

2. semesters rapport, 1. årsprøve, Datamatiker, EASJ Roskilde.



Indhold

[Introduktion. 5](#_Toc532543988)

[Problemformulering. 5](#_Toc532543989)

[Første iteration 7](#_Toc532543990)

[Inceptionfasen 7](#_Toc532543991)

[Projektetablering 7](#_Toc532543992)

[Forventningsafstemning 8](#_Toc532543993)

[Risikoanalyse 10](#_Toc532543994)

[Vision og Vision Statement for CC Byg. 12](#_Toc532543995)

[Product position statement 14](#_Toc532543996)

[Introduktion 14](#_Toc532543997)

[Interessentbeskrivelse 14](#_Toc532543998)

[Brugermiljø 14](#_Toc532543999)

[Produktoversigt – krav og behov: 15](#_Toc532544000)

[Kravspecifikation: 15](#_Toc532544001)

[Forretningsanalyse 17](#_Toc532544002)

[SWOT 18](#_Toc532544003)

[Porters Five Forces 20](#_Toc532544004)

[Product Backlog 22](#_Toc532544005)

[Prototype 28](#_Toc532544006)

[Mindmap 29](#_Toc532544007)

[Projektplan 30](#_Toc532544008)

[Business Case 33](#_Toc532544009)

[TESTPLAN 38](#_Toc532544010)

[1.0. Indledning. 38](#_Toc532544011)

[1.1. Formål. 38](#_Toc532544012)

[2.0. SCOPE 38](#_Toc532544013)

[3.0. Antagelser/Risiko 39](#_Toc532544014)

[3.1. Antagelser 39](#_Toc532544015)

[3.2. Risiko 39](#_Toc532544016)

[4.0. Tilgang til projektet. 40](#_Toc532544017)

[5.0. Test miljø 40](#_Toc532544018)

[6.0. Milesten / Aflevering 40](#_Toc532544019)

[6.1.  Test tidsplan 40](#_Toc532544020)

[6.2. Aflevering 42](#_Toc532544021)

[Versionsstyring 43](#_Toc532544022)

[Sekvensdiagrammer 44](#_Toc532544023)

[Kode 47](#_Toc532544024)

[Svar på underspørgsmål 48](#_Toc532544025)

[Underspørgsmål 1: Hvordan kan virksomheden få synliggjort fx manglende materialer? 48](#_Toc532544026)

[Underspørgsmål 2: Hvordan kan virksomheden nemmere håndtere igangværende byggeprojekter og allokere materialer til disse? 48](#_Toc532544027)

[Underspørgsmål 3: Hvordan vil en elektronisk løsning forbedre CC Bygs drift? 48](#_Toc532544028)

[Underspørgsmål 4: Hvordan kan man udvikle et lagerbeholdningssystem i C# med en database som hjælper virksomheden? 48](#_Toc532544029)

[Problemformulering – svar og konklusion. 49](#_Toc532544030)

[Litteraturliste 50](#_Toc532544031)

[Ressourceliste 50](#_Toc532544032)

[Bilag 52](#_Toc532544033)

# Introduktion.

CC Byg er en applikation, der skal bruges af et mindre murerfirma til at holde styr på lager, projekter, udgifter og medarbejdere. Applikationen og udarbejdelsen af denne rapport er lavet ud fra et ønske om at imødekomme kundens krav, behov og forventninger. Håbet er at applikationen vil kunne hjælpe med den daglige projektstyring samt fakturering, da den vil overskueliggøre vareforbruget på de forskellige projekter samt give et overblik over tidligere og igangværende projekter. Applikationen er udarbejdet i samarbejde med kunden med udgangspunkt i den givne problemformulering (se Problemformulering, s. 5).

CC Byg består af ejer samt tre ansatte. Der købes materialer gennem flere forskellige forhandlere samt køres mange byggeprojekter simultant. Til dato har firmaet benyttet sig af kalendere, udprint og egen hukommelse til at holde styr på forbruget af materialer på de forskellige byggesager, hvilket har været både besværligt og tidskrævende - samt giver en forøget risiko for at der sker fejl. Firmaet ønsker derfor en applikation, der kan overskueliggøre hvilke materialer, de forskellige kunder skal modtage fakturaer for. Applikationen skal altså kunne vise lagerbeholdningen for samtlige materialer, ændre i antallet af disse samt tilføje og administrere materialer til forskellige byggesager. Ydermere skal den kunne vise et medarbejderkartotek og tidligere lukkede byggesager.

# Problemformulering.

”Hvordan kan Unified Process (UP), C#-programmering og relationsdatabaser blive brugt til udvikling og implementering af et mindre IT-system?”

Vores underspørgsmål er:

* Hvordan kan virksomheden få synliggjort fx manglende materialer?
* Hvordan kan virksomheden nemmere håndtere igangværende byggeprojekter og allokere materialer til disse?
* Hvordan vil en elektronisk løsning forbedre CC Bygs drift?
* Hvordan kan man udvikle et lagerbeholdningssystem i C# med en database som hjælper virksomheden?

Vores rapport er udarbejdet ud fra ovenstående problemformulering og underspørgsmål. Det er meningen, at den skal kunne læses af kunden i forbindelse med samarbejdet og vil også kunne bruges som forretningsplan ved en eventuel videreudvikling af firmaet. Vores rapport skal vise, at vi kan benytte os af Unified Process, at vi kan programmere i C# samt benytte os af relationsdatabaser. Vi var oprindeligt 5 medlemmer, men vores ene gruppemedlem, Peter Kirkegaard, valgte at stoppe på uddannelsen kort tid efter vi var begyndt på anden iteration. Da Peter valgte at stoppe så tidligt i forløbet, har han ikke haft den store indflydelse eller påvirkning på vores projekt, da han kun var med under projektetableringen (og har stået for én enkelt use case). Vores gruppe har under det fulde projektarbejde derfor bestået af følgende medlemmer:

* Gitte Bilenberg
* Emil Hammer
* Daniel Møller
* Ofelia Symes

# Første iteration

## Inceptionfasen

Inceptionfasen er vores første iteration. Den hjælper med at tydeliggøre hvilket omfang projektet har, hvilken virksomhed vi arbejder med, hvilke forventninger og ønsker brugeren har samt en risikoanalyse over mulige risici. Ydermere har vi udarbejdet en projektetablering, der skal skabe overblik over projektets forløb samt udarbejde grundregler for vores interne samarbejde.

For at kunne lave en virksomhedsanalyse og dens behov, har vi benyttet os af forskellige analyseværktøjer, nemlig Business Model Canvas, Porters Five Forces, SWOT, Parametermix, BCG Matrix og en Business Case.

## Projektetablering

I forbindelse med vores projektopstart, har vi talt om vores styrker og svagheder, og søgt at afstemme vores forventninger og lave nogle faste rammer og aftaler for vores samarbejde. Ud fra disse styrker og svagheder har vi kunnet tydeliggøre, hvem der har hovedansvaret for de forskellige ting – både ift. hvad man er dygtig til, men også i forhold til at få delt viden og sørget for at alle for eksempel har deltaget ligeligt i både design- og programmeringsdelen af projektet. Det har også hjulpet til at tydeliggøre hvor vi hver især har skullet øve os lidt ekstra eller måske været i stand til at lære fra os. Vores forskellige styrker og svagheder er:

|  |  |  |
| --- | --- | --- |
| Navn | Styrker | Svagheder |
| Emil | Programmering | Design |
| Ofelia | Design | Programmering |
| Gitte | Design | Programmering |
| Daniel | Fleksibel | Ingen specialitet |

## Forventningsafstemning

Vores forventningsafstemning er udarbejdet ud fra et ønske om at sikre sig, at vi alle arbejder mod det samme mål og undgå uenigheder i gruppen. Vi har på samme tid fastlagt nogle rammer for hvornår vi skulle arbejde. Forventningsafstemningen skal kunne ses i sammenhæng med vores risikoanalyse – begge er med til at lægge grundstenen for samarbejdet og være med til at minimere risikoen for at vi går fejl af hinanden i vores kommunikation.

Ambitionsniveau:

o   Få et brugbart produkt.

o   At få et færdigt produkt før tid, så der er tid til finpudsning og testning.

o   At levere et ordentligt resultat, hvor rapporten tydeligt viser og understøtter vores programmeringsarbejde.

Regler og aftaler:

o   Mødes mandag, tirsdag, torsdag og fredag fra kl. 9:00 til kl. 14:30.

o   Daglige SCRUM-morgenmøder.

o   Præcise aftaler i forhold til hvad vi sørger for at lave hjemme til næste gang vi mødes eller ved sygdom.

## Risikoanalyse

Udarbejdet af: primært Ofelia.

Risikoanalysen skal – hvis uheldet sker – kunne hjælpe os videre på en effektiv måde, der minimerer følgeskader eller i bedste fald fuldstændigt kunne udelukke mange faldgruber. Faktisk fik vi hurtigt brug for vores risikoanalyse, da vi mistede et gruppemedlem i anden iteration. Her kunne risikoanalysen fortælle os hvordan vi bedst muligt kom videre, uden at det ville påvirke projektet i større grad. I denne situation valgte vi at følge vores opstillede løsning, og minimere omfanget af applikationen en smule, og i stedet arbejde på de ekstra udvidelser til sidst. På samme måde har risikoanalysen kunne understrege vigtigheden af fx at have backupfiler adskillige steder – fx har vi arbejdet i Github, hvor vi har kunnet versionsstyre ved behov, og på rapporten har vi brugt Google Docs til rapportskrivningen i de første tre iterationer, og til slut gået over til Microsoft Word med backup både på dropbox og i Github, da vi her bedre kunne sætte rapporten ordentligt op, men stadig havde behov for sikkerhedskopier og at alle kunne have adgang til rapporten simultant. Samtidig kunne vi udspecificere hvilke risici, vi allerhelst måtte undgå – dette hjalp ”Påvirkning” og ”Sandsynlighed” os med – for eksempel har ”Servernedbrud” en sandsynlighed på 1 og en påvirkning på 5, men var nem at afhjælpe ved at have sikkerhedskopier flere steder. Vi har benyttet risikoanalysen på samme måde i de andre punkter. Både *Sandsynlighed* og *Påvirkning* måles på en skala fra 1-5, hvor 1 er lavest og 5 er højest.

|  |  |  |  |
| --- | --- | --- | --- |
| Beskrivelse/risiko | Sandsynlighed 1-5 | Påvirkning 1-5 | Løsning |
| Servernedbrud | 1 | 5 | Backup på anden service/lokation. |
| Ufærdigt program | 4 | 3 | God planlægning, bedre kommunikation, bedre prioritering |
| Kode, der er svær at opdatere/ videreudvikle ved behov for andre ting | 4 | 3 | Beskrivende kommentarer, overholdelse af GRASP og DRY |
| Databasefejl | 2 | 3 | Ordentlig database- og usecase dokumentation. |
| Overskridelse af deadline for aflevering af projekt | 1 | 5 | Opdatering af planlægning oftere – en fast tidsplan for hvornår iterationerne skal være færdige. |
| Sygdom | 4 | 2 | God kommunikation – man melder ud hurtigst muligt hvis man er syg, og man prøver så vidt som muligt at arbejde hjemmefra ved sygdom. |
| Ændringer i gruppens opbygning | 1 | 3 | Vi kan skære ned på antallet af features eller beslutte os for at alle arbejder ekstra (fx mødes om onsdagen eller arbejder via Slack i weekenden). |
| Vi mister kode eller rapport | 2 | 5 | Vi bruger GitHub i Visual Studio og Google Docs til rapportskrivningen. Vi bruger også Notion til at kunne dele/synkronisere vores idéer og forskellige diagrammer, tekststykker og modeller. Til slut bliver rapporten skrevet i Microsoft Office Word, hvor den bliver sikkerhedskopieret til Dropbox og GitHub. |

# VERSIONSSTYRING

Ved projektopstart oprettede vi et delt repository på GitHub som hele gruppen kunne arbejde på i løbet af forløbet. Vores rapport var i starten sat op i en Google Docs dokument, men hen mod aflevering benyttede vi os I stedet af Microsoft Word, og benyttede os af GitHub til versionsstyring og Dropbox til sikkerhedskopiering. Grunden til at vi skiftede fra Google Docs er at formatet gav os problemer i forhold til opsætning og automatisk rettelse af korrekt stavede ord.

Vi prøvede at gøre brug af ”commit early, commit often”, da versionsstyringen muliggør brug af kode fra tidligere versioner. Denne funktion blev faktisk hurtigt taget i brug, da vi havde svært ved at få vores menudesign til at fungere optimalt i starten. Vi havde dog nået at arbejde en del på dette repository, så vi oprettede et nyt, da det kunne være nyttigt at have den oprindeligt skrevne kode senere.

## Vision og Vision Statement for CC Byg.

Vi har udarbejdet Vision som et led i forretningsanalysen. Med Vision samt et Vision Statement har vi kunnet skære ind til benet *præcist* hvad CC Byg har haft brug for i den udviklede applikation.

**Vision introduktion:** vi udvikler en applikation til murerfirmaet CC Byg. Firmaet kunne godt tænke sig en mere overskuelig måde at føre inventar i forhold til lagerbeholdning. Dette vil gøre det nemmere at vide hvilke materialer, der eventuelt skal bestilles hjem til nye og igangværende byggesager, og vil også forenkle fakturering, da det vil overskueliggøre hvilke materialer, der er blevet brugt på hvilke byggesager.

**Problemstilling:** at lave en app der flydende og brugervenligt kan opfylde CCBYG’s behov for overblik over lager, projekter, medarbejdere samt allokering af ressourcer.

**Problem:** CC Byg køber mange materialer, og har brug for at vide præcist hvilke materialer, der bruges hos hvilke kunder samt hvad der er på lager.

**Påvirkning:** CC Byg samt kunder.

**Indflydelse**: Det er besværligt at sikre sig, at firmaet har de rigtige mængder materialer på lager, uden at måtte bestille nye hjem. Det er svært og tidskrævende at fakturere kunder, da det er en større proces at få overblik over hvad de brugte produkter har kostet, både hvis de er fra forhandler, men også fra eget lager.

**Løsning:** enapplikation, der muliggør reducering af tidsforbrug for hver fakturering. Derudover præcision af faktureringsprocessen.Forbedre registrering af hvor de forskellige produkter benyttes. Dette bidrager til en bedre lagerstyring med mindre spild, og mindre tidsforbrug for at opnå dette. Brugervenlighed skal være i fokus, da applikationen ikke er målrettet superbrugere.

Vision Statement:

“Nem og enkel styring af lager og projekter”

## Product position statement

### Introduktion

**For**: CC Byg.

**Hvem:** administrationen (bogholder), direktøren samt de enkelte murersvende.

Produktnavn: CCByg.

**Som**: effektiviserer fakturering, lagerstyring og projektstyring.

**I modsætning til**: e-conomic , udprint, kalender og firmaets ansattes hukommelse.

**Vores produkt**: Forskellen er at vores produkt er mere specialiseret i forhold til lagerstyring i modsætning til for eksempel e-conomic, der har fokus på bogføring. Derudover er det mere præcist end de teknikker, firmaet har benyttet til dato, samt applikationen gør det muligt for eksempel at hente gamle projekter for at se hvad materialeforbruget har været som et led i nye tilbud.

### Interessentbeskrivelse

**Firma:** ejer og administrationen – ejer samt tre ansatte (heraf én kontoransat).  
**Kunder**: en applikation, der vil kunne reducere tid brugt på fakturering og lagerstyring, vil komme kunderne til gode, da der vil blive frigivet tid til at bruge direkte på byggesager. Vil eventuelt også kunne give en prisbesparelse.

### Brugermiljø

**Fokus**: systemet skal være funktionsdygtigt og brugervenligt. Det skal være målrettet den almindelige bruger.

**Antal brugere**: fra 1 og op.

**Task cycle:** kort og præcis – formålet er lige netop at forkorte denne.

**Hensyn**: netværksforbindelse er påkrævet og adgang fra en almindelig computer.

**Platform:** firmaet har på nuværende tidspunkt ingen platform, og har heller ikke haft nogen tidligere.

**Andre applikationer:** e-conomic og mailsystem, ingen af delene skal der integreres med.

### Produktoversigt – krav og behov:

**Krav:** CRUD-opsætning (create, read, update, delete) for materialer og projekter, easy-to-use lagerføring. Applikationen skal bruges på PC.

**Behov:** allokering af materialer til projekter som kan tilbageføres eller afsluttes. Overblik over lager. Tidligere projekter skal gemmes og kunne vises i forhold til materialeforbrug.

## Kravspecifikation:

Det har været vigtigt for os at tage udgangspunkt i kundens krav, og først derefter selv opstille en prioritering ud fra vores egne funktionelle og ikke-funktionelle krav. Kundens måde at opstille sine krav kan selvfølgelig også føres tilbage til denne måde med funktionelle og ikke-funktionelle krav.

|  |  |  |
| --- | --- | --- |
| Behov | Prioritet | Funktioner |
| Præcis oversigt over lagerinventar | 4 | Oprettelse af en ny type vare, som systemet gemmer.  Ændring i antallet af en type vare (både tilføje og fratrække) |
| CRUD | 5 | Hovedessensen er at applikationen kan oprette (create), læse/indlæse (read), opdater (update) og slette de forskellige varer. |
| Enkel brugerflade | 4 | Skal kunne bruges af den almindelige, gennemsnitlige bruger. |
| Tidligere projekter og materialeforbrug skal vises. | 3 | Tidligere projekters materialeforbrug kan hjælpe med fremtidige tilbud. |
| Allokering af materialer til projekter. | 3 | Materialer skal fastlåses projekter, og først kunne fjernes eller frigives ved afslutning af projektet. Muligvis bekræftelse før ændring? |

Andre produktkrav:

* Applikationen skal være så brugervenlig som overhovedet muligt.
* Pålidelig og forholdsvis hurtig opstart.
* Windows styresystem påkrævet.

|  |  |  |
| --- | --- | --- |
| Krav | Prioritet | Noter |
| Brugervenlighed | 5 | Brugervenlighed er vigtigt, da vi ikke taler om superbrugere her - det er den gængse bruger. |
| Stabilitet | 3 | Stabilitet er nedprioriteret i forhold til brugervenlighed, mobilitet og pålidelighed, da det selvfølgelig er vigtigt at applikationen skal fungere, men hvis den lukker ned en sjælden gang i mellem, må den åbnes igen. Her har vi valgt, at det var vigtigere, at den var brugervenlig. |
| Mobilitet | 1 | Mobilitet er underordnet, da der ikke er tale om, at applikationen skal kunne bruges på mobiltelefoner, tablets eller lignende. |
| Pålidelighed | 4 | Pålideligheden viser noget om, at applikationen skal gøre, hvad brugeren beder den om. Hvis brugeren beder om at få gemt eller ændret noget, skal programmet gøre dette hver gang. |

# Forretningsanalyse

I vores forretningsanalyse har vi benyttet os af forskellige analysemodeller:

**SWOT**: Vi har udarbejdet SWOT for at skabe et overblik over kundens behov. SWOT har hjulpet os med at belyse de interne faktorer, nemlig firmaets *Strengths* og *Weaknesses* samt de eksterne faktorer: *Opportunities* og *Threaths*. Især svaghederne og mulighederne har vist, at en optimalt bygget databaseapplikation, der overskueliggør lager-, projekt- og medarbejderstyring, vil kunne bidrage til en øget indtjening i kraft af den besparede tid, der ellers ville være brugt på at danne sig et overblik over firmaets aktuelle situation. I SWOT-analysen kunne vi for eksempel se under *Weaknesses*, at det var en reel svaghed for firmaet et bruge så meget tid på fakturering og lagerstyring – dette tidsspild vil vi derfor søge at eliminere ved at effektivisere begge dele. Samtidig vil sådan en effektivisering frigive flere timer i firmaet, der eventuelt vil kunne veje op for manglen på murersvende og lærlinge i murerbranchen pt.

**Porters Five Forces**: Porters Five Forces har været endnu en brik i vores samlede forretningsanalyse. Den har bidraget til en yderligere specificering af de eksterne faktorer fra SWOT-analysen. Samt den fortæller os hvor stærk vores kunde er i branchen og hvordan de står i en konkurrence situation.

**Business Model Canvas:** Vores business model canvas har vi lavet under 1. iteration, som et led i opstartsfasen inden vi lavede vores forretningsplan. Den hjælper med at danne et overblik over hvorvidt det er et projekt, der er værd at arbejde videre med. Dette gør den ved at vi afdækker forskellige områder inden for hvad der er vigtigt for både kunder og firmaet selv.

## SWOT

Udarbejdet af: Gitte

|  |  |
| --- | --- |
| Strengths:  Lille firma med få ansatte – fleksibilitet og stor styring.  Godt omdømme / renommé - mindsker reklameudgifter  Større overblik over opgaverne fra ledelsen.  Ikke så mange udgifter til lønninger og for eksempel bogholderi.  Hurtigere kommunikation mellem ledelse og medarbejdere.  Hurtigere og nemmere vidensdeling mellem alle de ansatte.  Loyale medarbejdere grundet lille, tæt firma. | Weaknesses:  Mere udsat ved sygdom på grund af få medarbejdere.  Kan ikke byde ind på meget store opgaver.  Større udgifter ved indkøb af materialer, da ikke for mange monetære midler må være bundet op i lageret.  Må afvise projekter grundet manglende ledige ansatte.  Tidskrævende fakturering på grund af manglende overblik. |
| Opportunities:  Mulighed for at bibeholde alle medarbejdere og dermed viden ved recession.  Mulighed for at udvide med flere medarbejdere.  Mulighed for at tage freelancere ind ved enkelte store projekter.  Mulighed for at lave bedre tilbud da det er et mindre firma, der har færre udgifter.  Mulighed for flere opgaver grundet mindsket konkurrence. | Threats:  Større firmaer har flere medarbejdere at sætte på en opgave.  Større firmaer kan købe større ind til lager og dermed skære ned på materialeomkostninger.  Mangel på murersvende.  Mangel på lærlinge. |

## Porters Five Forces

Udarbejdet af: Emil.

Rivalisering i branchen:

Rivalisering i branchen er midt i mellem da der eksisterer mange murerfirmaer som oftest tilbyder nogenlunde det samme produkt. Truslen fra nye indtrængere er lav, da der er mangel på murere, og der ikke er den samme grad af konkurrence fra fx østeuropæiske arbejdsmænd. Der er en lav exit barriere. Branchen er lige nu i vækst og der er stor efterspørgsel på murere.

Truslen fra nye indtrængere:

Truslen fra nye indtrængere er høj da indtrængningsbarrieren er lav på grund af den nødvendige startkapital er lav og at produktdifferentieringen er lille. Det gør at der hurtigt kan komme andre ind på markedet. Stordriftsfordele inkluderer at man kan byde ind på store projekter samt at få mængderabat på materialer man køber ind.

Købernes forhandlingsstyrke:

Køberne er magtfulde da der findes mange murerfirmaer og at der stort set ingen produktdifferentiering eksisterer. Der er dog vækst i branchen og stor efterspørgsel på murere så det presser også købernes forhandlingsstyrke i negativ retning.

Leverandørernes forhandlingsstyrke:

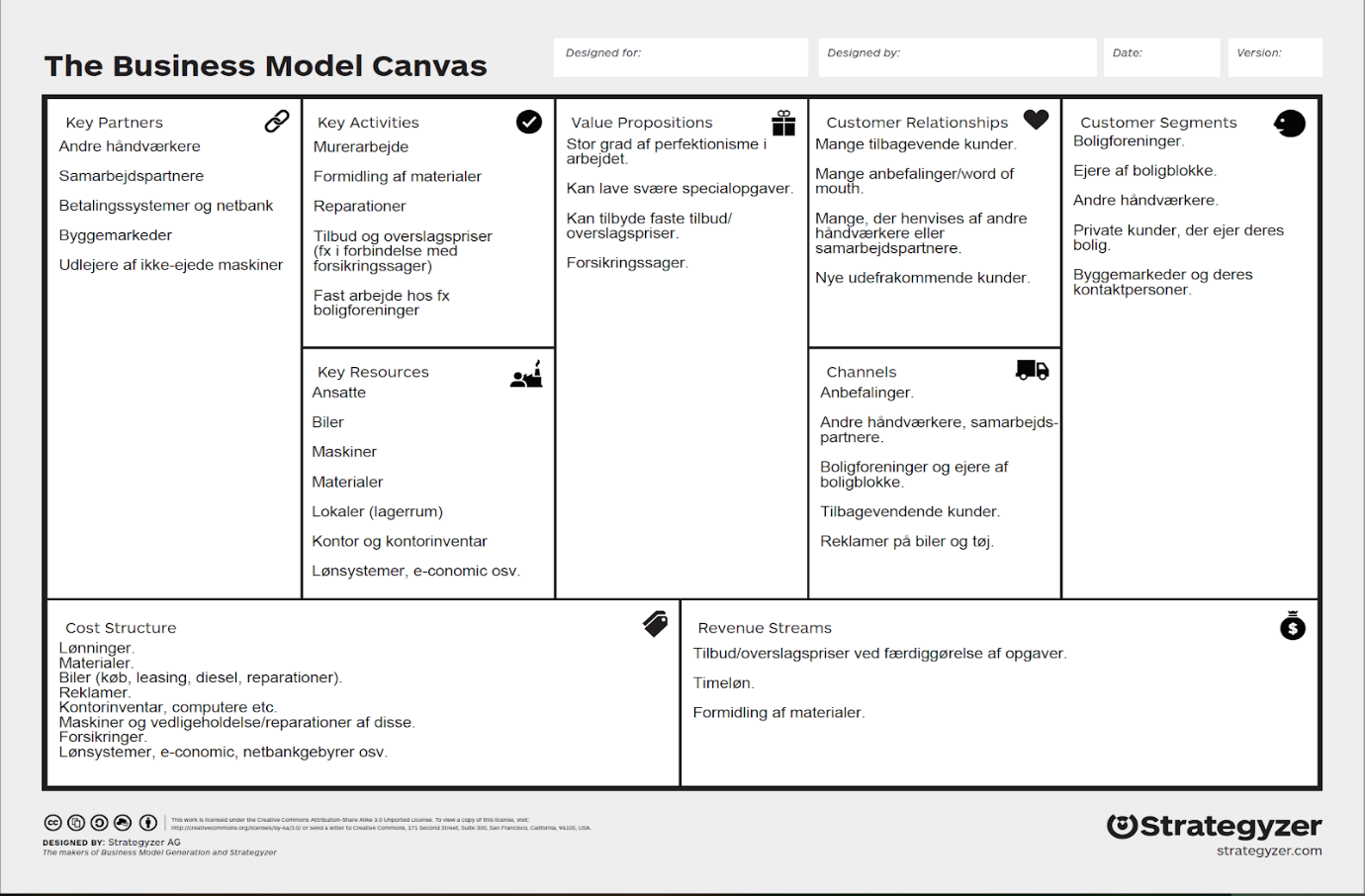
Leverandørerne er svage da der eksisterer mange leverandører, der kan producere nær identiske produkter. Samtidigt er det let at erstatte og ansætte nye murersvende da oplæringstiden er kort da der stort set ingen produktdifferentiering eksisterer.

Truslen fra substituerende produkter:

Truslen fra substituerende produkter er lav da det er svært at erstatte murerfirmaet med andet. Eksempler på substituerende produkter kunne være tømrerfaget og eller at man bruger færdiglavet betonelementer eller lignende.

Business Model Canvas.

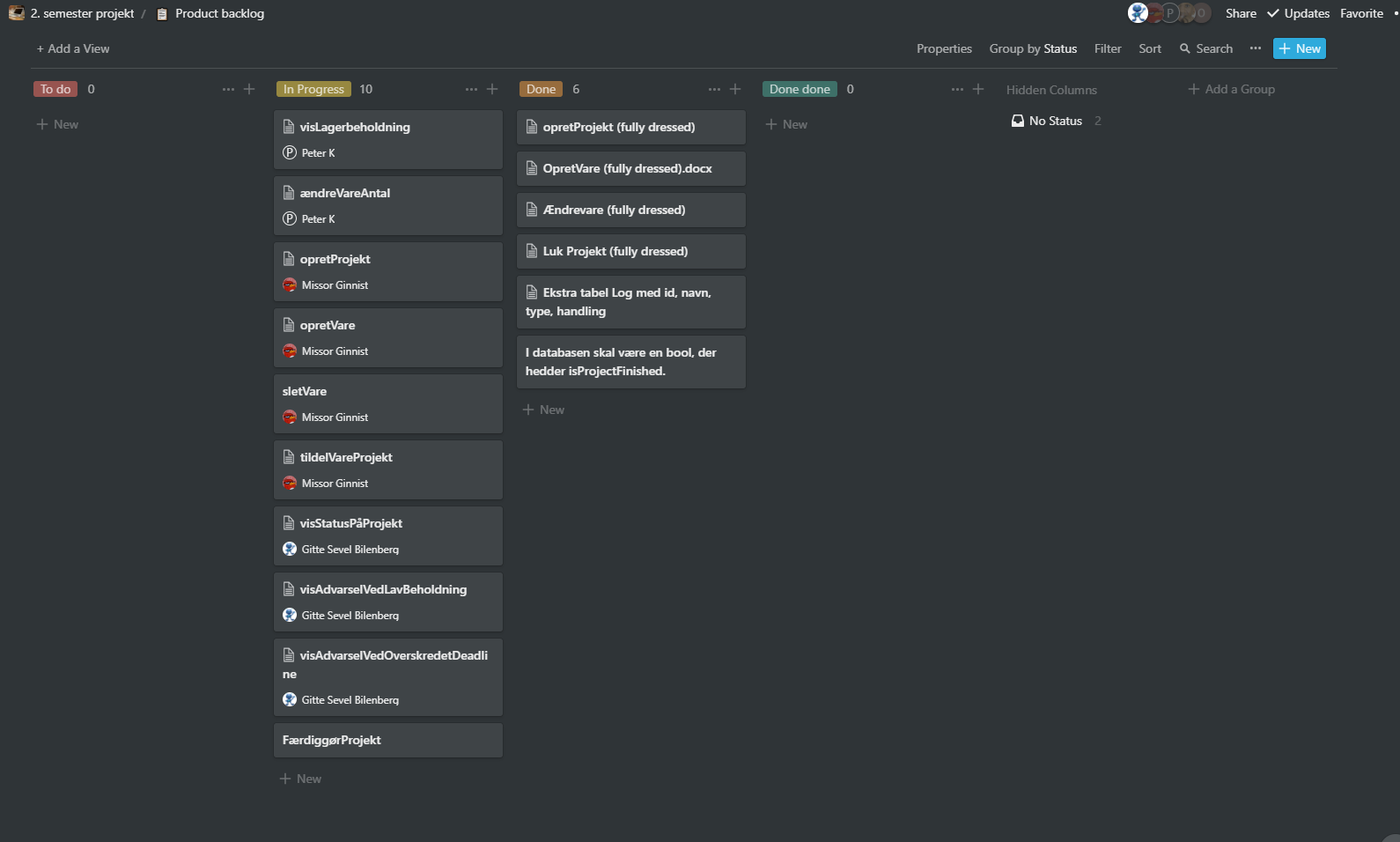
Udarbejdet af: Ofelia.



# Product Backlog - Alle:

*Udarbejdet af: alle*

Bruges til at teamet kan have overblik over hvad der skal laves, er i produktion og hvad der er færdiglavet



Use Case

|  |  |
| --- | --- |
| Use Case Name | OpretNyVare |
| Scope | Opretter ny vare i databasen og startbeholdning |
| Level/Niveau | Hovedmål |
| Primary Actor | Admin |
| Stakeholders & Interests | Admin & medarbejdere |
| Preconditions | Logget på med admin rettigheder |
| Success Guarantee | Nyt produkt, antal og minimumsbeholdning tilføjet til databasen |
| Main Success Scenario | 1.      Åbn ”Opret ny vare” menu  2.      Skriv navn på produktet  3.      Vælg type/kategori og evt. underkategorier  4.      Kort beskrivelse af produktet  5.      Vælg antal af nyt vareprodukt  6.      Sæt minimumbeholdning for advarsel  7.      Kontroller korrekt data  8.      Gem indtastninger |
| Extensions | 1. Åbn ”Opret ny vare” menu  2. Skriv navn på produktet  a) Skriver forkert navn    3. Vælg type/kategori og evt. underkategorier  a) Vælger forkerte kategorier    4. Kort beskrivelse af produktet  5. Vælg antal af nyt vareprodukt  6. Sæt minimumbeholdning for advarsel  a) Sætter ikke minimumsbeholdning – giver fejlmeddelelse om mangel    7. Kontroller korrekt data  8. Gem indtastninger  a) Gemmer ikke data |
| Special Requirements | Menuer til at vælge kategorier, indtaste info i |
| Technology and Data Variations List | PC |
| Frequency of Occurrence | Ofte i begyndelsen, senere kun sjældent |
| Miscellaneous | Hvad er relevant som ikke ellers nævnes i andre punkter. Bl.a. åbne problemer der skal løses. |

Use Case Gitte

|  |  |
| --- | --- |
| Use Case Name | opretProjekt |
| Scope | Opretter et byggeprojekt bundet til en klient hvor man kan uddele materialer fra lagerbeholdninger |
| Level/Niveau | Hovedmål |
| Primary Actor | Admin/bruger |
| Stakeholders & Interests | Brugere |
| Preconditions | Materialer oprettet I systemet. Mængde af materialer stor nok til projekt. |
| Success Guarantee | Materialer tildelt til projekt. Projekt givet en slutdato. |
| Main Success Scenario | 1.   Navngiv projekt  2.   Angiv adresse for projekt  3.      Tildel korrekt mængde af materialer til projekt, så de ikke kan tildeles andre projekter  4.      Angiv dato for projekts afslutning |
| Extensions | 1.   Navngiv projekt  2.   Angiv adresse for projekt  a)      Projekt allerede i gang på addresse  3.      Tildel korrekt mængde af materialer til projekt, så de ikke kan tildeles andre projekter  a)      Manglende mængde af produkter  4.      Angiv dato for projekts afslutning |
| Special Requirements | Related non-functional requirements. |
| Technology and Data Variations List | PC + mobil |
| Frequency of Occurrence | Varierende |
| Miscellaneous | Such as open issues. |

Use Case Gitte

|  |  |
| --- | --- |
| Use Case Name | ÆndreVare |
| Scope | Tilføje / eller slette en  vare i databasen og beholdning |
| Level/Niveau | Hovedmål |
| Primary Actor | Admin |
| Stakeholders & Interests | Admin & medarbejdere |
| Preconditions | Logget på med admin rettigheder |
| Success Guarantee | Sletning af produkt,  ændring af antal i databasen |
| Main Success Scenario | 1.      Åbn ”Ændre vare” menu  2.      Skriv navn på produktet  3.      Vælg type/kategori og evt. underkategorier  4.      Vælg antal af et vareprodukt  5.      Vælg tilføj/slet  6.      Kontroller korrekt data  7.      Gem indtastninger |
| Extensions | 1. Åbn ”Ændre vare” menu  2. Skriv navn på produktet  a) Skriver forkert navn    3. Vælg type/kategori og evt. underkategorier  a) Vælger forkerte kategorier    4. Kort beskrivelse af produktet  5. Vælg antal af nyt vareprodukt  6. Sæt minimumbeholdning for advarsel  a) Sætter ikke minimumsbeholdning – giver fejlmeddelelse om mangel    7. Kontroller korrekt data  8. Gem indtastninger  a) Gemmer ikke data |
| Special Requirements | Menuer til at vælge kategorier, indtaste info i |
| Technology and Data Variations List | PC, mobil |
| Frequency of Occurrence | Ofte |
| Miscellaneous | Hvad er relevant som ikke ellers nævnes i andre punkter. Bl.a. åbne problemer der skal løses. |

# Prototype

*Udarbejdet af: Gitte*

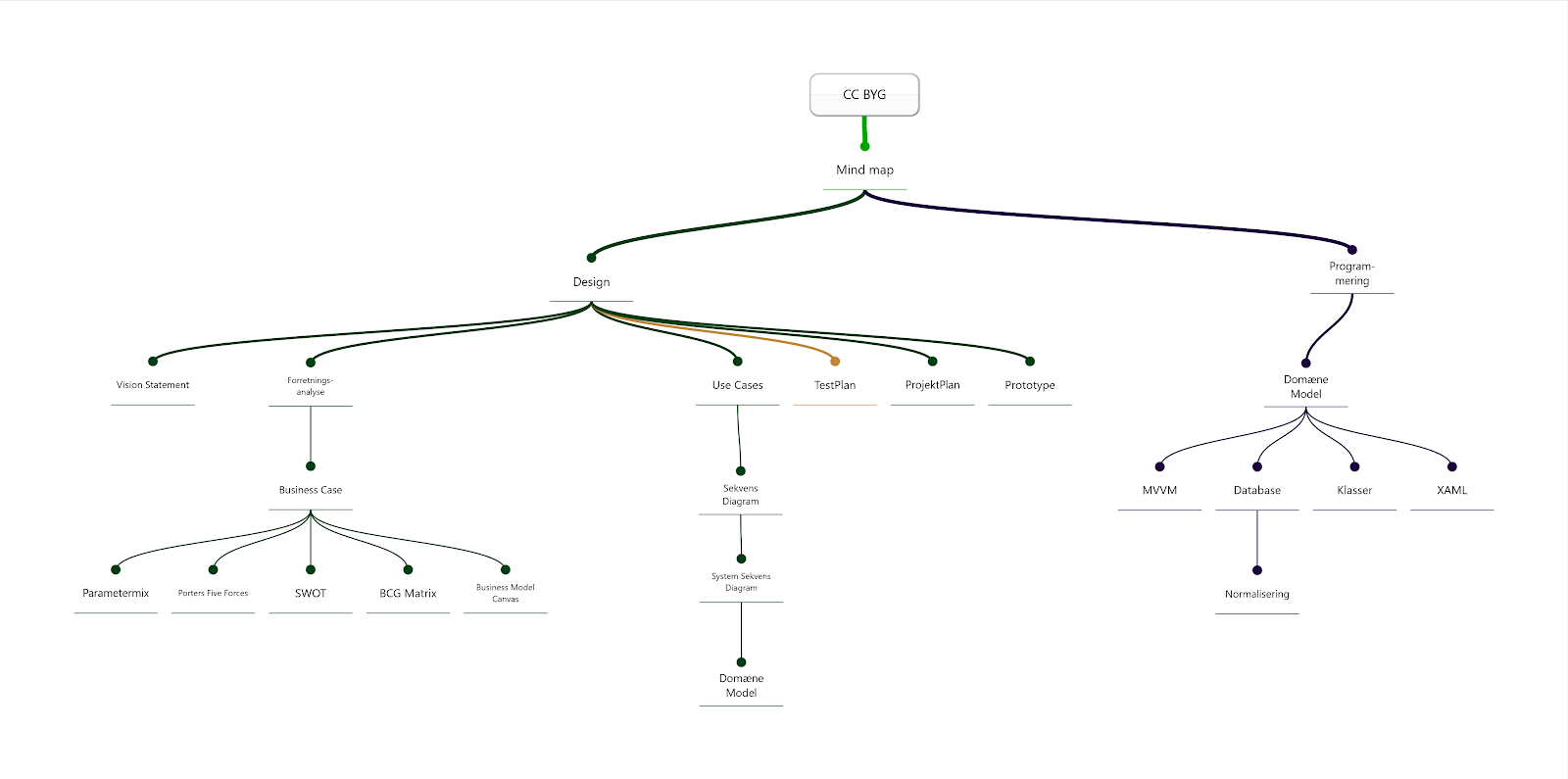
Vi lavede denne prototype som et forslag til hvordan vores app kunne se ud samt at have en ide om hvilket funktioner vi ville have i vores app.

<https://marvelapp.com/653f865>

Vi valgte efterfølgende at “smide” vores prototype væk, og blev enig om et andet GUI samt en anden opsætning af vores funktioner.  
Dette blev gjort for at få en større funktionalitet ind vores app og en større overskuelighed for brugeren.

## Mindmap

*Udarbejdet af: Gitte*



## Projektplan

*Udarbejdet af: Ofelia*

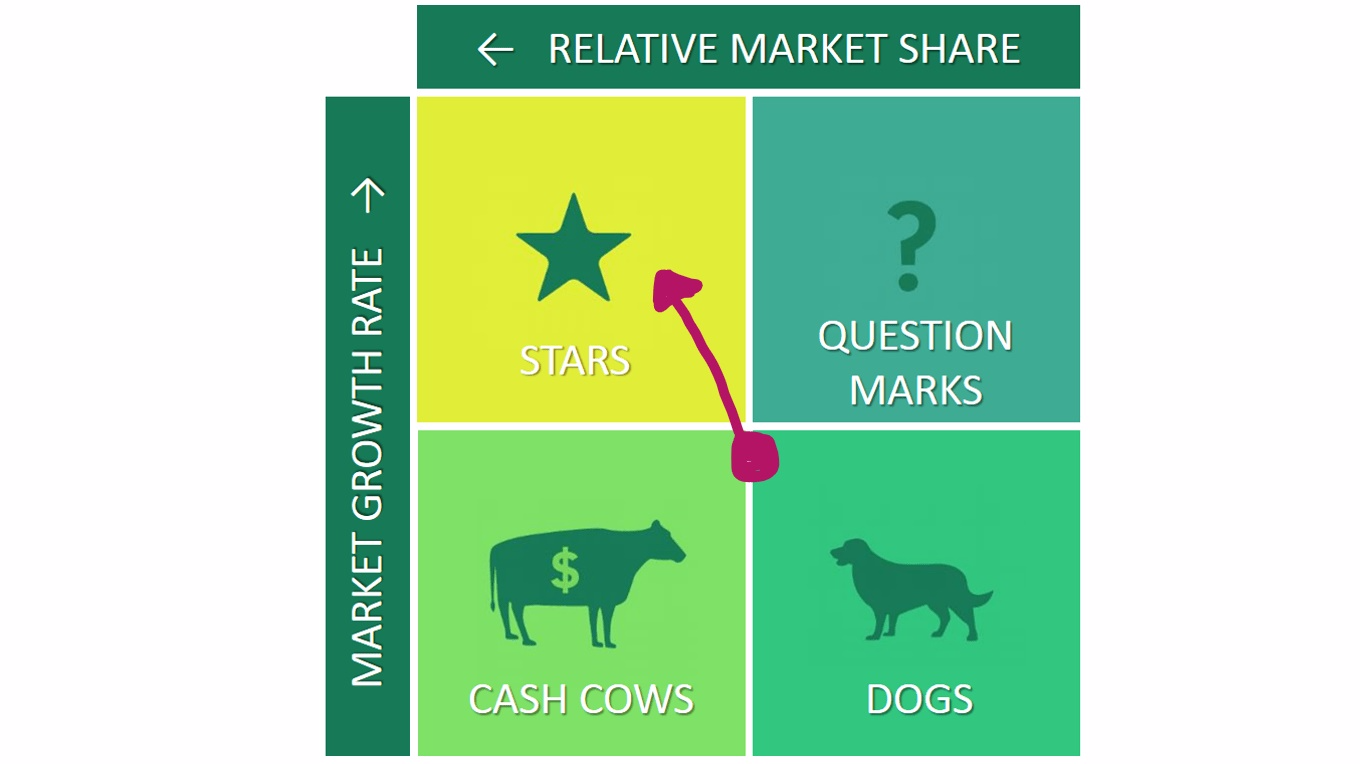
|  |  |
| --- | --- |
| Projektets titel | CCByg |
| Initiativtager | EASJ, herunder Per Laursen og Vibeke Sandau. |
| Baggrund | 2. semesterprojekt, datamatiker. Vi har valgt at udarbejde en app til CC Byg, da firmaet har brug for en nem måde at lagerføre varer og administrere projekter og medarbejdere på – dette lever fint op til den stillede opgave, da det var et krav, at der skulle være et databasesystem indbygget i vores aflevering. |
| Formål | Vi skal benytte os af den viden, som vi har fået på 1. og 2. semester i praksis. Vi skal demonstrere at vi formår at benytte både construction og design, og få disse to til at arbejde sammen i udviklingen af ét projekt. |
| Leverancer | Vi afleverer en applikation (construction-del) og en rapport (design), der viser, hvordan vi har benyttet design i constructionarbejdet, og omvendt. |
| Succeskriterier | At vi ender med en færdigudviklet applikation, der fungerer efter hensigten.  At vi alle får brugt de kompetencer vi har lært gennem de sidste to semestre og alle deltager i både kodnings- og designarbejdet.  At vi får afleveret en rapport, der viser, at vi kan benytte os af de forskellige designprocesser, modeller og diagrammer. Både rapporten og applikationen skal møde de kriterier, der er stillet fra undervisernes side. |
| Ressourcer | Vi skal gøre brug af vores kompetencer, og bruge den givne tid så optimalt som muligt.  At vi får gjort brug af vores underviseres og medstuderendes viden. Vi har en afsat tidsperiode til at lave projektet. Vi har forskellige materialer, noter og bøger fra 1. og 2. semester. |
| Tidsplan | Første iteration: afsluttedes, da vi afleverede 1. delaflevering i slut oktober.  Vi regner med at vores iterationer varer knap 2 uger hver.  Vi vil gerne være så godt som færdige med projektet d. 14. december kl. 14, så vi derefter har god tid til fælles gennemgang og redigering, ved behov. |
| Organisering | Emil Hammer  Ofelia Symes  Gitte Bilenberg  Daniel Møller |
| Interessenter | EASJ, herunder Per Laursen og Vibeke Sandau. |
| Overlevering til drift | Projektet skal afleveres i Wiseflow. |
| Risikoanalyse | Vi regner med at blive færdige. Ved behov for ekstra features, som vi ikke oprindeligt har regnet med, vil vi arbejde på dette til sidst, og dermed minimere risikoen for at vores samlede projekt ikke bliver færdig. Vi har også afsat fem dage til sidst til fejlfinding og redigering. Vi benytter os af github og google docs, og der bør derfor, forhåbentligt, ikke være risiko for at miste projektet pga. computernedbrud eller lignende. |

## Business Case

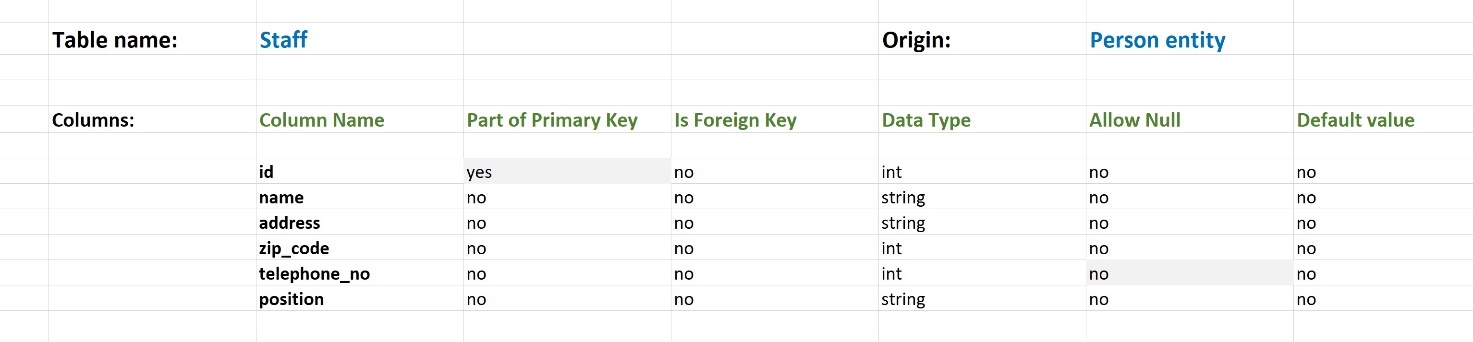
*Udarbejdet af: Ofelia*

|  |  |  |  |
| --- | --- | --- | --- |
| Business Case | | | |
| TITEL PÅ PROJEKTET | CC Byg | | |
| TEASER TIL PROJEKTET  (Max 160 tegn) | En overskuelig måde at holde styr på varelager, proj ekter og medarbejdere. | | |
| PROJEKTRESUMÉ | CC Byg er en app, hvor de primære funktioner er inventarstyring og oversigt over projekter og medarbejdere. Underfunktioner vil være at kunne tilføje, slette og ændre antal af varer, at kunne se oversigt over medarbejdere og tildele dem på forskellige projekter samt at kunne se hvilke projekter der er i gang, opdatere og lukke disse og se hvilke materialer, der er brugt på hvilke projekter. | | |
| BAGGRUND OG FORMÅL MED PROJEKTET | Formålet med dette projekt er at teste de værktøjer og den viden, som vi har tilegnet os de sidste to semestre. Vi har gjort brug af de forskellige modeller og diagrammer vi har lært i design, herunder business case, use case, domænemodeller, systemsekvens-, sekvens- og klassediagrammer og vores viden om test og vision. Vi har også arbejdet ud fra den arbejdsmetode vi har lært i form af forskellige iterationer/sprints. I programmeringsdelen har vi haft mulighed for at repetere og udvikle vores evner indenfor C#, XAML, SQL og LINQ. CC Byg projektet kommer derfor rigtig fint omkring alle disse emner. | | |
| PROJEKTPERIODE FRA OG TIL | Fra d. 12. oktober (hvor vi afleverede 1. projektetablering) til d. 19. december 2018. | | |
| AKTIVITETER OG HANDLING | Første iteration: vi startede med designdelen og at udarbejde flere use cases, herunder et par fully dressed. Ud fra disse use cases har vi lavet vores forretningsanalyse, herunder sekvensdiagrammer, domænemodeller. Derudover har vi lavet porters five forces, SWOT, vision statement, business model canvas, BCG Matrix, Parametermix og en projektetablering. Ved afslutningen af første iteration afleverede vi vores første delaflevering, der indeholdt vores projektetablering, nogle få use cases, en domænemodel og vores risikoanalyse..  Anden iteration:  Derefter har vi kodet simultant med, at vi har arbejdet med de forskellige designdiagrammer og modeller - på denne måde har vi kunnet koordinere og arbejde rundt omkring nogle af de udfordringer, vi har mødt. | | |
| FASER (MILEPÆLE) I PROJEKTET | Vi satser på at gennemgå tre iterationer. | | |
| PROJEKTDELTAGERE | Internt | Eksternt | Andre Interessenter |
| Ofelia  Emil  Peter  Gitte  Daniel |  |  |
| KOMMUNIKATION OG FORMIDLING | Vi har haft nogle faste aftaler ift. hvornår vi skulle mødes for at arbejde og hvordan vi har arbejdet, når ikke vi har mødtes fysisk. Vi har også aftalt på forhånd hvilke ting vi skulle arbejde med hver især til næste gang eller ved sygdom. | | |
| FORVENTEDE RESULTATER / SLUTPRODUKTER | Vi forventer at aflevere et færdigudviklet produkt senest d. 19. december 2018. Vi forventer at vores applikation lever op til både de krav, der er stillet til 2. semesteropgaven, men også de krav/funktioner, som vi selv har sat op. Når disse funktioner er kodet, kan vi altid vurdere, om vi vil tilføje yderligere iterationer og funktioner, da vi så ved at applikationen fungerer for hver afsluttet iteration. Vi forventer ligeledes en fuldt gennemarbejdet rapport, der viser, at vi kan benytte os af de forskellige arbejdsprocesser, diagrammer og modeller. | | |
| FORVENTET EFFEKT  Herunder:  PLAN FOR EVALUERING AF UDBYTTE / EFFEKT | Vi forventer, at den færdige applikation vil gøre det mere overskueligt at føre lager, holde opsyn med projekter samt kunne se informationer om de forskellige medarbejdere. Vi vil på denne måde eliminere en langsommelig lagerføring, der til dags dato har foregået via kalender, løse papirer og besigtigelse af det fysiske lager. På samme måde bliver det også nemmere at fakturere kunder og lave nye tilbud, da projekter – både nye og gamle – vil være gemt i applikationen. | | |
| RISICI | At vi ikke bliver færdige til tiden.     |  |  |  |  | | --- | --- | --- | --- | | Konsekvens    Sandsynlighed | Lille | Mellem | Høj | | Meget usandsynligt |  |  | X | | Sandsynligt |  |  |  | | Meget sandsynligt |  |  |  | | | |
| FINANSIERING | Ikke relevant. | | |

BCG Matrix -Ofelia



Anden iteration



# TESTPLAN

## 1.0. Indledning.

Meningen med denne testplan er at gøre klart hvad der skal testes og hvornår.

## 1.1. Formål.

Formålet med denne testplan er at kommunikere testplanen til projektets medlemmer og gøre det klart for alle, hvad der skal testes, og hvornår.

HOLDMEDLEMMER

|  |  |
| --- | --- |
| Navn | Rolle |
| Gitte Bilenberg |  |
| Ofelia Symes |  |
| Daniel Møller |  |
| Emil Hammer |  |

## 2.0. SCOPE

Ved afslutningen af første fase, skal testeren være i stand til at gøre følgende:

1. Lave en manuel test med så mange skridt som muligt.

2. Gemme det.

3. Hente det og bruge det mens man kører en test.

4. Videregive resultaterne med kommentarer.

Det antages at brugerne kan åbne programmet, så dette er ikke en del af testen.

## 3.0. Antagelser/Risiko

### 3.1. Antagelser

1.       Man kan oprette en vare

2.       Man kan rette i antallet af varer

3.       Man kan tildele en vare til et projekt

4.       Man kan ændre i et projekt

5.       Man kan lukke et projekt

6.       Man kan oprette en medarbejder

7.       Man kan ændre i en medarbejders oplysninger

8.       Man kan tildele en enkelt medarbejder til et projekt.

### 3.2. Risiko

Vi har fundet følgende risici

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Risiko | Påvirkning | Udløser | Plan for afhjælpning |
| 1 | Jo bedre brugeren kender programmet, jo flere funktioner ønskes | Mellem | Ændring af program | Opdatering af program til version 2.0 |
| 2 | At andre end primære brugere kan gå ind i programmet og ændre oplysninger | Høj | Ændring af program | Tilføjelse af log in med password |
| 3 | Der kan opstå fejl i forbindelsen mellem program og database, hvis der mistes forbindelse til internettet. | Høj |  |  |
| 4 |  |  |  |  |

## 4.0. Tilgang til projektet.

Vores tilgang til projektet har været agil scrum. Vi har prøvet at gøre de kritiske dele af projektet færdige først, teste dem samt dokumentere processen.

Derefter har vi diskuteret vores resultater og taget dette med til næste iteration, hvor vi har udbygget vores projekt og kode, og gentaget processen med testning, dokumentation og diskussion.

Dette har vi planlagt at gøre i 3 iterationer.

## 5.0. Test miljø

Vores App er blevet testet i Visual Studio, da alle gruppemedlemmer har dette program og har kendskab til brugen.

## 6.0. Milesten / Aflevering

### 6.1.  Test tidsplan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Opgave | Start | Slut | Indsats | Kommentar |
| Mindmap |  |  |  |  |
| SWOT |  |  |  |  |
| Parametermix |  |  |  |  |
| BCMG |  |  |  |  |
| Porters five forces |  |  |  |  |
| Use Cases |  |  |  |  |
| Sekvens diagrammer |  |  |  |  |
| System sekvens diagrammer |  |  |  |  |
| Domænemodel |  |  |  |  |
| Start af kodning |  |  |  |  |
| Test af kodning |  |  |  |  |
| Business analyse |  |  |  |  |
| Forretningsanalyse |  |  |  |  |
| Rettelse af kode |  |  |  |  |
| Udbygning af kode |  |  |  |  |
| Test af kode |  |  |  |  |
| Udbygning og afslutning af kode |  |  |  |  |
| Test af kode |  |  |  |  |
| Gruppens godkendelse af kode/app |  |  |  |  |
| Samling af rapport |  |  |  |  |
| Afslutning af rapport |  |  |  |  |
| Aflevering af rapport med app i Wiseflow |  |  |  |  |

### 6.2. Aflevering

|  |  |
| --- | --- |
| Det der skal laves | Laves af |
| Rapport | Emil Hammer  Daniel Møller  Ofelia Symes  Gitte Bilenberg |
| Programmering | Emil Hammer  Daniel Møller  Ofelia Symes  Gitte Bilenberg |

# Versionsstyring

Vi startede en delt repository som hele gruppen kunne arbejde på i starten af forløbet, hvor vi smed projektet op på for hurtig samling af opgaverne. Vores rapport var i starten sat op i en Google Docs dokument, men var senere smidt over i et .docx dokument og sat op på samme repository som opgaven på grund af tekniske besværligheder på Google platformen.

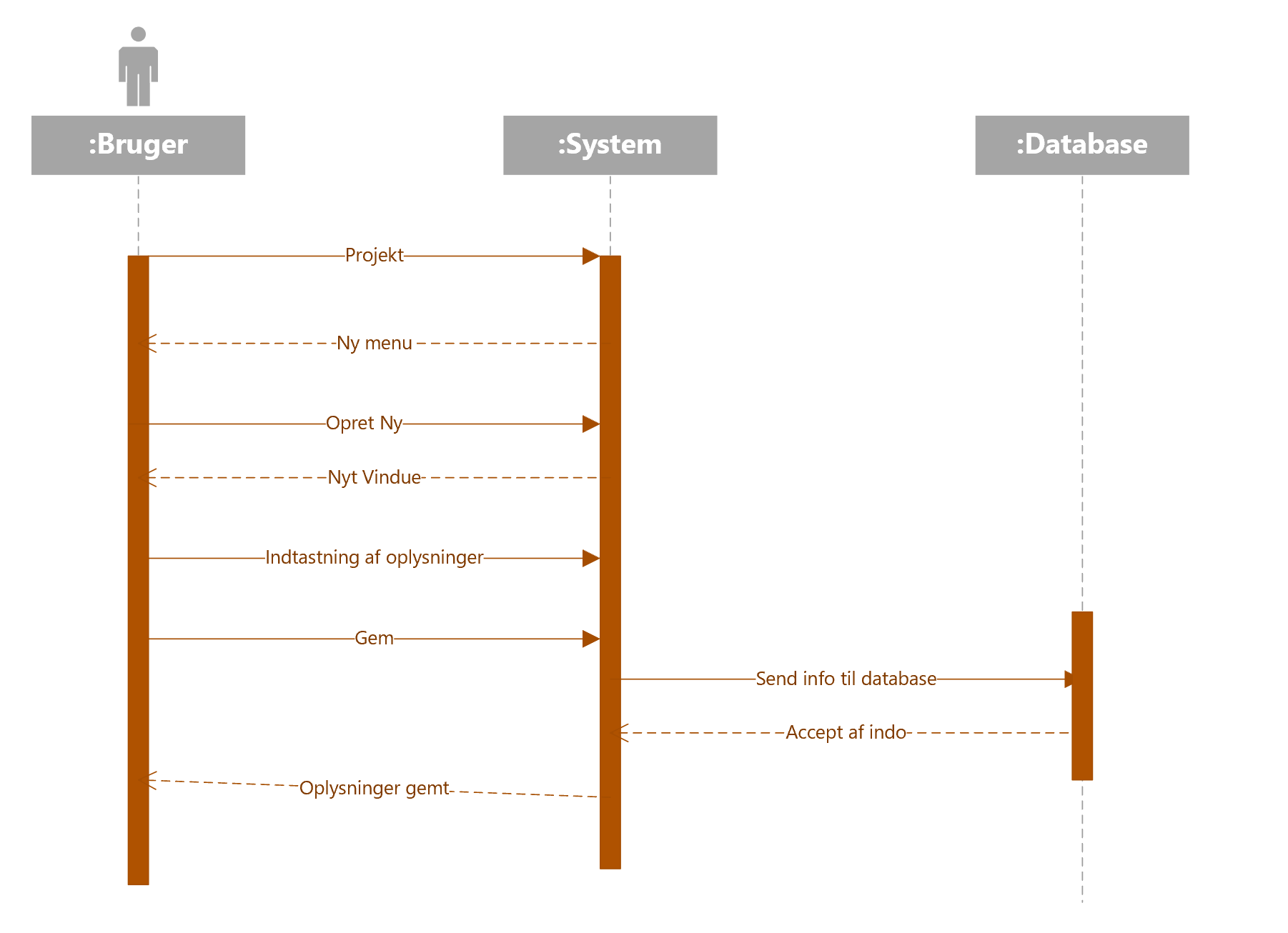
Daniel gjorde brug af princippet med "commit early, commit often" grundet muligheden for at hoppe tilbage i versioner for at hente ting og linjer af kode som kan fejle senere for at bringe det tilbage til et brugbart stadie. Denne funktion blev faktisk taget i brug, da vi havde en mindre katastrofal fejl i vores menudesign. Dog kunne Emil ikke lide at slette og overskrive filer, så vi blev nødt til at skabe en ny repository.

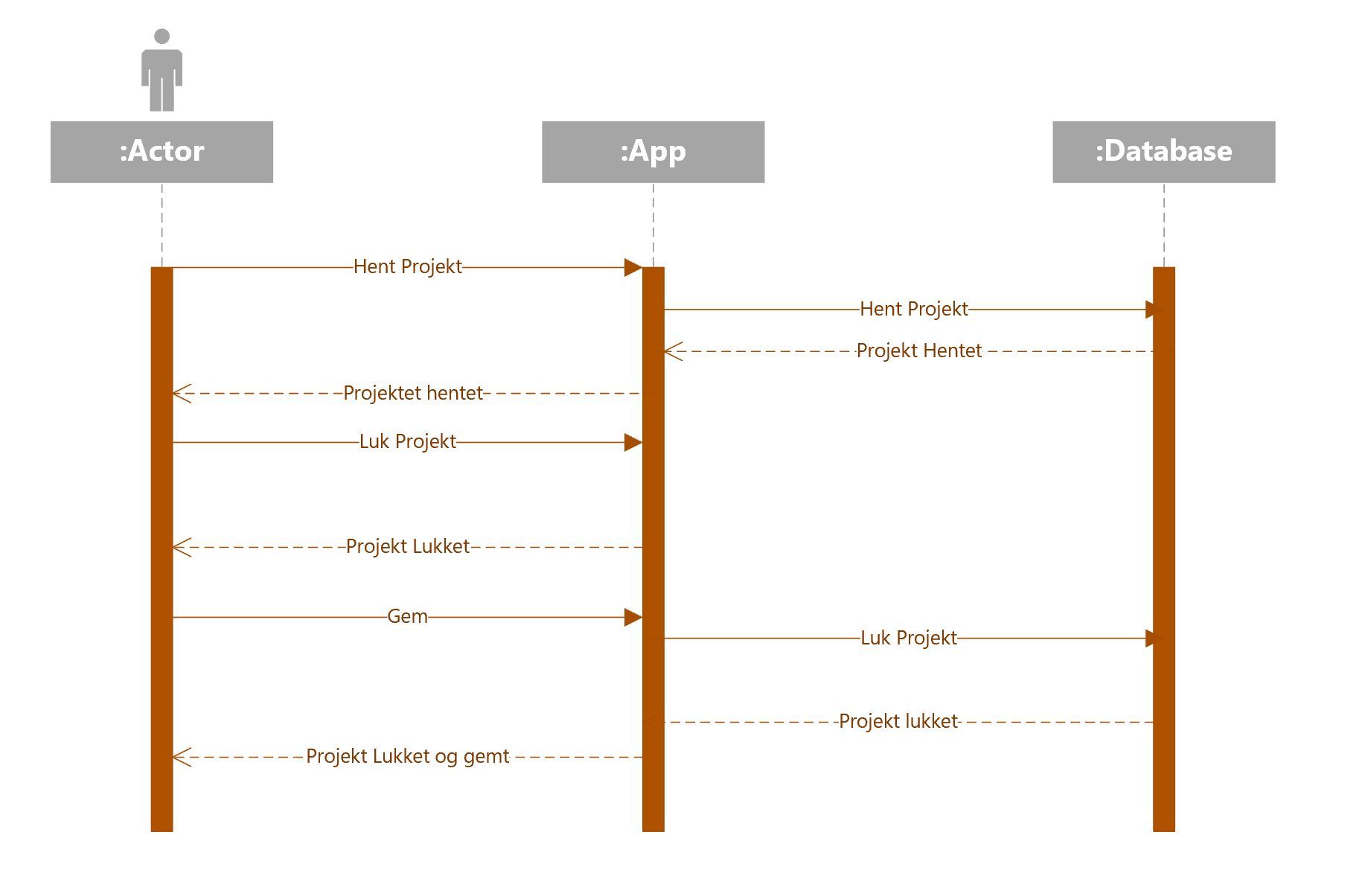
Desværre er Emil noget af en perfektionist, som er lidt bange for at uploade noget som måske ikke virker fuldstændigt, hvilket har givet problemer da f.eks. menuen virkede, men vores CRUD ikke gjorde. Dette ville ikke have været et problem da GUI sagtens kan udvikles uden en fuldstændigt virkende CRUD, men udviklingen af vores GUI blev som konsekvens en smule forsinket.

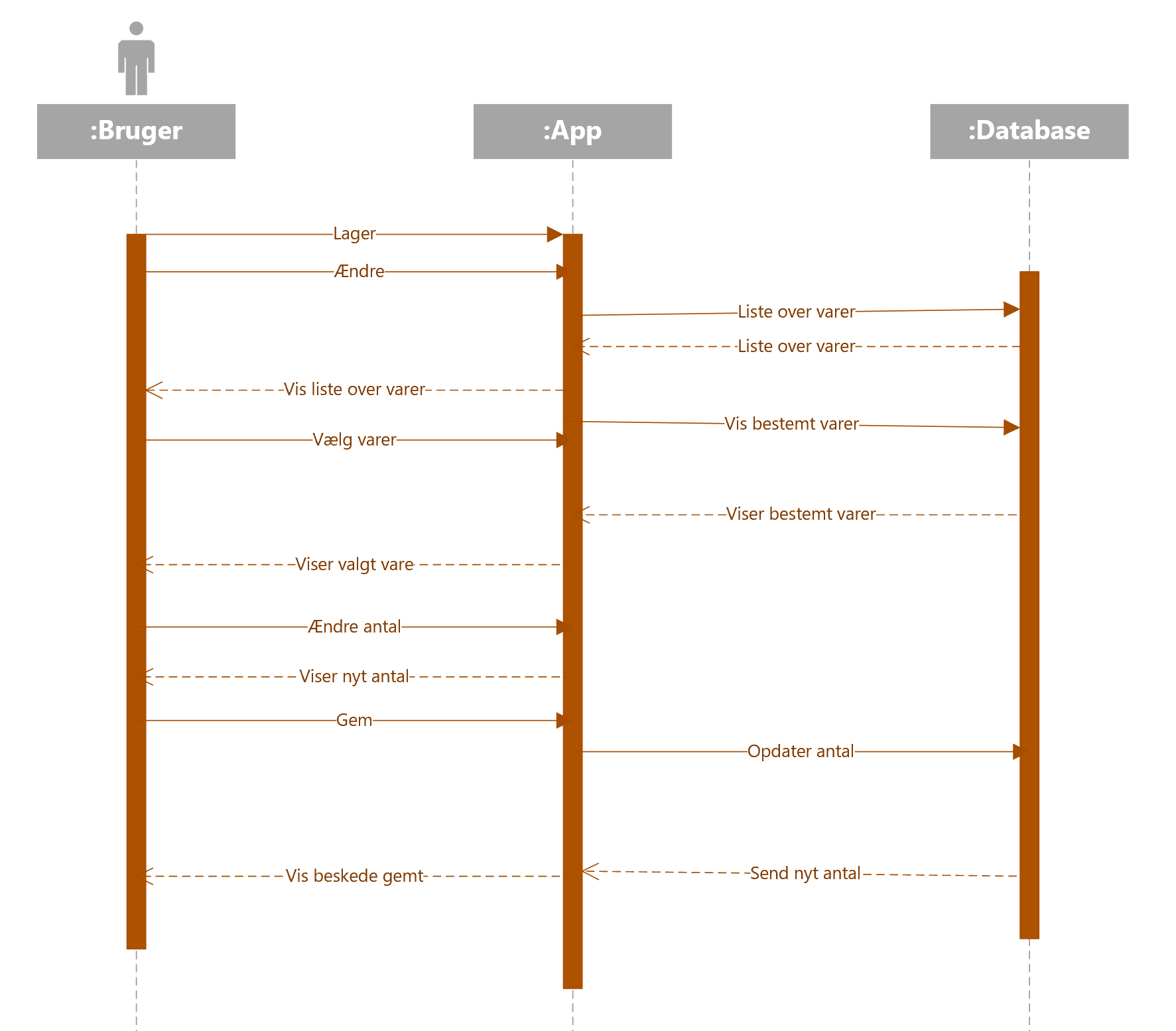
Branching bliver ikke brugt så flittigt som det burde. Daniel lavede en branch for at han kunne arbejde med projektet uden at skabe for store forstyrrelser i master-branchen men det endte op med at al udvikling blev lavet i den i stedet.

Alt i alt fandt Daniel det meget hjælpsomt at bruge versionsstyring, dog kræver det at alle er med på alle aspekter af legen, før det hjælper rigtigt meget. Hvilket også var det problem, som vi baksede med. Emil fnøs flere gange af idéen med at lave en ny branch til masteren, fordi han ikke så pointen i det. Mange og tidlige uploads af vores opgave kunne have hjulpet os et par gange i forløbet, men det kræver at alle der arbejder på projektet gør det samme. Ellers er der enkelte der bare skovler store opdateringer på de andres små commits, hvilket skaber mange konflikter.

# Sekvensdiagrammer







# Kode

# Svar på underspørgsmål

OPSUMMERER VORES SVAR FRA RAPPORTEN

Vores underspørgsmål er:

## Underspørgsmål 1: Hvordan kan virksomheden få synliggjort fx manglende materialer?

En af de vigtigste funktioner i programmet er lige netop at få synliggjort mængden af forskellige materialer og at kunne ændre i dette. DER SKAL MERE TEKST HER.

## Underspørgsmål 2: Hvordan kan virksomheden nemmere håndtere igangværende byggeprojekter og allokere materialer til disse?

Igangværende byggeprojekter vil stå som åbne byggesager, så længe de ikke bliver aktivt lukket (ER DETTE STADIG EN FUNKTION I PROGRAMMET?). På denne måde vil virksomheden kontinuerligt være opmærksomme på hvilke byggesager, der stadig er markeret som igangværende, og hvilke der er lukkede. På samme måde kan materialer tildeles til de forskellige byggesager, og der skabes dermed et overblik – både over materialeforbrug til dato på igangværende byggesager, men også

## Underspørgsmål 3: Hvordan vil en elektronisk løsning forbedre CC Bygs drift?

Som vi har beskrevet igennem rapporten, har CC Byg indtil videre haft et system, der bestod af kalender og udprint. På denne måde vil en elektronisk løsning højst sandsynligt forbedre driften betydeligt, især når det kommer til lager- og projektstyring. MERE TEKST TAK

## Underspørgsmål 4: Hvordan kan man udvikle et lagerbeholdningssystem i C# med en database som hjælper virksomheden?

Databasen i CCByg-applikationen holder styr på både de forskellige varer (og mængder af disse), projekter samt medarbejdere. Virksomheden vil derfor være i stand til UMIDDELBART MED DET SAMME at kunne danne sig et overblik over antallet af varer på lager, projekter – og hvilke varer, der benyttes på forskellige projekter – samt hente oplysninger om medarbejdere frem. Virksomheden har både mulighed for at skabe, læse, opdatere og slette informationer i alle tre kategorier og får dermed selv fuld styring over applikationen.”Hvordan kan Unified Process (UP), C#-programmering og relationsdatabaser blive brugt til udvikling og implementering af et mindre IT-system?”

# Problemformulering – svar og konklusion.

”Hvordan kan Unified Process (UP), C#-programmering og relationsdatabaser blive brugt til udvikling og implementering af et mindre IT-system?”

# Litteraturliste

Craig Larman - Applying UML and Patterns\_ An Introduction to Object-oriented Analysis and Design and Iterative Development (2005, Prentice Hall PTR)

Ian Sommerville - Software Engineering, 9th Edition   (2011, Addison-Wesley)

Thomas M Connolly, Carolyn E Begg - Database Solutions\_ A step by step guide to building databases (2nd Edition)   (2003, Addison Wesley)

Ian Sommerville - Software Engineering, 9th Edition   (2011, Addison-Wesley)

<https://fronter.com/easj/links/files.phtml/5c0e375217dbd.1621230229$690303253$/Arkiv/Softwaredesign/Litteratur/FURPS.docx>

<https://fronter.com/easj/links/files.phtml/1621230229$690303253$/Arkiv/Softwaredesign/Litteratur/ScrumIn5Minutes.pdf>

<https://fronter.com/easj/links/files.phtml/5c0e375217dbd.1621230229$690303253$/Arkiv/Softwaredesign/Litteratur/Scientific+methology.docx>

## Ressourceliste

GitHub, hvor vores projekt ligger på nedenstående link:  
<https://github.com/XxXStopHammerTimeXxX/CCB>

Microsoft Azure

Marvel.com

Google Docs

# 

# Bilag

