omdirigerer brugeren til undersiden for at oprette tilbud.</p>

## US04

Af Matthias

ID	BESKRIVELSE
<b>US04.1</b>	<p>Der skal implementeres en godkendelses-funktion, således at når et firma har oprettet sig, og befinder sig på ventelisten (jf. US04.4), har en administrativ bruger mulighed for at godkende firmaet. Når brugeren trykker på 'Godkend', skal firmaets status i databasen ændres fra inaktiv til at være aktiv.</p> <p>Derudover skal der også være en knap 'Afvis', i tilfælde af at firmaet ikke skal godkendes af administratoren. Denne funktion sletter firmaet helt fra databasen.</p>
<b>US04.2</b>	<p>I denne task, skal der implementeres to forskellige undersider der fungerer som oversigter: en over eksisterende kunder og en over aktive firmaer. Ud fra hver bruger (både kunde og firma) på disse undersider, skal der være en 'Edit' knap. Denne edit knap skal redirecte til hhv. US02.6 for kunde og US01.6 for firma. Disse undersider skal fungere som beskrevet i de nævnte tasks, således at eksisterende information allerede er udfyldt og patterns er overholdt.</p>
<b>US04.3</b>	<p>På den nævnte underside i US04.2, skal der også eksistere en slettefunktion, der præsenteres i form af en 'Delete' knap ud fra hver bruger, på begge undersider. Når en administrativ bruger trykker på delete-knappen ud fra et firma eller en kunde, skal brugeren og dertilhørende data slettes fra databasen.</p>
<b>US04.4</b>	<p>Der skal implementeres en underside til godkendelse af nyoprettede firmaer. Dette fungerer således, at en administrativ bruger har en oversigt på undersiden, over hvilke firmaer der venter på at blive enten godkendt eller afvist. De firmaer der vises på denne oversigt, skal være firmaer der er sat som inaktive i databasen.</p>

## US05

Af Matthias

ID	BESKRIVELSE
<b>US05.1</b>	Der skal implementeres en funktion, der arbejder sammen med US05.2, således at når et køb bliver bekræftet, skal kvantiteten for tilbuddet i databasen, dekrementeres med antal købte. Derudover skal et tilbud ændres til værende inaktiv når lagerstatus rammer nul. Det skal heller ikke være muligt at købe flere antal, hvis dette betyder at lagerbeholdningen går i minus.
<b>US05.2</b>	Når en kunde har udset sig til et tilbud og befinder sig på undersiden for det specifikke tilbud (jf. US05.3), skal der være en 'Buy' knap. Når kunden trykker på denne knap, skal informationer om kvantitet (hvor mange kunden ønsker at købe) overføres til databasen og US05.1 udføres. Købsfunktionen sørger også for at lagre informationer om salget i databasen, så firmaer senere hen kan se statistik over disse salg.
<b>US05.3</b>	Der skal oprettes en underside for køb af tilbud til kunder. Denne underside kommer i spil, når en kunde befinder sig på kunde-forsiden (jf. US08.3) og kunden trykker på 'Køb' ud fra et specifikt tilbud. Derefter bliver de redirected med deres brugersession til den omtalte underside i denne task, hvor de kan vælge hvilken kvantitet af tilbuddet de ønsker at købe. Der skal være pattern på denne kvantitetsfunktion, så det ikke er muligt at købe en negativ kvantitet. På denne underside skal der vises følgende informationer på tilbuddet: Navn, pris, lagerantal og billede. Knappen fra task US05.1 skal vises på denne underside.
<b>US05.4</b>	<p>I denne task skal der implementeres en underside som bruger-typen kunde har adgang til. Denne underside skal vise en købsoversigt, over hvilke tilbud en kunde har købt. Alle informationer der eksisterer på tilbuddet skal hentes fra databasen og vises i en tabel-oversigt på denne underside.</p> <p>Der skal også være et tilknyttet id til hver ordre, så en kunde har mulighed for at vise dette til firmaet når personen fysisk afhenter sit købte produkt.</p>

---

## US06

Af Matthias

ID	BESKRIVELSE
US06.1	<p>I denne task skal der oprettes en underside, som er tilgængelig når man er logget ind som et firma. Denne oversigt skal vise statistik over, hvor mange tilbud firmaet har solgt og hvad deres samlede indtægt har været for disse tilbud.</p> <p>Information skal læses korrekt fra databasen, og der skal beregnes et felt til pris, hvor mængden af tilbud solgt ganges med pris pr. tilbud.</p>

---

## US07

ID	BESKRIVELSE
US07.1	<p>Der skal i navbaren for en kunde, være et punkt 'Nearby offers' som redirecter til den underside som skal oprettes i denne task.</p> <p>Her vil der blive vist alle tilbud indenfor 5 km. radius, baseret på kundens postnummer og adresse, som hentes fra databasen. Måden hvorpå tilbud indenfor 5 km. radius skal hentes, er ved at lave API-kald til 'Geolocation'.</p>
US07.2	<p>Hvis kunden befinder sig på undersiden for 'Nearby offers' (se US07.1) og ønsker at skifte til et andet postnummer, skal der være et input-felt i toppen, så kan man indtaste et nyt postnummer og se hvilke tilbud der er inden for 5 km. radius af det indtastede postnummer. Dette skal sørge for at de oprindelige tilbud som var tilknyttet kundens rigtige postnummer og adresse, skal fjernes og i stedet vises alle tilbud for det nye postnummer.</p>

ID	TASK	BESKRIVELSE
US08.1	Opret brugerspecifikke navigeringsbarer på alle undersider tilknyttet brugeren.	<p>Der skal implementeres 3 forskellige slags navigationsbarer på applikationen, som skal indeholde følgende:</p> <ul style="list-style-type: none"> <li>• <b>Kunde</b> <ul style="list-style-type: none"> <li>○ Logo (jf. US8.2)</li> <li>○ Tilbud (Aktive tilbud, købshistorik, jf. US5.4)</li> <li>○ Kontoprofil (redigering) (jf. US2.6)</li> </ul> </li> <li>• <b>Firma</b> <ul style="list-style-type: none"> <li>○ Logo (jf. US8.2)</li> <li>○ Tilbud (Oversigt, jf. US03.3)</li> <li>○ Statistik (jf. 6.1)</li> <li>○ Kontoprofil (Redigering, jf. 1.6)</li> </ul> </li> <li>• <b>Administrator</b> <ul style="list-style-type: none"> <li>○ Logo (jf. US8.2)</li> <li>○ Kunder (jf. US4.2)</li> <li>○ Firmaer (jf. US4.2)</li> <li>○ Venteliste (jf. US4.4)</li> </ul> </li> </ul> <p>Hver navigationsbar, skal specifikt tilpasses ud fra hvilken login-type der er aktiv på siden. Navigationsbaren skal følge med på alle undersider, så man altid kan navigere.</p>
US08.2	Opret logo og tilknyt navigeringsbaren.	På kundens opfordring skal der laves et 'TheNext' logo, for at gøre applikationen mere fuldstændig. Ingen layout-krav, der skal dog laves en redirect til forsiden når man trykker på logoet afhængigt af, hvilken brugertype der er logget ind – se US08.3 for de forskellige forsider.
US08.3	Opret brugerspecifikke forsider.	<p>Der skal oprettes 3 forskellige forsider, som man skal lande på efter man har logget ind, afhængigt af hvilken bruger-type man er:</p> <ul style="list-style-type: none"> <li>• <b>Kunde</b> <ul style="list-style-type: none"> <li>○ Forsiden indeholder oversigten over alle udbudte tilbud. Denne side skal hente alle aktive tilbud fra databasen, sortere dem efter dato og præsentere dem med billede til venstre, samt information til højre og en 'Buy'-knap, der redirecter til underside for køb af tilbud (se US05.3).</li> </ul> </li> <li>• <b>Firma</b> <ul style="list-style-type: none"> <li>○ Forsiden for et firma er opstillet således, at der er 3 ikoner med en CTA-knap til hver: 'Opret tilbud', 'Se statistik', 'Se mine tilbud'</li> </ul> </li> <li>• <b>Administrator</b> <ul style="list-style-type: none"> <li>○ Forsiden når en administrator logger ind, skal landingssiden være ventelisten over firmaer som skal godkendes eller afvises (se US04.4).</li> </ul> </li> </ul>



---

**US09**

ID	BESKRIVELSE
<b>US09.1</b>	<p>Når en kunde er i gang med at oprette sig som bruger på hjemmesiden, og udfylder de nødvendige informationer(jf. US02.3), skal der være et link med teksten 'I accept the terms and conditions', som linker til en modal med en auto-genereret Lorem-Ipsum som senere hen kan udfyldes af TheNext. Denne modal skal have et 'x' i højre hjørne, som lukker modalen.</p> <p>Der skal ikke eksistere yderligere funktionalitet i modalen, da der allerede er en checkboks ved siden af 'terms and conditions' linket.</p>

---

**US10**

ID	BESKRIVELSE
<b>US10.1</b>	<p>Når man befinder sig på /login siden, skal der under login-formularen være to links som redirecter. Det ene 'Opret kunde' skal redirecte til /create/customer, så en kunde har mulighed for at oprette sig på applikationen.</p> <p>Det andet link 'Opret firma' skal redirecte til /create/company, så et firma har mulighed for at oprette sig på applikationen.</p>

## END-TESTS

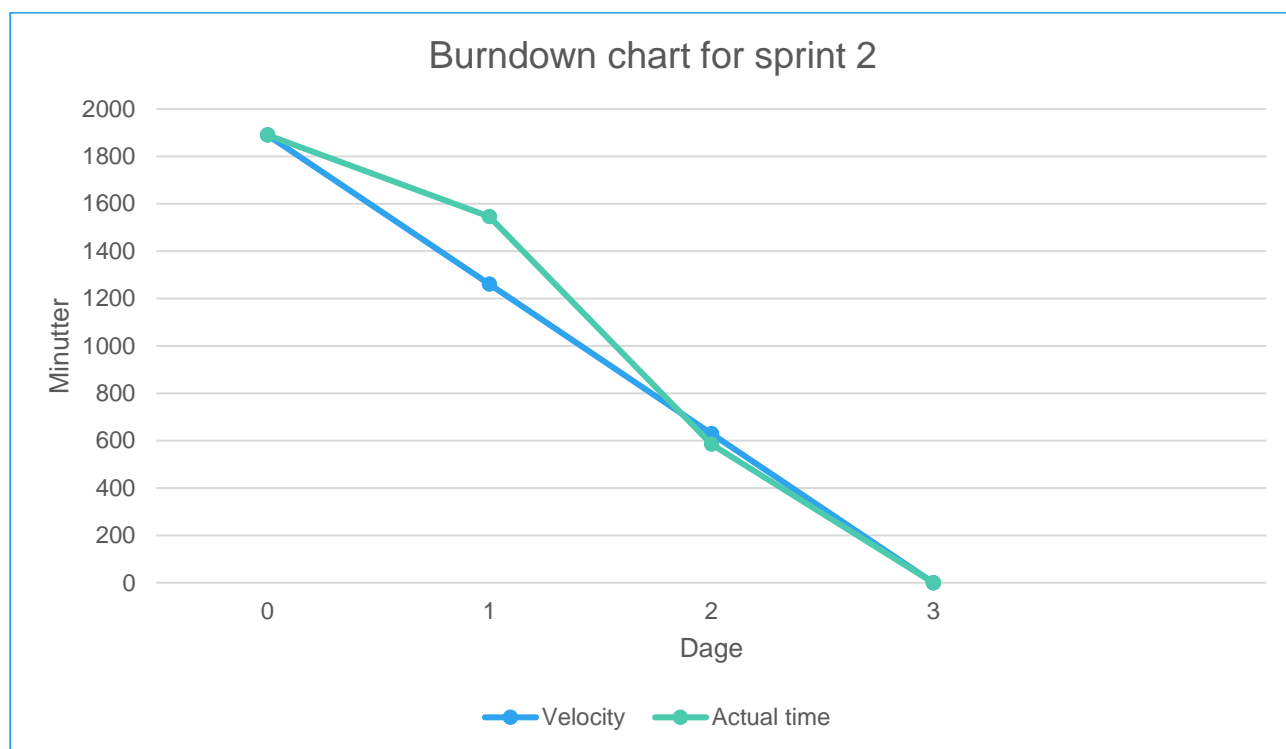
Af Matthias

ID	HVIS	NÅR	SÅ
<b>US05.1.1</b>	En kunde har fundet et tilbud og trykker 'Buy'	Kvantiteten overskrider resterende lagerbeholdning	Vises fejlmeddelelse om at den ønskede kvantitet er for høj.
<b>US05.1.2</b>	En kunde har fundet et tilbud og indtaster ønsket kvantitet	Kvantiteten er negativ	Vil der opstå en fejl, da input-feltet er låst til at være minimum 1.
<b>US05.1.3</b>	En kunde har fundet et tilbud og trykker 'Buy'	Kvantiteten stemmer overens med resterende lagerbeholdning, så det går i nul	Gennemføres købet og tilbuddet sættes som 'inaktiv' i databasen.
<b>US05.1.4</b>	En kunde har fundet et tilbud og trykker 'Buy'	Et gyldigt antal kvantitet er valgt	Formindskes lagerbeholdningen i databasen og købet gennemføres.
<b>US04.1.1</b>	En administrativ bruger befinder sig på ventelisten over nyligt oprettede firmaer	Administratoren trykker 'Accept' ud fra et firma	Ændres firmaets status fra inaktiv til aktiv i databasen, og firmaet har mulighed for logge ind og benytte sig af applikationen.
<b>US04.1.2</b>	En administrativ bruger befinder sig på ventelisten over nyligt oprettede firmaer	Administratoren trykker 'Decline' ud fra et firma	Bliver firmaet slettet fra databasen og fjernet fra ventelisten.
<b>US03.2.1</b>	Et firma er i gang med at redigere et eksisterende tilbud	Brugeren trykker på 'Reset' knappen	Genindsættes de informationer, som tilbuddet allerede indeholdte.
<b>US03.2.2</b>	Et firma er i gang med at redigere et eksisterende tilbud	Brugeren trykker på 'Cancel' knappen	Brugeren redirectes til /offer oversigten
<b>US03.2.3</b>	Et firma er i gang med at redigere et eksisterende tilbud	Brugeren trykker på 'Save' knappen	Brugeren redirectes til /offer oversigten og tilbuddet bliver gemt korrekt i databasen.
<b>US03.2.4</b>	Et firma er i gang med at redigere et eksisterende tilbud	Brugeren trykker på 'Delete' knappen	Slettes tilbuddet fra oversigten og fra databasen.

<b>US08.3.1</b>	En bruger logger ind på applikationen	Bruger-typen er 'Firma'	Brugeren henvises til den korrekte forside for et firma.
<b>US08.3.2</b>	En bruger logger ind på applikationen	Bruger-typen er 'Kunde'	Brugeren henvises til den korrekte forside for en kunde.
<b>US08.3.3</b>	En bruger logger ind på applikationen	Bruger-typen er 'Administrator'	Brugeren henvises til den korrekte forside for en administrator.

## BURNDOWN CHART

Af Casper



7: burn down chart for sprint 2.

Vores andet burn down chart ser væsentligt mere anderledes ud end vores første. Vi har ud fra vores rådighedstimer vurderet, at vi kan nå vores sprint goal på tre dage, da vi ikke har andre sideløbende projekter.

Vi blev desværre mødt af nogle komplikationer den første dag, hvilket gjorde at vi måtte relokalisere til en anden arbejdsplads. Dette gik der en del spildtid med, hvilket vores actual graf også afspejler - vi nåede ikke vores velocity første dag.

For at nå vores mål måtte vi derfor indhente de ekstra timer næste dag, hvilket heldigvis kunne lade sig gøre. Derfor nåede vi altså de resterende tasks på sprintens sidste dag.

## SPRINT REVIEW

*Af Casper og Christian*

Sprint nr. 2 er nu blevet afsluttet. I det følgende review og retrospekt vil vi tage udgangspunkt i, at denne sprint ikke vil være vores sidste, men eftersom projektets deadline ikke giver tid til flere sprints, vil vi efterfølgende ikke inkludere nye krav og estimeringer i rapporten.

---

### INKREMENTERINGEN

Denne inkrementering indeholder væsentligt flere implementerede funktioner end den forrige sprint. Mange af de tilføjelser der er kommet, er bygget op omkring de basale funktionaliteter som sidste sprint lagde grobund til.

Vi nåede samtlige user stories i vores sprint backlog, og applikationen kan nu følgende:

- Et firma kan nu:
  - Oprette tilbud når de er logget ind. Der kan angives navn, mængde, pris og der kan tilknyttes et URL-billede af den vare som bliver tilbudt.
  - Se en liste over deres aktive tilbud, hvor mange der er tilbage, start/slutdato, hvad stk. prisen er, og URL'en tilknyttet til billedet.
  - En statistik over alle aktive og inaktive tilbud, mængden af varen, hvor mange der er blevet købt, stk. pris og den samlede omsætning.
- En kunde kan nu:
  - Se en liste over alle tilbud der er aktive hos alle firmaer, med tilknyttet pris, en lorem ipsum beskrivelse, og hvor mange der er tilbage af de pågældende tilbud.
  - Vælge et tilbud, hvor mange af dette kunden vil købe, og få en bekræftelse af købet som en kvittering.
  - En oversigt over de tilbud som kunden har købt, inklusiv et salgs-ID, navn, antal købt og pris pr. styk.
- En administrator kan nu:
  - Godkende nyoprettede firmaer. Et firma vil ikke kunne logge ind før det er blevet verificeret af en administrator.
  - Se en liste over både kunder og firmaer, hvor begge brugertyper kan redigeres.
  - Administratoren kan desuden slette en bruger fuldstændig fra databasen.
- Som ikkeeksisterende bruger kan du nu oprette dig fra login-siden.
- Alle brugertyper har nu en navigationsbar i toppen, der henviser til den enkelte brugertypes relevante undersider.
- Jf. sidste review er der nu implementeret en knap der viser et lorem ipsum over tjenestens vilkår og betingelser, som brugeren skal godkende under oprettelsen.

---

## DEMONSTRATION & FEEDBACK

Efter demonstrationen af de implementerede funktioner havde kunden følgende feedback:

1. Kunden er meget positiv over layout indtil videre – han havde dog følgende underpunkter:
  - a. Han vil gerne have farver på siderne. Specielt den grønne farve, som Your Local benytter sig af, vil han gerne have på navigationsbaren.
  - b. Noget 'fylde' i navigationsbaren – eventuelt en titel på siden oppe i headeren.
  - c. Købte tilbud som firma kunne godt have en visuel model af udbudte tilbud, og hvor mange der er blevet, eller blev solgt.
  - d. En footer til kunde og firma indeholdende kontakthinformation, copyright, driftsstatus, link til vilkår & betingelser ect.
2. Det ville være nærliggende at have en side, hvor man kunne se sine kontakthinformationer som kunde og firma, uden at skulle redigere sine oplysninger.
3. Kunden vil gerne have, at man i fremtiden skal kunne oprette butikker under det firma man er logget ind som. Dette vil i fremtiden gøre det muligt at have en administrativ firmabrunder, der kan give overblik over tilknyttede butikker, med henblik på salgsstatistik og andre målepunkter.
4. En bruger skal fremadrettet kunne gå ind og slette sig selv fra tjenesten. Vi har i dialog aftalt, at når en kunde eller et firma sletter sig selv, bliver de blot sat som inaktive i databasen, i forhold til fremtidig markedsføring. Dog skal en administrator kunne fjerne en bruger helt fra databasen hvis brugeren ønsker dette, jf. persondataforordningen GDPR
5. Kunden vil gerne kunne målrette de tilbud der bliver udbudt til kunderne. Derfor vil han gerne have, at en kunde der logger ind, skal have vist de tilbud på forsiden, som denne er interesseret i.
6. Derudover vil kunden gerne have en søgefunktion der kan søge på kategorier, dato eller pris.
7. Kunden bed mærke i, at et tilbud der blev oprettet havde et URL som billede. I fremtiden kunne han godt tænke sig, at dette var en fil der kunne uploades fra en server eller lokalt.
8. Engang ude i fremtiden kunne kunden også godt tænke sig en implementeret supportchat, hvor en ikke-administrativ bruger kan forbinde til en chat, hvor en supportmedarbejder kan assistere brugeren.
9. User Story 7 blev diskuteret. I begyndelsen af projektet var kundekravet at der skal kunne gives besked til kunder om, hvilke tilbud der var i nærheden af dem, vha. en geolokation. Præmissen for denne funktion er, at der skal tilkobles en geolokation for kunden, og dette vurderer vi som udviklere ikke at have kompetencer til på nuværende tidspunkt. User story 7 vil derfor blive omformuleret til, at man som kunde kan søge på postnumre eller landområder, og blive lagt sammen med den user story der efterspørger søgeparametre.

Vi har efter kundens feedback fået tilføjet følgende user stories til product backloggen:

USER STORIES
Som stakeholder vil jeg gerne have tilføjet layout nogle layoutpunkter, der gør siden mere komplet, for at give brugeren en bedre oplevelse af applikationen.
Som firma skal jeg kunne oprette butikker under firmaets konto, med særskilt login, så hver butik kan logge ind og oprette tilbud, der er specifikt tilknyttet til sin egen butik.
Som ikke-administrativ bruger skal jeg kunne gå ind og slette (deaktivere) min konto fra Your Locals tjeneste.
Som kunde skal jeg kunne se en oversigt over tilbud der er relevante for mine interesser, så jeg kun får vist det jeg har tilvalgt som kategorier.
Som kunde vil jeg gerne kunne søge efter tilbud på specifikke parametre, så jeg kan indrette mine køb efter mine behov.
Som stakeholder vil jeg gerne have lavet en funktion, der giver firmaer mulighed for at uploade et billede som fil i stedet for en URL, så det i fremtiden vil gøre det lettere at tilknytte relevante billeder til tilbuddet.
Som stakeholder vil jeg gerne have implementeret en supportchat for både kunder og firmaer, så begge brugertyper i fremtiden har mulighed for at få teknisk support direkte via siden.

---

## DISKUSSION OG PROJEKTERING

Denne del af sprint reviewet vil ikke gøre sig gældende, da vi ikke vil fortsætte projektet herefter.

### SPRINT RETROSPEKT

*Af Patrick og Matthias*

#### Positivt

1. Vi har i løbet af denne sprint forbedret vores indsats i forhold til versionskontrol og reviews. Der har overordnet set været lagt mere fokus på disse, så der for hver tilføjelse til rapporten eller efter hver afsluttet task, er blevet lavet reviews.
2. Vores fokus på planlægning var også i højsædet i løbet af denne sprint. Vores planlægning inden sprint start og vores vedligeholdelse og punktlighed i forhold til denne planlægning var markant bedre end i første sprint.
3. Vi havde i denne sprint bedre kommunikation, og fordi holdets medlemmer generelt har været bedre til at træde i kraft ift. deres ansvarsområder, havde vi lettere ved at nå til enighed om de problemstillinger vi stødte ind i.
4. Grundet vores bedre planlægning i denne sprint, var fravær blandt alle gruppemedlemmer væsentligt lavere end i første sprint. Alle var klart indforstået med, hvilken mængde arbejdstimer der blev forventet af dem for hver dag og dette resulterede i en konsistent arbejdsdag.
5. I denne sprint var der også færre forstyrrelser udefra. Vi havde afsluttet vores sideløbende projekter og afleveret dem, så vi havde nemmere ved både at planlægge præcist og holde skarpt fokus på vores arbejde. Det påvirkede både vores arbejdsmiljø og vores arbejdsmorale positivt.
6. Vi har som helhed fastholdt de positive punkter fra første sprint, og haft fokus på fortsat både at overholde og forbedre os i dem. Derudover har vi vægtet højt at komme nogle af de negative punkter til livs og forbedre os på disse områder.
7. Alle de ovenstående punkter har bidraget til, at vi i løbet af denne sprint havde øget effektivitet og fokus. Vi kom gnidningsfrit igennem de udfordringer vi stødte på og var gode til at holde gejsten høj på trods af lange arbejdsdage. Alles meninger og holdninger bliver taget til overvejelse og der er en gensidig respekt mellem alle medlemmer, hvilket skaber et godt arbejdsmiljø.

#### Negativt:

1. Vores tidsestimering haltede igen i denne sprint. Vi har forbedret os siden første sprint, hvilket indikerer at vores re-estimering var til fordel. På trods af en forbedring, må vi konstatere at vores estimer ikke er tilfredsstillende, og dette vil vi komme med løsningsforslag til i vores afsnit om ændringer.
2. I forlængelse af vores dårlige estimeringer, danner der sig et klart mønster af, at vi ikke har været gode nok, til at samle tasks hvis vi vurderer dem til ikke at være tilstrækkelige nok til 1 times estimat. Dette havde hjulpet markant på vores estimer i både user story 9 og 10, som kun tog hhv. 5 og 15 minutter. Derudover var der en håndfuld af vores tasks, som hverken passede på 2, 4 eller 8 timer, men i større grad ramte imellem disse estimerings værdier. I disse tilfælde skulle vi have været bedre til at dele dem op eller slå dem sammen med andre tasks, for at ramme et mere præcist estimat.

3. Vi har tre user stories som kun indeholder en enkelt task. Vi følte os nødsaget til at gøre dette, da vi fik nye krav fra vores kunde, men da sprinten var påbegyndt, opdagede vi at det havde givet bedre mening at sætte dem sammen med andre eksisterende user-stories.
4. Vi stødte i løbet af sprinten på en uforudset udfordring. Vi blev desværre overrasket af håndværkere i vores faste arbejdslokale og var af den grund nødsaget til at relokere for dagen. Dette resulterede i at vi flyttede station fra København til Herlev, hvilket gjorde at vi blev forsinket i forhold til denne dags planlægning. Grundet dette var vi nødt til både at forlænge vores arbejdsdag og samtidig omstrukturere vores plan for den efterfølgende dag. På trods af at have en negativ indvirkning på vores projekt, var vores håndtering af situationen god og påvirkede os ikke mere end hvad der kunne undgås.
5. I løbet af denne sprint, blev vi klar over at vores baseline ikke gav særlig stor værdi grundet projektets omfang. Dette resulterede i at vi ikke fik vedligeholdt og opdateret vores baseline som ønsket, men at det i stedet faldt lidt i baggrunden. Vi prioriterede vores versionskontrol og planlægning. På trods af vores manglede fokus på baseline, havde det ikke konsekvenser, da vi i vores team har god kommunikation indbyrdes hvilket bidrog til et godt overblik.
6. Vi blev i løbet af sprinten enige om, at vores daglige arbejdsplan og logbog ikke var til stort gavn, i forhold til den tid vi brugte på det. Vi arbejdede i et fast lokale, med et online SCRUM-board og et OneNote dokument med oversigt over TO-DO, derfor ville en daglig logbog og arbejdsplan være unødvendig.

#### **Ændringer:**

1. Fremadrettet vil vi have større fokus på at sammenlægge eller opdele vores tasks, så vi kan få en mere præcis estimering i vores Planning Poker. Derudover vil vi lave en bedre dybdegående analyse af, hvilke tidligere komponenter vi kan genbruge eller tilpasse til hver enkelt task. Vi har diskuteret indbyrdes i gruppen og kommet frem til, at vi eventuelt kunne bruge andre estimeringsmetoder end planning poker til vores tasks - evt. function points.
2. Vi har på trods af færre problemer, stadig haft problemer med vores versionsstyring. Primært har vores rapport, som skrives i Word, ikke været særlig kompatibel med GitHubs versionsstyring. Vi stødte på problemer med at tekst og andre værdier forsvandt, hvis rapporten var åben idét man hentede seneste opdatering. Derudover var det ikke muligt at arbejde flere på rapporten samtidigt - derfor var vi nødt til at lave vores arbejde i eksterne filer, for senere hen at samle det i vores hovedrapport. For at komme dette til livs, vil vi fremadrettet bruge redskaber som google docs, OneNote eller WordOnline, hvor ændringer opdateres live.



## TASKS ESTIMATION VS. TASKS ACTUAL

Af Casper & Matthias

Her vises en tabel indeholdende alle tasks, både færdige og ufærdige. Tabellen indeholder fire kolonner: task ID, den første estimerede tid, den anden estimerede tid, og det den tid vi rent faktisk brugte på tasken.

Efter hver sprint diskuterer vi løbende, om vores tasks skal reestimeres. Estimeringernes version angiver hvilken sprint de er estimeret til:

TASK ID	ESTIMATED TIME 1.0 (TIMER)	ESTIMATED TIME 2.0 (TIMER)	ACUAL TIME (MIN)
US01.1	1,5	-	110
US01.2	2	-	80
US01.4	2	-	30
US01.5	2	-	90
US01.6	2,5	-	55
US01.7	1,5	-	120
US01.8	3,5	-	70
US02.1	1	-	25
US02.3	2,5	-	60
US02.4	1,5	-	40
US02.5	2	-	20
US02.6	3	-	15
US03.1	3	3,5	75
US03.2	4	1	30
US03.3	-	1	50
US04.1	5,5	1,25	59
US04.2	3,5	3,75	50
US04.3	2	1	20
US04.4	-	2	30
US05.1	1	2	39
US05.2	2	1,75	65
US05.3	2	2,25	75
US05.4	-	2,25	23
US06.1	1	2,25	20
US06.2	1,5	Lagt sammen med 6.1	X
US07.1	7,5	10	X

<b>US07.2</b>	3,5	1,75	X
<b>US08.1</b>	-	2,25	
<b>US08.2</b>	-	1	
<b>US08.3</b>	-	2,25	130
<b>US09.1</b>	-	1	5
<b>US10.1</b>	-	1	15

## INFORMAL REVIEWS

*Af Matthias*

Vi har løbende igennem projektet, benyttet os af informal reviews. Dette har medført at vi indbyrdes i gruppen på alle tidspunkter har været indforstået med, hvad de resterende gruppemedlemmer arbejdede med og løbende kunne kvalitetssikre alle dele af både rapport og program.

Denne form for review har ikke kun været benyttet på koden, men også når et afsnit i rapporten, en UML-model eller andre artefakter blev lavet.

Hvis der blev fundet eventuelle fejl eller mangler i løbet af disse informal reviews, tog vi diskussionen i fælles forum, hvor disse fejl/mangler efterfølgende skulle udbedres af den ansvarlige.

## FORMAL REVIEWS

*Af Matthias*

Formal reviews har fungeret således, at vi sad i to forskellige pair-programming par, og hver gang et par havde afsluttet en task, blev den sat i 'test' på vores SCRUM-board. Herefter lavede den anden gruppe et formal review på det færdige stykke kode ved lejlighed, før det blev godkendt og sat i 'Done' kolonnen.

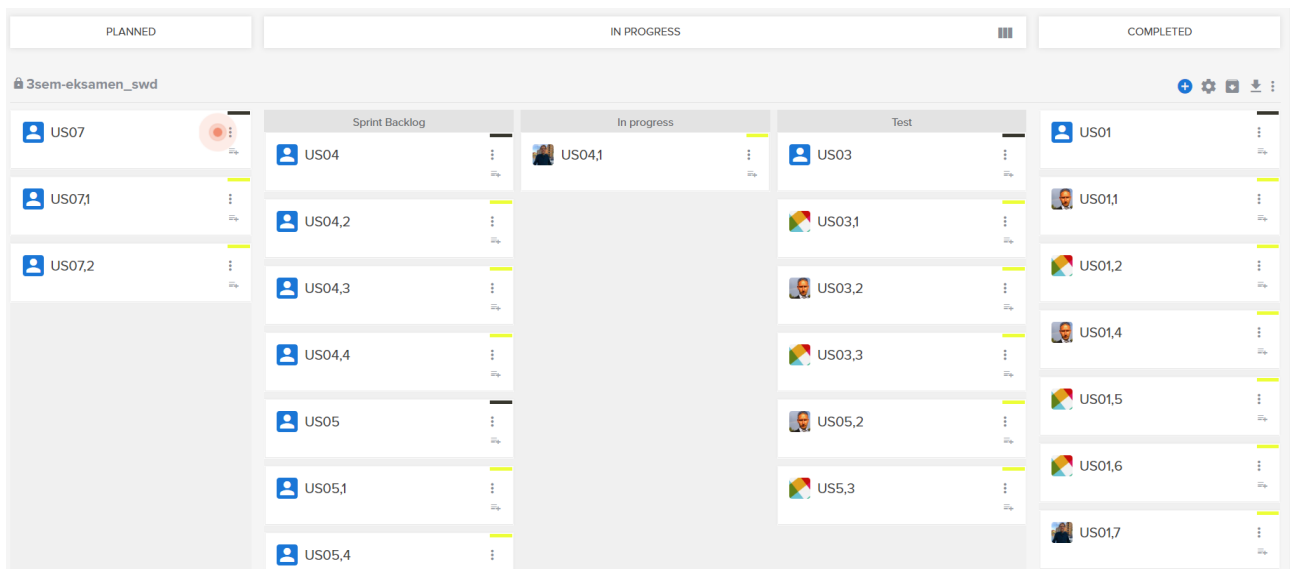
Vores formal reviews havde de samme tre udfald som et formal review som regel har: Godkendt, afvist(og senere hen, nyt formal review) eller små tilføjelser som skulle ændres, før det kunne godkendes - men uden behov for endnu et review. Hvis et formal review blev afvist, skulle den ansvarlige lave ændringer og sende det til informal review igen. Hvis reviewet blev godkendt, blev det placeret i 'done' kolonnen, hvor vi efterfølgende lavede 'end tests' på de tasks, hvor det var vigtigt at de havde korrekte outputs kontra inputs.

Selve produktet af de formelle reviews, resulterede i de ovennævnte forskellige muligheder, men miljøet omkring reviewet lænede sig i større grad op af informal review.

## SCRUM-BOARD

Af Christian

Vi har brugt scrumboard i løbet af dette projekt, både for at holde en konkret struktur igennem hele processen, og for at få en visualisering af hvor vi har været i forløbet. Et scrum board er et værktøj der hjælper et team med at få en visuel forståelse af, hvilke tasks der skal laves og hvordan prioriteringen er sat op.



8; Snippet af vores scrum-board i løbet af sprint 2, der viser hvor vi er i forløbet, og hvad status de forskellige tasks har.

Dette værktøj har hjulpet teamet med at holde et overordnet overblik, over hvilke opgaver der skal løses i det igangværende sprint. Det har skabt en tryghed og en stabilitet for teamet, da der aldrig har været tvivl om hvilke opgaver der skal løses, og hvilke der er afsluttet. Derved var man eksempelvis ikke i tvivl om der kunne arbejdes på en task, der eventuelt var afhængig af en anden task. Det fungerede også godt som motiverende redskab, da man løbende kunne se en fremgang i sprintet.

Vi har valgt at tilføje en "test"-kolonne på vores board, for at indikere hvilke tilføjelser (i programmet) der afventede et formal review. Derved herskede der aldrig tvivl om hvilke funktioner der var blevet godkendt, og samtidig blevet en del af baseline, som bliver indikeret af vores "done"-kolonne.

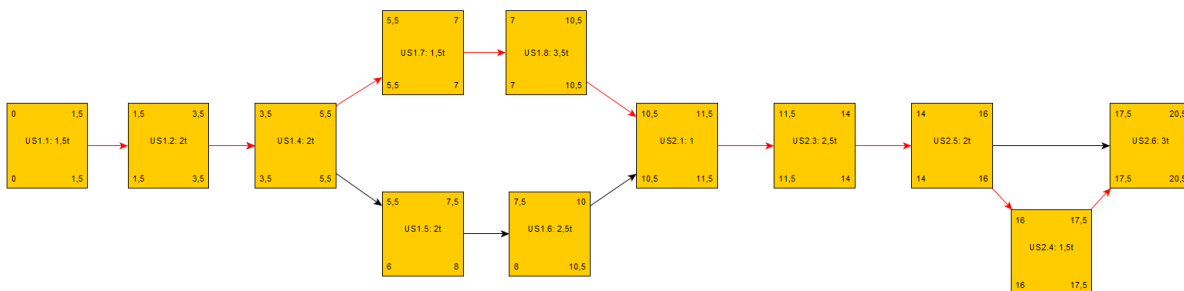
*Af Patrick og Christian*



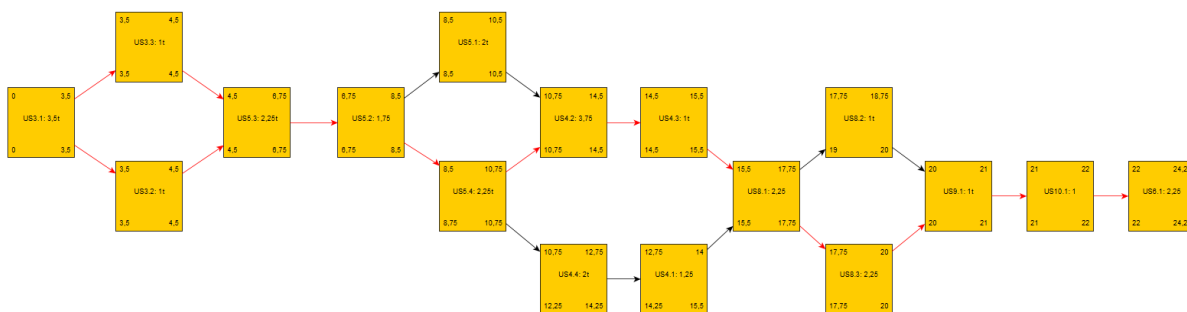
67

## TASK NETWORKS

Af Casper og Christian



10: task network for sprint 1



11: task network for sprint 2

Vores Task-network viser hhv. sprint 1 og 2's start og slutpunkt samt hvilke tasks der er afhængige af hinanden. Inde i tasken kan vi øverst se Early-start og Early-finish som er det tidspunkt vi tidligst kan gå i gang med tasken. Nederst i taskene kan vi se Late-start og Late-finish hvilket viser hvad tid vi senest skal gå i gang med denne task for at blive færdige til tiden. Derudover vises det i midten af "kasserne" hvilken task der er tale om, og hvad tidsestimatet ligger på. De sammenhængende røde pile er Critical Path hvilket er den vej til slutproduktet som viser hvornår vi tidligst kan være færdige med produktet. Disse diagrammer viser også slack, hvilket er den tid hvor der er en difference mellem early start og late start. Slack opstår ved at de tasks som ikke er en del af critical path kan være færdige hurtigere end de tasks der er i critical path, det er altså den tidsforskel der bliver skabt når taskene der ikke er i critical path kan startes senere end early start. Man udregner slack ved late start - early start.

Task network har for os været en god visualisering af vores tasks og rækkefølgen de heri skal laves for at blive færdige på det aftalte tidspunkt. Det hjælper os også med at påbegynde nye tasks sideløbende med de tasks der allerede er i gang, for at sørge for at ingen sidder og venter på at nogen bliver færdige med den task før at de kan begynde. Det giver os derfor nogen klare guidelines for hvornår vi skal begynde at udvikle tasks.

## GLOSSARY

Af Matthias & Christian

UDTRYK	BESKRIVELSE
<b>Kunde(brugertype)</b>	En kunde der benytter sig af applikationen.
<b>Firma(brugertype)</b>	Et firma der benytter sig af applikationen.
<b>Admin / administrator(brugertype)</b>	En administrator der benytter sig af applikationen.
<b>Bruger</b>	Bruger dækker over både kunde, firma, admin eller andre som benytter sig af applikationen.
<b>Pattern</b>	Her henvises til HTML funktionen pattern, der gør det muligt at bestemme hvilken syntaks der skal indtastes på input-fields, før de bliver godkendt. Eks: registrerings-nr. må kun være 4 cifre, derfor sættes pattern til 4 tal mellem 0-9.
<b>Title(s)</b>	Titles henviser til HTML funktionen, som vi benytter os af, når vi ønsker at give brugeren en fejlmeddelelse når et Pattern ikke er blevet overholdt. Således kan brugeren se, hvad der skal ændres før at informationerne er gyldige.
<b>Product owner</b>	En ansvarlig person, der sørger for at opgaver bliver prioriteret (sammen med kunden) korrekt, for at skabe mest værdi for kunden.
<b>Stakeholder</b>	En interessegruppe der har med produktet at gøre. (kunde, ansat, samarbejdspartner og lign.)
<b>SCRUM-master</b>	En ikke-autoritær leder, der assisterer det resterende team med at følge processen.
<b>Project manager</b>	Ansvarshavende for at et projekt kommer i mål, og tilsvarende planlægning af dette. Ydermere står vedkommende i spidsen for at styre teamet, og de ressourcer der er tilegnet.
<b>SaaS(Software as a service)</b>	Software As A Service giver en bruger mulighed for at oprette forbindelse til cloudbaserende applikationer via. internettet. SaaS fungerer som en komplet softwareløsning, som organisationer kan leje sig ind på - almindelige eksempler på SaaS kunne være: mail, kalender og Microsoft Office. <sup>1</sup>
<b>White-label</b>	En white-label løsning, er når et firma eller en organisation har udviklet et produkt, som andre firmaer/organisationer kan benytte sig af, og at

<sup>1</sup> <https://azure.microsoft.com/da-dk/overview/what-is-saas/>

	det udadtil virker som om at de selv har udviklet produktet.
<b>YourLocal</b>	YourLocal er en ny dansk app, der giver spontane tilbud fra lokale forretninger i dit nærområde. Samtidig hjælper du med at bekæmpe madspild. YourLocal er en dansk app, hvor du kan finde spontane tilbud og afhjælpe mad- og resourcespild. <sup>2</sup>
<b>CTA</b>	Call to Action er, som det engelske ord direkte betyder, en "opfordring til en handling". CTA er et meget brugt begreb i onlineverden, og bruges rigtig mange steder, hvor man vil opfordre sine kunder / brugere til en handling på sin Webshop eller hjemmeside. <sup>3</sup>
<b>Lorem Ipsum</b>	Lorem ipsum, også kendt som mumletekst, er en fyldtekst. Teksten bruges som fyldtekst, for at ordene ikke skal forstyrre det grafiske udtryk under selve layout-processen. <sup>4</sup>
<b>Virksomhed/firma/butik</b>	Tre ord der dækker over samme begreb de fleste steder det er anvendt i løbet af rapporten: Henviser til den virksomhed/firma/butik der har/ønsker at oprette sig på TheNext whitelabel løsning for YourLocal, som senere hen kan oprette og sælge tilbud på platformen.
<b>Product backlog</b>	En sorteret liste af elementer (tasks), der er i den rækkefølge teamet forventer disse bliver udført i.
<b>PaaS</b>	Platform As A Service er en tjeneste, der giver mulighed for at tilkøbe de tjenester du har brug for, alt afhængig af dine behov, fra en cloudbaseret udbyder, og får adgang til dem fra en sikker internetforbindelse.
<b>Geolokation</b>	En måde hvorpå man kan bestemme en præcis lokation, på for eksempel computer eller telefon. Dette sker oftest via GPS koordinater.
<b>API-kald</b>	En måde hvorpå man kan hente eksempelvis data fra et system, i et andet system. Således interagerer et stykke software med et andet.
<b>REST API</b>	En RESTful API er et applikations interface som bruger HTTP requests til at GET, PUT, POST og DELETE data.

<sup>2</sup> <https://www.mobilsiden.dk/tips/yourlocal-fa-spontane-tilbud-pa-mobilen-og-bekaemp-madspild-tip.lid.34854/>

<sup>3</sup> <https://www.medialine.dk/blog/lav-en-call-to-action-knap.html>

<sup>4</sup> [https://da.wikipedia.org/wiki/Lorem\\_ipsum](https://da.wikipedia.org/wiki/Lorem_ipsum)

<b>JPA</b>	"Java Persistence API"(se ovenfor for API) bruges til at styre relationel data, i applikationer (eksempelvis MySQL database) der understøtter Java.
------------	---

## KONKLUSION

### *Af Alle*

De krav kunden gav os som udgangspunkt, og de tilføjede krav vi fik stillet undervejs, har vi mødt, med undtagelse af kravet om geolokation. Denne feature vurderede vi ikke til at være nødvendig for en fungerende prototype, men vil være relevant for fremtidige inkrementeringer.

Kunden har været tilfreds med vores produkt efter hver inkrementering, og pga. en god dialog og forståelse for projektet, har kunden kunne komme med supplerende og/eller nye krav efter hver afsluttet sprint.

Kunden har haft et meget specifikt sæt krav til opsætning af programmet, hvilket både har haft fordele og ulemper.

Det har været positivt at kunden vidste hvad projektet skulle indeholde, og det har derfor sparet os for meget analyse og klargørelse af projektets omfang.

Til gengæld har kunden stillet nogle krav, som vi ikke har haft mulighed for at møde i de første inkrementeringer. Vi løste dette i samarbejde med kunden, og konkluderede, at kravet ville høre til i en inkrementering længere ude i fremtiden.

Den procesmodel vi har arbejdet ud fra, har været det rigtige valg til dette projekt, da vi ikke har skulle opfylde krav fra dag til dag i et hurtigt udviklende marked. Der har været mere fokus på kvalitet og funktionalitet, end på kvantitet og hurtig aflevering, hvilket har givet os mulighed for eftertænkksomhed i forhold til alle de funktionaliteter vi har afleveret.

De diagrammer vi har valgt at inkludere i rapporten, har givet os et godt overblik, og en overordnet god forståelse for systemets arkitektur, samt en god planlægningsstruktur. Dette har resulteret i at alle medlemmer har haft et konstant overblik i projektets udvikling.



## LITTERATURLISTE

Af Matthias

- Bootstrap: Components. Udgivet af Bootstrap. Internetadresse: <https://getbootstrap.com/> - (Internet)
- Bootsnipp: Templates. Udgivet af Bootsnipp. Internetadresse: <https://bootsnipp.com/> - (Internet)
- Thymeleaf: Documentation. Udgivet af Thymeleaf. Internetadresse: <https://www.thymeleaf.org/> - (Internet)
- Workstreams: Workstream flows in Slack. Udgivet af Workstreams. Internetadresse: <https://www.workstreams.ai/> - (Internet)
- Larman, Craig: *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3. udg. Pearson Education, Inc, 2004. Database: - Besøgt d. .. (Bog)
- SCRUM: Resources. Udgivet af Scrum. Internetadresse: <https://www.scrum.org/> - Besøgt(Internet)
- Wikipedia: Lorem Ipsum. Udgivet af Wikipedia. Internetadresse: [https://da.wikipedia.org/wiki/Lorem\\_ipsum](https://da.wikipedia.org/wiki/Lorem_ipsum) - Besøgt d. 16.12.2018 (Internet)
- Microsoft : What is SaaS. Udgivet af Microsoft Azure. Internetadresse: <https://azure.microsoft.com/da-dk/overview/what-is-saas/> - Besøgt d. 16.12.2018 (Internet)
- MobilSiden: Undgå madspild. Udgivet af MobilSiden. Internetadresse: <https://www.mobilsiden.dk/tips/yourlocal-fa-spontane-tilbud-pa-mobilen-og-bekaemp-madspild-tip,lid.34854/> - Besøgt d. 16.12.2018 (Internet)
- MediaLine: Call to Action. Udgivet af MediaLine. Internetadresse: <https://www.medialine.dk/blog/lav-en-call-to-action-knap.html> - Besøgt d. 16.12.2018 (Internet)
- Roger Pressman, Bruce Maxim - *Software Engineering - A Practitioner's Approach* 8th Edition