TheNext

3. Semester Eksamensprojekt i Software Design

Casper Frost, Patrick Sirich, Matthias Skou, Christian Strunge

2018

Indhold

[Problemformulering 3](#_Toc532387147)

[Introduktion 4](#_Toc532387148)

[Glossary 4](#_Toc532387149)

[Opsætning 5](#_Toc532387150)

[Rollefordeling og ansvarsområder\* 5](#_Toc532387151)

[Arbejdsmiljø 5](#_Toc532387152)

[Coding standards 5](#_Toc532387153)

[Design standards 5](#_Toc532387154)

[Patterns 5](#_Toc532387155)

[Udviklingsmodel 5](#_Toc532387156)

[Argumentation/Methology comparison 5](#_Toc532387157)

[Retningslinjer for udviklingsmodellen 5](#_Toc532387158)

[Analyse 6](#_Toc532387159)

[Kunden 6](#_Toc532387160)

[Risikoanalyse & risikoplan 6](#_Toc532387161)

[Identifikation 6](#_Toc532387162)

[Risikoanalyse 7](#_Toc532387163)

[Risikoplan 8](#_Toc532387164)

[Kravspecifikation 10](#_Toc532387165)

[Paper prototype 11](#_Toc532387166)

[User stories 12](#_Toc532387167)

[Tasks 13](#_Toc532387168)

[US01 13](#_Toc532387169)

[US02 13](#_Toc532387170)

[US03 14](#_Toc532387171)

[Us04 14](#_Toc532387172)

[Us05 15](#_Toc532387173)

[US06 16](#_Toc532387174)

[US07 16](#_Toc532387175)

[US08 16](#_Toc532387176)

[US09 17](#_Toc532387177)

[US10 17](#_Toc532387178)

[Gantt-chart 18](#_Toc532387179)

[Metrics\* 18](#_Toc532387180)

[SPRINT 1 18](#_Toc532387181)

[SPRINT PLANNING 18](#_Toc532387182)

[Hvad kan vi aflevere som følge af den kommende sprint? 18](#_Toc532387183)

[Hvilke arbejdsmetoder/værktøjer skal vi anvende? 18](#_Toc532387184)

[Sprint goal 18](#_Toc532387185)

[TASK DESCRIPTION 19](#_Toc532387186)

[USER story 1 19](#_Toc532387187)

[USER story 2 19](#_Toc532387188)

[ACCEPTANCE TEST 20](#_Toc532387189)

[Burndown chart 1 22](#_Toc532387190)

[SPRINT review 22](#_Toc532387191)

[Introduktion 23](#_Toc532387192)

[Inkrementeringen 23](#_Toc532387193)

[Demonstration & Feedback 23](#_Toc532387194)

[Diskussion og projektering 24](#_Toc532387195)

[Sprint retrospekt 24](#_Toc532387196)

[Positivt 24](#_Toc532387197)

[Negativt 24](#_Toc532387198)

[Ændringer (process improvement)\* 25](#_Toc532387199)

[Sprint 2 26](#_Toc532387200)

[Sprint planning 26](#_Toc532387201)

[Rådighedstimer 26](#_Toc532387202)

[Gennemgang af sprint backlog 26](#_Toc532387203)

[TASK DESCRIPTION 27](#_Toc532387204)

[User story 3 27](#_Toc532387205)

[USER story 4 28](#_Toc532387206)

[USER story 5 28](#_Toc532387207)

[USER story 8 29](#_Toc532387208)

[Burndown chart 2 30](#_Toc532387209)

[Sprint review 30](#_Toc532387210)

[Sprint retrospekt 30](#_Toc532387211)

[Tasks estimation vs. Tasks actual(\*) 31](#_Toc532387212)

[Informal reviews 32](#_Toc532387213)

[Formal reviews 32](#_Toc532387214)

[Konklusion 32](#_Toc532387215)

[Bilag 32](#_Toc532387216)

# 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 have 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 andet 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 Your Local, som er drevet af The Next, en serviceudbyder vi vil redegøre for i rapportafsnittet om kunden.

# Problemformulering

*Af Alle*

# Glossary

*Af Matthias & Christian*

|  |  |
| --- | --- |
| ORD | BESKRIVELSE |
| Kunde(bruger-type) | En kunde der benytter sig af applikationen. |
| Firma(bruger-type) | Et firma der benytter sig af applikationen. |
| Admin / administrator(bruger-type) | En administrator der benytter sig af applikationen. |
| Bruger | Bruger dækker over både kunde, firma, admin eller andre som benytter sig af applikationen. |
| Pattern | 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. |
| Title(s) | 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. |
| Product owner | En ansvarlig person, der sørger for at opgaver bliver prioriteret (sammen med kunden) korrekt, for at skabe mest værdi for kunden. |
| Stakeholder | En interessegruppe der har med produktet at gøre. (kunde, ansat, samarbejdspartner og lign.) |
| SCRUM-master | En ikke-autoritær leder, der assisterer det resterende team med at følge processen. |
| Project manager |  |
| SaaS(Software as a service) |  |
| White-label |  |
| Widget |  |
| YourLocal |  |
| CTA |  |
| Lorem Ipsum |  |
| Virksomhed/firma/butik |  |
| Product backlog | En sorteret liste af elementer (tasks), der er i den rækkefølge teamet forventer disse bliver udført i. |
| Sprint planning | Et møde med alle teamets medlemmer, hvori der bliver skabt en forståelse, og der bliver valgt hvilket arbejde der skal laves i det sprint der skal startes. |

# Opsætning

## Rollefordeling og ansvarsområder\*

*Af Casper*

Patrick: Product owner & stakeholder-fuldmægtig

Casper: Rapport-ansvarlig & stand-in product owner

Matthias: Kode- og task-ansvarlig

Christian: SCRUM-master & Project manager

## Arbejdsmiljø

Mangel på fast lokale, arbejdsro, forstyrrelser.  
Sideløbende undervisning og mandatory projekter(og div. fremlæggelser), svært at planlægge en sprint og overholde den.  
Problemer med intelliJ, gentagende forsinkelser, workspace xml.

### Coding standards

### Design standards

### Patterns

## Udviklingsmodel

### Argumentation/Methology comparison

SCRUM er task driven, ikke test. Vi implementerer dog acceptance tests i vores proces, de kommer dog først til anvendelse under test-fasen.

Vi bruger vision og mission fra UP?

### Retningslinjer for udviklingsmodellen

# Analyse

*Af Patrick & Matthias*

## 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.la. 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 | * Misforståelse af krav til kunden * Drastisk ændring af krav mod slutningen af projektet * Mangel på lokale |
| Personalerisiko | * Længerevarende sygdom * Gruppemedlem frafalder projektet * Tidsoverskridelse i sprints |
| Kunderisiko | * Nedskæringer i budget * Nedlukning af virksomhed |
| Ressourcerisiko | * Hardwarenedbrud * Tab af data (overskrivning af data etc.) * Fejl i version control * Servernedbrud |
| Estimeringsrisiko | * Misforståelser i planlægning * Større udfordringer i koden * Tests viser store fejl * Mindre afvigelser i tidsestimering * Større afvigelser i tidsestimering * Uforudsete sideløbende studierelaterede arbejdsopgaver |
| Produktrisiko | * Et produkt der ikke er brugervenligt nok * Fejlkommunikation der resulterer i manglende/forkerte features |

### 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 |
| Servernedbrud | 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 stakeholder 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. Butik logger på TheNext platformen
2. Butik indtaster deal, som de vil tilbyde til dét community TheNext har aftale med ifht. Butikkens 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 butikken (€ 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
* … og sikkert meget mere som i kan opfinde
* 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)
* … og sikkert meget mere som i kan opfinde
* 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
* Rabat% (0-100)
* 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 brugergrænseflade 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 storys, som danner ramme for de funktionaliteter applikationen skal kunne. User storys 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 storys vil vi bryde op i tasks, som vil blive estimeret, og til sidst præsentere for kunden, som vil prioritere hvilke user storys der vægter højest:

|  |
| --- |
| User stories |
| Som firma skal jeg kunne oprette en bruger, så jeg kan logge ind senere med relevant indhold og rettigheder. |
| 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. |
| Som firma skal jeg kunne oprette nye tilbud, som skal vises til kunder, så kunderne har mulighed for at købe vores tilbud. |
| 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. |
| Som kunde skal jeg kunne købe et tilbud, så jeg er sikker på at mit køb er blevet godkendt. |
| Som firma skal jeg kunne se en oversigt over hvor mange tilbud der er blevet købt, så jeg kan holde statistik over mine salg. |
| 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. |

### Paper prototype

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 protoype.



## TaskS

*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*.

Herunder en opstilling af tabeller, som hver indeholder en user story, dennes tasks, og en three point value og planning poker estimering ud fra:

### 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 |

### 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 |

### 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 |

### 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 |

### 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 restrictions. | 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 |

### 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 |

### 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 modtage en meddelelse om 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 |

### 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 |

### 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 |

### 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 |

## Gantt-chart

## Metrics\*

Planning poker

Three-point-estimation

# SPRINT 1

## SPRINT PLANNING

*Af Christian & Matthias*

### Hvad kan vi aflevere som følge af den kommende sprint?

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(med 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.

### Hvilke arbejdsmetoder/værktøjer skal vi anvende?

Vi vil i løbet af denne sprint, arbejde i pair-programming og løbende sørge for at holde vores SCRUM-board samt burndown-chart opdateret. Vi vil benytte os af JPA til at oprette samt administrere vores database. Vi vil følge vores aftalte coding-standarder og arbejde ud fra aftale design-patterns, med især MVC i fokus. Vi vil benytte os af Spring til at lave web-applikationen og bootstrap(eller evt. egen css) til designet af applikationen. Vi vil skrive acceptance-tests til alle vores tasks og løbende sørge for at de er blevet opfyldt, før tasken kan godkendes.

### Sprint goal

Målet med denne sprint, er at vi ud fra vores kundes prioriterer, kan aflevere et færdigt produkt som kan benyttes selvstændigt. Kunden har prioriteret disse user-stories, da det er vigtigt at firmaer tidligst muligt kan oprette sig og være forberedt inden applikationen officielt bliver opført.

## TASK DESCRIPTION

*Af Matthias & Christian*

### USER story 1

|  |  |  |
| --- | --- | --- |
| ID | TASK | BESKRIVELSE |
| US01.1 | 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. |
| US01.2 | 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. |
| US01.3 | 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. |
| US01.5 | 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](https://www.facebook.com/) ManyToMany tabel indeholdende id på category og id på firma. Et firma skal også kunne vælge ikke at tilknytte kategorier. |
| US01.6 | Funktion til redigering af stamdata. | 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).  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. |
| US01.7 | 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. |
| US01.8 | Login-funktion der adskiller de forskellige brugere via login. | 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.  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 differenciere hvorvidt om det er et firma eller ej. |

### USER story 2

|  |  |  |
| --- | --- | --- |
| ID | TASK | BESKRIVELSE |
| US02.1 | 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. |
| US02.3 | Underside til kundeinformation med kategorier. | 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.  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.  Derudover skal der laves checkboxes til alle eksisterende kategorier, så kunden kan vælge hvilke kategorier personen har interesse for.  Alle disse felter skal gemmes korrekt i databasen, såfremt at kundens input er gyldigt. |
| US02.4 | 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. |
| US02.5 | 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. |
| US02.6 | Funktion til redigering af stamdata. | 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.  Det skal ikke være muligt at indtaste ugyldige inputs, felterne skal følge anvendte patterns og titles fra opret firma. |

## ACCEPTANCE TEST

*Af alle*

Vi har valgt at benytte denne form for acceptance test, hvor vi i ID kolonnen har opsat det på følgende made: Testen er sat op således:

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

Vi har taget udgangspunkt I en række tasks fra vores US01, for at demonstrere hvordan testsne 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-templaten |
| 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 |
| US01.8.2 | En bruger logger ind på applikationen | Login-typen er ikke et firma | Programmet genkender ikke login-typen og viser en fejlmeddelelse |
| US01.8.3 | 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 |

## Burndown chart 1

Eftersom denne sprint indeholdt en del opsætning og nogle opstartsproblemer, samt et sideløbende projekt, ser vores burndown-chart derfor lidt atypisk ud.

## 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 stakeholders 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 stakeholders, og dermed have deres 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 deltagenes roller.
2. Et oplæg fra product owneren der fortæller, hvilke Product Backlog produkter der er blevet færdiggjort (og eventuelle user storys der ikke blev).
3. Udviklerne forklarer kortfattet hvilke problemer der blev mødt, og hvordan de blev løst. Fokus vil være på problemer, som stakeholders kunne have interesse i, og ikke tekniske komplikationer.
4. En demonstration af inkrementeringen, hvor stakeholder 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 stakeholderen – 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 de/den repræsentative stakeholder(s) ikke føle at det der bliver demonstreret, er et færdigt produkt, som der skal bedømmes som helhed. Stakeholders 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 stakeholders 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 Storys 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 stakeholder 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 redigére sine oplysninger, oppe i højre hjørne ved siden af en ’log ud’ knap.
   1. 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 Storys:

|  |
| --- |
| USER STORYS |
| 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 ikke-eksisterende bruger skal jeg kunne oprette en bruger fra login-siden. |

### Diskussion og projektering

Efter demonstrationen vil vi kigge på vores Product Backlog, og lave et sandsynligt tidsestimat af hvornår projektet forventes færdigt.

Vi har tilføjet user story 8, 9 og 10. Nævn noget om kundens prioritering for næste sprint.

## 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 udviklerene 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 arbejdsgang 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.

1. 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 arbejdsgang, 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 klone 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.

1. 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.
2. 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.
3. 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 unødig forvirring. Manglende definition på den enkelte artefakt til baselinen, har desuden også haft indflydelse på forvirringen.

1. /\* I vores kontrakt har vi vedtaget, at fraværende gruppemedlemmer får uddelegeret arbejde for at vi kan overholde vores estimeringer. Nogle medlemmer af gruppe var ikke tilfredse med, at dette ikke blev overholdt, og derfor vil der blive sat ekstra fokus på dette. \*/

### Ændringer (process improvement)\*

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 storys, 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 | ONSDAG | TORSDAG | FREDAG |
| Matthias | 7 | 5 | 5 |
| Casper | 5 | 5 | 5 |
| Christian | 7 | 5 | 5 |
| Patrick | 4 | 4,5 | 5 |

### Gennemgang af sprint backlog

I vores product backlog har vi følgende User Storys:

US03

US04

US05

US06

US07

US08

US09

US10

Kunden har prioriteret de næste user storys således:

1. US03
2. US05
3. US04
4. US08
5. US09
6. US10
7. US06

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 redigére 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.

## TASK DESCRIPTION

*Af Matthias*

### User story 3

|  |  |  |
| --- | --- | --- |
| ID | TASK | BESKRIVELSE |
| US03.1 | Underside til oprettelse af tilbud, med tilknytning af billede. | 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.  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.  Derudover skal der være en ’Reset’ knap, som fjerner alt indtastet information og en ’Cancel’ knap der redirecter brugeren tilbage. |
| US03.2 | Redigerings- og slettefunktion af eksisterende tilbud | 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.  Når brugeren trykker gem, skal informationerne gemmes korrekt i databasen, såfremt de er gyldige.  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.  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. |
| US03.3 | Underside med oversigt over aktive tilbud | 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. |

### USER story 4

*Af Matthias*

|  |  |
| --- | --- |
| ID | BESKRIVELSE |
| US04.1 | 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.  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. |
| US04.2 | 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. |
| US04.3 | På den nævnte underside i US04.2, skal der også eksistere en slettefunktion, der præsenteres i form at 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. |
| US04.4 | 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. |

### USER story 5

*Af Matthias*

|  |  |
| --- | --- |
| ID | BESKRIVELSE |
| US05.1 | 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. |
| US05.2 | 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. |
| US05.3 | 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. |
| US05.4 | 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.  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. |

### USER story 8

*Af Matthias*

|  |  |  |
| --- | --- | --- |
| ID | TASK | BESKRIVELSE |
| US08.1 | Opret brugerspecifikke navigeringsbarer på alle undersider tilknyttet brugeren. | Der skal implementeres 3 forskellige slags navigationsbarer på applikationen, som skal indeholde følgende:   * **Kunde**   + Logo (jf. US8.2)   + Tilbud (Aktive tilbud, købshistorik, jf. US5.4)   + Kontoprofil (redigering) (jf. US2.6) * **Firma**   + Logo (jf. US8.2)   + Tilbud (Oversigt, jf. US03.3)   + Statistik (jf. 6.1)   + Kontoprofil (Redigering, jf. 1.6) * **Administrator**   + Logo (jf. US8.2)   + Kunder (jf. US4.2)   + Firmaer (jf. US4.2)   + Venteliste (jf. US4.4)   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. |
| 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. | Der skal oprettes 3 forskellige forsider, som man skal lande på efter man har logget ind, afhængigt af hvilken bruger-type man er:   * **Kunde**   + 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). * **Firma**   + Forsiden for et firma er opstillet således, at der er 3 widgets med en CTA-knap til hver: ’Opret tilbud’, ’Se statistik’, ’Se mine tilbud’ * **Administrator**   + Forsiden når en administrator logger ind, skal landingssiden være ventelisten over firmaer som skal godkendes eller afvises(se US04.4). |

## Burndown chart 2

## Sprint review

## Sprint retrospekt

# Tasks estimation vs. Tasks actual(\*)

*Af Casper & Matthias*

Her vises en tabel indeholdende alle tasks, både færdige og ufærdige. Tabellen indeholder tre kolonner: task ID, den 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 afspejler 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 |  |
| US05.2 | 2 | 1,75 | 65 |
| US05.3 | 2 | 2,25 | 75 |
| US05.4 | - | 2,25 |  |
| US06.1 | 1 | 2,25 |  |
| US06.2 | 1,5 | Lagt sammen med 6.1 |  |
| US07.1 | 7,5 | 10 |  |
| US07.2 | 3,5 | 1,75 |  |
| US08.1 | - | 2,25 |  |
| US08.2 | - | 1 |  |
| US08.3 | - | 2,25 |  |
| US09.1 | - | 1 |  |
| US10.1 | - | 1 |  |

# Informal reviews

# Formal reviews

|  |  |  |  |
| --- | --- | --- | --- |
| ID | HVIS | NÅR | SÅ |
| US01.7.1 | Et firma opretter en konto | Informationerne er gyldige | Opret bruger I database |
| US01.7.2 | Et firma opretter en konto | Informationerne er ugyldige | Gemmes data ikke, og fejlmeddelelse vises |
| US01.7.3 | Et firma opretter en konto | E-mail allerede findes I database | Gemmes data ikke, og fejlmeddelelse om duplikering vises |

Lav tests på:

* US5.1

# Konklusion

Cays opfordring: Skriv om hvordan det har været at have en kunde, som på forhånd har stillet krav til databaseopsætningen. Fordel/ulempe?

# Bilag