|  |  |
| --- | --- |
|  | **DM71** |
|  | Gruppe 4  Daniel – Erik – Kasper - Jacob |

|  |
| --- |
| **[Vestbjerg byggecenter]** |
|  |

Indholdsfortegnelse

[Forord 3](#_Toc263333729)

[Indledning 3](#_Toc263333730)

[Kodestandard 3](#_Toc263333731)

[It-forundersøgelse 4](#_Toc263333732)

[BATOFF 4](#_Toc263333733)

[Virksomhedskarakteristik 4](#_Toc263333734)

[Organisationskultur 5](#_Toc263333735)

[Regnskabsanalyse 7](#_Toc263333736)

[Problemer, hypoteser og løsningsmuligheder 10](#_Toc263333737)

[SWOT Analyse 10](#_Toc263333738)

[Valg af strategi 10](#_Toc263333739)

[Ansoff’s vækstmatrice 11](#_Toc263333740)

[IT strategi 12](#_Toc263333741)

[Applikationer og information 13](#_Toc263333742)

[Teknologi 13](#_Toc263333743)

[Organiseringen af IT-funktionerne 13](#_Toc263333744)

[It-handlingsplan 13](#_Toc263333745)

[Implementering: 15](#_Toc263333746)

[Brugerdeltagelse: 16](#_Toc263333747)

[Iteration 2 17](#_Toc263333748)

[Aktivitetsdiagram 17](#_Toc263333749)

[Use-cases 18](#_Toc263333750)

[Domænemodel 21](#_Toc263333751)

[Systemsekvensdiagrammer 26](#_Toc263333752)

[Operationskontrakter 28](#_Toc263333753)

[Designmodel 29](#_Toc263333754)

[Designklassediagram 35](#_Toc263333755)

[Singleton 36](#_Toc263333756)

[Beskrivelse af design klassediagrammet 36](#_Toc263333757)

[Arkitektur 37](#_Toc263333758)

[Unittest 37](#_Toc263333759)

[Konklusion 38](#_Toc263333760)

## Forord

Denne rapport er udarbejdet af DM71-gruppe4 som dokumentation i forbindelse med 1.semesterprojekt ved datamatikerstudiet på UCN. Det forventes at læseren har forstand på Unified Modeling Language (UML) for at forstå diagrammerne, der indgår i rapporten.

## Indledning

Vestbjerg Byggecenter A/S ønsker et nyt it-system, da deres nuværende system, som er baseret UNIX, ikke er tilstrækkelig konkurrencedygtigt. Virksomheden har opstillet en række krav til hvilken funktionalitet sådan et system skal indeholde. Disse krav er opdelt på kort og på langt sigt. På kort sigt, stræber de efter et salgssystem, herunder de basale applikationer, såsom lagerstyring, ordrestyring, kundehåndtering, medarbejderhåndtering og salgsstatistikker. På langt sigt, ønsker de integration af e-handel.

Dette projekt beskæftiger sig med Vestbjerg Byggecenter A/S kommende it-system, herunder analyse af selve virksomheden og implementering af systemet. Til at starte med vil virksomheden blive analyseres, en såkaldt it-forundersøgelse, for at opnå en forståelse af organisationen, strukturen, deres strategi, vision og arbejdsgang. Dette sikrer at systemet bliver udviklet optimalt i forhold til ledelsen og medarbejdernes kompetencer. Dernæst gennemgås udviklingen for it-systemet, som tager udgangspunkt i it-forundersøgelse. Udviklingen er en iterativ-proces og er i dette projekt delt op i tre iterationer. I første iteration, gennemgås kravspecifikation og use-cases, herunder prioritering af disse. Den anden iteration dokumenterer implementering af den højst prioriterede use-case, samt design processen af denne. Den tredje eller sidste iteration, berører de samme områder, som anden iteration, med den undtagelse at det er for de næstprioriterede use-cases.

### Kodestandard

Navngivning:

* Alle metoder og variabler skrives på engelsk.
* Metodenavne skal være beskrivende for hvad metoden gør.
* Variabelnavne skal være beskrivende for hvad de indeholder.
* Variabel- og metodenavne skrives med lille begyndelses bogstav. Hvis navnet består af flere ord skrives de følgende ord med stort begyndelses bogstav.
* Klassenavne skrives med stort begyndelses bogstav.

Indrykning:

* Der anvendes den standard som Bluej foreslår, altså en tab indrykning.

Løkker:

* Start og slut tuborg {/} skal stå over hinanden for at gøre koden mest mulig overskuelig.

Dokumentation:

* Alle metoder skal kommenteres, så det fremgår hvilke parametre og evt. retur værdier der er (@param, @return).

## It-forundersøgelse

For at få et klart overblik over hvilket system Vesterbjerg Byggecenter A/S ønsker, samt hvordan systemet skal se ud og hvilke funktioner der indeholder, bliver vi nødt til at foretage en it-forundersøgelse, før vi kan begynde at opsætte krav-specifikationer.

### BATOFF

Systemet skal avendes til at administrere varer, ordrer, udlejning, kunder, ansatte. Det bruges af ansatte og ledelsen i virksomheden, hvor forudsætningen er at de har basale it-kundskaber. Det bliver i udviklet i java, hvilket betyder at det kan kører både på Linux, Mac eller Windows. Af hardware krav, er en billig pc-platform tilstrækkelig, som kobles til en central server. Java runtime skal installeres før systemet kan afvikles.

#### Betingelser:

Ledelse, samt medarbejdere med varierende it-kundskaber. Udvikles i samarbejde med brugerne. Brugervenligheden prioriteres højest.

#### Anvendelse:

Støtte til styring og overvågning af vare, ordrer, udlejning, salg, kunder og ansatte. Medarbejdere, der arbejder med udlejning, salg, administration og lagerstyring.

#### Teknologi:

Pc’er med forbindelse til en central server. På sigt med adgang til specielle funktioner via internettet. Det er påkrævet at java runtime er installeret.

#### Objekter:

Kunde, ansatte, varer, salg, udlejning

#### Funktionalitet:

Registrering af kunder, medarbejder og varer, samt administration og overvågning af ordre, salg og udlejning. Systemet skal primært bruges til administration af varer, kunder, ansatte og udlejning.

#### Filosofi:

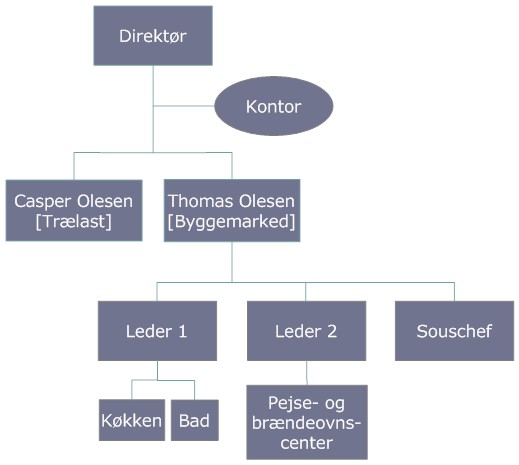
Et administrativt værktøj med overvågningsfunkiton.

### Virksomhedskarakteristik

#### Organisationsprincip

Vi er kommet frem til det er Linjestabsprincippet, selvom der kun er en lille stabsfunktion som består af fire medarbejdere. Linjeprincippet er godt for medarbejderne fordi de ved hvem deres chef er, og de altid ved hvor de arbejder, så de ikke bare bliver flyttet rundt mellem forskellige afdelinger. Lederne kan også blive aflastet ved hjælp af stabsfunktioner. Problemet ved linjestabsprincippet kan være at der er uenigheder om ansvarsplacering, og muligheder for konflikter mellem linjeledere og stabsledere.

Vestbjerg byggecenter har en dybde på fire niveauer, og den øverste leder Anders Olesen har et kontrolspænd på to.



### Organisationskultur

#### Lederkarakteristik

Menneskesyn, lederstil, Adizes lederroller, Ledergitter, autoritet

##### Anders Olesen

Baseret på McGregors X- og Y- syn mener vi han har et Y – Menneskesyn, da han har stor tillid til sine ansatte og de får meget frihed og et stort ansvar. Derudover vil gerne give sine medarbejdere gode arbejdsforhold, og han har også megen kommunikation med sine ansatte, da han nærmest er ”ven” med alle de ansatte. Anders Olesen har integratorrollen i Vestbjerg byggecenter, han skaber meget motivation og korpsånd hos de ansatte, han leder også gennem teamwork idet han har en ledergruppe bestående af fire personer. I ledergitteret er Anders giver vi 4,9 han er mere end hyggeonkel end han er holdleder, fordi han fokusere meget på medarbejdernes behov, og knap så meget på de produktive mål, men alligevel er han ikke bange for at afskedige folk hvis de ikke laver noget. Anders passer ikke rigtig ind i Jack Bobo’s ledelsesstile, dette kan nok begrundes med han sidder så højt i hierarkiet, det vil sige, han ikke behøver at svinge pisken. Dette kan også begrundes med han ikke har særligt stort ansvar, og har overladt ansvaret til sine sønner og souschefen. Derfor har han heller ikke nogen rigtig autoritet.

##### Casper Olesen

Casper Olesen har et X-syn. Det virker som om han har mindre tillid til sine ansatte, da han gerne vil hjælpe alle med at løse deres opgaver så godt som overhovedet muligt, dette kan også ses som en form for kontrol.

Ser vi på Jack Bobo’s ledelsesstile, er Caspers adfærd ikke helt passende. Casper vil gerne hjælpe sine medarbejdere til at gøre tingene bedst muligt, dette skyldes også han har en uddannelse inden for byggebranchen og mener han ved hvordan alting skal håndteres bedst muligt.

I Adizes 4 lederroller passer Casper perfekt ind i rollen som producent. Han har stor faglig viden omkring sit fag, han lyder som en travl mand, eftersom han skal ud og hjælpe alle sine medarbejdere med at gøre deres arbejde bedst muligt, dette kan også skyldes han har et stort præstationsbehov.

##### Thomas Olesen

I forhold til ”McGregors X og Y menneskesyn”, har Thomas Olesen et Y-menneskesyn, idet han fokuserer på den enkelte medarbejders trivsel på arbejdspladsen. Derudover sørger han også for at medarbejderne har mulighed for selvrealisering ved at udvikle sig via arbejdet.

Gå vi over til Jack Bobo’s ledelsesstile, passer hans adfærd ikke direkte ind i de 5 ledelsesstile, som er nævnt. Han har sin egen unikke måde at håndtere sine medarbejdere på, ved udstråle en meget karismatisk ledelsesstil.

Med hensyn til Schiens fire menneskesyn, har han en blanding mellem det selvrealiserende og det rationelles syn. Dette ses ved at han som førnævnt, sørger for bl.a. at arbejdsforholdene og de sociale forhold er i orden.

Ud fra Adizes lederroller har han integratorrollen, idet han efter bedste evne prøver at skabe en god kemi i virksomheden, gennem sociale arrangementer, hvilket i sidste ende kan give mere motivation og et bedre teamwork på arbejdspladsen. I virksomheden gennemfører han også et såkaldt MUS (medarbejderudviklingssamtaler) for at sikre at de enkelte medarbejder føler sig værdsat og får mulighed for selvudvikling.

Baseret på ”Blake & Moutons ledergitter”, får han tallet 6,9, da han er en mellemting mellem ”Hyggeonkel” og ”Holdlederen”, men mest de sidstnævnte. Han fokuserer nemlig på medarbejdernes behov, således at de trives bedst. Dette sker, som førnævnt, vha. sociale aktiviteter, såsom indendørs fodbold, bowlingarrangementer og lign. Han forsøger også efter bedste evne at motivere og engagere sine medarbejdere gennem det førnævnte MUS.

I forhold til autoritet, er Thomas præget af en karismatisk autoritet, idet han prøver at tilgodese medarbejdernes behov. Han forstår at snakke med medarbejderne og sætte dem op til opgaverne.

##### Virksomhedskulturtype

Virksomheden er baseret på familiekulturen, da lederen er mere orienteret mod den enkelte person end mod opgaveløsningen. Der er et tæt forhold mellem de ansatte. Anders Olesen, som til dels står for ledelsen af virksomheden, står øverst i hierarkiet, og kulturen er præget af traditioner og skikke. Han har selv startet virksomheden fra bunden, og han har sine egne ideer om hvordan virksomheden skal drives og ledes, men er dog opmærksom på at ledelsesopgaverne ikke er som de var engang. Derfor er en del ledelsen overladt til hans sønner, da de er mere moderne med hensyn til ledelseskultur.

##### Konsekvenser for motivation og arbejdstilfredshed

Grundet ledelsens y-menneske syn, kan dette medfører at medarbejderne er mere positiv stemt, og derfor mere effektive. På den måde kan antallet af interne konflikter mellem de ansatte nedsættes, idet mulighederne bliver bedre for at kommunikere med hinanden, hvorimod hvis dette ikke havde været tilfælde, kunne der opstå flere konflikter, hvilket i sidste ende kan medføre nedsat effektivitet. Der fokusere meget på at medarbejderne skal rystes sammen og have noget tilfælles gennem forskellige sociale aktiviteter i virksomheden.

# Regnskabsanalyse

Vi foretager en regnskabs analyse for at danne os et overblik over hvordan udviklingen er i virksomheden. Er udviklingen god, kan dette betyde at virksomheden har råd til at fastansætte en person, udelukkende til system-administation.

Virksomhedens regnskab for de sidste 3 år.

Resultatopgørelser:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **2007** | **2008** | **2009** |
| Omsætning |  | 62560 | 68680 | 69050 |
| Vareforbrug |  | 43792 | 47389 | 46954 |
| Salgsprovision |  | 800 | 810 | 830 |
| **Dækningsbidrag** |  | 17968 | 20480,8 | 21266 |
| **Kontante kap. omk.** |  |  |  |  |
| Salgsfremmende omk. |  | 2300 | 2400 | 2560 |
| Gager |  | 9690 | 9800 | 10230 |
| Lokaleomkostninger |  | 1200 | 1240 | 1269 |
| Reparation og vedligeholdelse |  | 450 | 560 | 600 |
| Bilers driftsomkostninger |  | 560 | 789 | 867 |
| Tab på tilgodehavender |  | 345 | 456 | 432 |
| Øvrige omkostninger |  | 345 | 467 | 678 |
| Kontante kapacitetsomk. |  | 14890 | 15712 | 16636 |
| **Indtjeningsbidrag** |  | 3078 | 4769 | 4630 |
| Afskrivninger |  | 1201 | 1208 | 1302 |
| **Resultat før renter** |  | 1877 | 3561 | 3328 |
| Renteomkostninger |  | 456 | 543 | 432 |
| **Resultat før skat** |  | 1421 | 3018 | 2896 |
| Selskabsskat |  | 355 | 754 | 724 |
| Nettoresultat |  | 1066 | 2263 | 2172 |
|  |  |  |  |  |

Balancer:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Aktiver | 2006 | 2007 | 2008 | 2009 |
| Anlægsaktiver |  | 11.456 | 12.451 | 12.455 |
| Varelager | 12.000 | 11.000 | 12.300 | 12.800 |
| Varedebitorer |  | 3.456 | 4.500 | 4.300 |
| Likvide beholdninger |  | 1.000 | 1200 | 1300 |
| Aktiver i alt |  | 26.912 | 30.451 | 30.855 |
| Passiver |  |  |  |  |
| Aktiekapital |  | 5.000 | 5.000 | 5.000 |
| Reserver |  | 6.700 | 8.020 | 10.200 |
| Lån i ejendom |  | 8.434 | 8.034 | 6.250 |
| Kassekredit (max.2.000) |  | 490 | 1.690 | 1.399 |
| Momsgæld |  | 565 | 670 | 1.056 |
| Varekreditorer |  | 4.900 | 5.030 | 5.000 |
| Andre kreditorer |  | 456 | 1.267 | 1.248 |
| Skyldig selskabsskat |  | 367 | 740 | 702 |
| Passiver i alt |  | 26.912 | 30.451 | 30.855 |
| *Egenkapital* |  | 14336 | 15037 | 14843 |

Virksomhedens regnskab ser godt ud, ud fra tallene på regnskabet. Omsætningen er stigende, og endnu vigtigere, dækningsbidraget er ligeledes stigende. For at få et helhedsindtryk af virksomhedens økonomi og udvikling skal man udregne nogle nøgletal og analysere dem.

For at undersøge virksomhedens rentabilitet, og hvor god virksomheden er til at tjene penge og lave overskud, skal vi først kigge på afkastningsgraden.

Afkastningsgrad (resultat før renter \* 100 / aktiver):

|  |  |  |
| --- | --- | --- |
| 2007 | 2008 | 2009 |
| 16,38 | 28,6 | 26,72 |

Virksomhedens afkastningsgrad er stigende, på trods af et mindre fald fra 2008 til 2009. Afkastningsgraden er usædvanlig høj, og må siges at være meget flot. Trods et mindre fald forrenter virksomheden stadig 26 øre pr. krone de investerer i virksomheden, hvilket er særdeles meget. Afkastningsgraden er steget med 10,34 % -point, hvilket svarer til en stigning på 63%.

Sammenligner man med markedsrenten(6-7%), der må siges at ligge meget lavere er det virkelig en virksomhed med høj indtjeningsevne.

For at se hvad der ligger til grund for den høje stigning i rentabiliteten kan vi kigge på nogle flere nøgletal. F. eks Overskudsgraden, kan sige meget om rentabilitetens udvikling.

Overskudgrad(resultat før renter \* 100 / nettoomsætning):

|  |  |  |
| --- | --- | --- |
| 2007 | 2008 | 2009 |
| 3,0 | 5,18 | 4,81 |

Som forventet ser vi en markant stigning i Overskudsgraden (60 %), og derved en forbedret indtjeningsevne. Den forbedrede indtjeningsevne er hovedårsagen til den store stigning i Afkastningsgraden.

Dette betyder dog ikke at det er den eneste faktor, der kan have påvirket. Ligeledes kan aktivernes omsætningshastighed fortælle om virksomhedens kapitaltilpasning er forbedret.

Aktivernes omsætningshastighed(nettoomsætning / aktiver):

|  |  |  |
| --- | --- | --- |
| 2007 | 2008 | 2009 |
| 5,46g | 5,52g | 5,54g |

Vi ser en lille forbedring (1 %) på aktivernes omsætningshastighed, hvilket viser at virksomheden kan forbedre tilpasningen af den investerede kapital til aktiverne.

Vi kan se at virksomhedens nøgletal er rigtig flotte, men afkastningsgraden viser kun virksomhedens forrentning af hele kapitalen, uanset om denne er finansieret af ejerne eller gældsforpligtet. Derfor er det relevant at kigge på Egenkapitalens forrentning, for at få et indblik i hvor meget ejerne har forrentet af deres indskud i virksomheden.

Egenkapitalens forrentning(resultat efter renter \* 100 / egenkapital):

(Vi skal her bruge resultatet efter renter, da renterne jo tilfalder gælden og ikke egenkapitalen)

|  |  |  |
| --- | --- | --- |
| 2007 | 2008 | 2009 |
| 9,91 | 20,07 | 19,51 |

Egenkapitalens forrentning er steget fra 9,91 til 19,51 (97 %), hvilket er meget tilfredsstillende, hvis man sammenligner med markedsrenten (6-7 %). Virksomheden har tjent på den gæld de har haft, da Egenkapitalens forrentning ligger langt over Afkastningsgraden.

Udviklingen i virksomhedens rentabilitet er ud over alle grænser imponerende og meget tilfredsstillende for virksomheden, der er i høj udvikling. Der er derfor råd til at få det it-system der er bedst for virksomheden, samt få fastansat en system-administator.

|  |  |
| --- | --- |
| Strengths (intern) | Weaknesses (intern) |
| * God økonomi * Høj omsætning * Meget erfaring i branchen. * Administrerende direktør med meget branche erfaring, opnået efter mange år i faget. * Afdelingsledere med gode uddannelser indenfor deres områder. * Medarbejdere der har været i virksomheden i flere år * Stor kundekreds | * Ingen logistik på lagerne (lokationsstyring) * Dårligt system (mange mangler) * Manglende optimering * Besværlig udlejning af værktøj og lign. |
| Opportunities (ekstern) | Threats (ekstern) |
| * Nye kunder * Opkøb af konkurrenter, med dårligere økonomi (finanskrise) * Ny teknologi (E-handel) | * Svækket købekraft (finanskrise) * Byggestop (finanskrise) * Konkurrenter |

### Problemer, hypoteser og løsningsmuligheder

Tabel over problemer og vores bud på en løsning dertil.

|  |  |  |
| --- | --- | --- |
| Problemer | Hypoteser | Løsningsmuligheder |
| Man ved ikke hvor varerne befinder sig i virksomheden | Der anvendes ikke lokationsnumre så man kan se hvor varerne er | IT til oversigter over varerne |
| Langsom betjening ved udlejning af værktøjer | Alt udlejning sker manuelt | IT til at styre udlejning og hjemkaldelse |
| Synlighed af prisskilte er dårlig | Der er ikke mulighed for forskellige størrelser | IT til brug af printning af prisskilte |
| Der er ikke nogen kontrol af solgte varer pr. sælger | Ingen statistikker | IT til liste over solgte varer |

### SWOT Analyse

### Valg af strategi

Ledelsesgruppen vil primært fokusere på en maxi-maxi strategi, hvor de vil udnytte virksomhedens interne styrker, så som god økonomi og meget branche-erfaring. Derudover vil de prøve at tage udnyttelse af de eksterne muligheder de har, f. eks udvide deres kundekreds og slå nogle af de mindre konkurrenter ud.

Sekundært ønsker de at minimere deres svagheder, ved implementering af et nyt IT-system, der skal optimere den daglige proces i virksomheden.

### Ansoff’s vækstmatrice

|  |  |
| --- | --- |
| Produkter | |
| Nuværende | Nye |
| Markeder | Nuværende markeder | **Markedspenetrering** | **Produktudvikling** |
| Nye markeder | **Markedsudvikling** | **Diversifikation** |

Ud fra Ansoff’s vækstmatrice er der to strategiske veje, virksomheden kan gå, for at øge deres vækst.

Den første vej de kan gå, som også er den de bør fokusere på er Markedspenetrering. Denne strategiske vej, har til formål at udvide virksomhedens markedsandel, med de samme produkter. Måden hvorpå markedspenetrering opnås er ved at anvende de relevante handlingsparametre, også kendt som de 4(7) p’er: Pris, Produkt, Place, Promotion, People, Proces samt Prøve.

Pris er en relevant handlingsparameter og kan til dels anvendes sammen med Promotion. Det spiller ofte en stor rolle for kunderne, hvad prisen er og om de eventuelt kan indhente rabatter. Ledelsesgruppen bør overveje at indføre nogle rabatter og tilbud, der skal føre til flere salg, og eventuelle mersalg, for både private og erhvervs-kunder.

Ligeledes er Product relevant, da den ligesom pris har en stor indflydelse på hvor kunderne vælger at handle. Virksomheden skal være sikker på at de altid har de bedste produkter, med den bedste kvalitet kontra pris.

People er i høj grad en af de vigtigste handlingsparametre for virksomheden, i nogle tilfælde mere relevant end pris! Dette gælder både for private- samt erhvervskunder, da kundes oplevelse er succeskriterium for om de handler det samme sted næste gang. Nogle kunder betaler gerne en højere pris, hvis de føler de får en bedre oplevelse før, under samt efter handlen. Virksomheden har allerede en stor kundekreds, men ligger i et lokalt miljø, og møder ofte de samme kunder. Desuden har alle medarbejderne i virksomheden været der i flere år, og har derfor haft tid til at lære de mest tilbagevendende kunder at kende, selvfølgelig på forskellige plan. Alle medarbejderne i virksomheden bør sætte mere fokus på at få et personligt forhold til deres kunder, især erhvervskunderne er meget vigtige. Et personligt spørgsmål, eller et smil af tømrerens joke, kan afgøre om kunden føler sig værdsat og velkommen.

Proces er en vigtig handlingsparameter for at få det interne i virksomheden optimeret. Virksomheden bør implementere et lagersystem, der kan optimere deres genbestilling af varer, spildtid på lokation af varer samt øge eller sørge for overholdelse af leveringsaftaler til kunder.

Den anden strategiske vej virksomheden kan vælge er Markedsudvikling, hvor de skal forsøge at komme i kontakt med nye kundegruppe og præsentere deres vareudvalg til dem. Denne vej mindre relevant end Markedspenetrering, men dette betyder ikke at de må glemme den. Selvom de ikke bør gå denne vej primært, kan der stadig være kunder at hente, selv ved lav arbejdsindsats på geografisk markedsudvikling. Virksomheden bør bruge deres bedste kort, hvilket er brancheforståelse. De har været i branchen i mange år, og har derved opnået en vis status, der måske kan benyttes til at stjæle kunder fra de store byggemarkeder, som f.eks. Bauhaus. Desuden skal Vestbjerg Byggecenter A/S også ligge stor vægt på at de har personale, der er dedikeret til faget og branchen, til forskel fra de større byggemarkeder.

### IT strategi

Virksomhedens IT-strategi går ud på at skifte et UNIX-baseret system ud med et nyt, da alle lige fra ledelse til medarbejder er trætte af det nuværende system. Dette system har nemlig visse mangler med hensyn til informationer til styring af en virksomhed. Det de ønsker er et mere brugervenligt system, med udvidet funktionalitet, såsom kundehåndtering, lagerstyring, ordrestyring, udlejningssystem, e-handel og med mulighed for at lave salgsstatistikker.

Kundehåndteringssystemet skal gøre det muligt at håndtere kunder.

Ved lagerstyring skal det være muligt at indrapportere nye varer og leverandørdata, samt fjerne disse igen. Fx når en kunde købe en vare, skal denne fjernes fra lagerbeholdningen. Vedrørende lageroplysninger, skal der være kontrol med hvor meget der er på lager af de forskellige varer. Mere specifik, vil det sige at der skal være et minimum- og et maksimumlager samt genbestillingsmængder når lageret kommer under minimumsbeholdningen. Derudover skal der samtidig kontrolleres på lagerbeholdning, at der ikke bliver bestilt for store mængder, således at det overstiger maksimumsbeholdningen for den pågældende vare. På den måde undgås at der er for lidt eller for meget af en vare.

Ordrestyringen skal give mulighed for at holde styr på ordrer. Dette indebærer at der er mulighed for at følge en vare, fra den bliver bestilt til den faktureres. Derudover skal der være mulighed for at lave tilbud, ordre, ordrebekræftelse, ekspeditionseddel, følgeseddel og faktura.

Udlejningssystemet har tidligere kørt manuelt, men i det nye system, skal det være muligt at styre udlejning og hjemkaldelse af udlejningsmaskiner.

Med hensyn til e-handel, har virksomheden en hjemmeside, da de er med i kæden XL-Byg, hvorfra nethandel kan foretages. Den er dog ikke fuldstændigt integreret hvad angår lagerstyring og bogholderi. Disse applikationer ønskes implementeret. Derudover ønskes der også mulighed for at handle med grossister og håndværkere via internettet.

I salgssystemet ønskes der, udover de basale funktioner, en yderligere funktionalitet, således at det er muligt for ledelsen at lave salgsstatistikker på kunder, varer og leverandører samt på hver medarbejder. Der skal være mulighed for en rabatordning ved forskellige former for indkøb, dog under visse betingelser.

Med hensyn til registrering af nye varer, salg og udlejning, skal det foregå vha. stregkoder.

Systemet vil blive implementeret trinvis, således at ikke alle funktioner i virksomheden kommer til ophøre, mens denne proces er i gang. På den måde mister virksomheden ikke omsætning under omstillingen.

Det nye system vil dog i sidste ende komme til at berøre alle medarbejdere og ledelsens daglige drift pga. af de store ændringer i IT-systemet. 

### Applikationer og information

På kort sigt:

IT-systemet skal implementeres, således at det bliver lettere at håndtere salg, ordre, vare, kunder, samt mulighed for lave salgsstatistikker.

På længere sigt:

Mulighed for bedre integration med e-handel, således at der kan handles med grossister og håndværkere.

### Teknologi

Der bliver stillet pc’er til rådigheder, som via et netværk har adgang til en server, hvor al information lagres. Derfor vil kravene være almindelige pc’er, samt en server. Derudover skal der opsættes et netværk, således at disse kan kommunikere med hinanden.

### Organiseringen af IT-funktionerne

Der skal hyres et tredje-parts firma til at vedligeholde systemet i fremtiden. Der skal laves en special aftale da det ikke er særlig ofte der vil være problemer med deres system. Det vil være overflødigt at have en fastansat til at ordne sådanne problemer. Så de skal have lavet en aftale sådan, at de kan ringe til firmaet og betale for en engangs ydelse.

### It-handlingsplan

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Business domain | | | | | | Technology domain | | | | To  tal |
|  | ROI | SM | CA | MI | CR | OR | SA | DU | TU | IR |
| Vægt | +5 | +3 | +4 | +4 | +1 | -1 | +2 | -1 | -2 | -1 |
| Lagerstyring | 25 | 12 | 12 | 20 | 4 | 0 | 6 | -1 | -6 | -1 | 71 |
| Ordrestyring | 20 | 12 | 16 | 20 | 4 | 0 | 6 | -1 | -6 | -1 | 70 |
| Udlejningssystem | 10 | 3 | 12 | 12 | 0 | 1 | 2 | -2 | -6 | -1 | 31 |
| Kundehåndtering | 15 | 12 | 12 | 8 | 0 | 1 | 4 | -2 | -6 | -1 | 43 |
| E-handel | 15 | 3 | 12 | 12 | 4 | -4 | 2 | -4 | -8 | -2 | 30 |
| Salgssystem | 20 | 15 | 16 | 20 | 3 | 0 | 10 | -1 | -6 | 0 | 77 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Vurderingsfaktorer:  ROI = Return of investment  SM = Strategic match  CA = Competitive advantage  MI = Management information  CR = Competitive response  OR = Organizational or project risk  SA = Strategic IS architecture  DU = Definitional uncertainty  TU = Technical uncertainty  IR = IS infrastructure risk | | | | | | | | | | | |

Tabel (reference til tabel) viser en Parker/Benson matrice for Vestbjerg Byggecenter A/S, hvor de enkelte applikationer får en prioritering, baseret på deres totale score. Dvs. applikationen med den højeste score, får første prioritet.

Ud fra (reference til tabel), er prioritering som følger:

1. Salgssystem
2. Lagerstyring
3. Ordrestyring
4. Kundehåndtering
5. Udlejningssystem
6. E-handel

Salgssystemet er vores første prioritet, da dette er fundamentet for de andre applikationer i systemet.

Lagerstyring er vores anden prioritet, fordi det giver mere fleksibilitet hvad angår håndtering af vareindkøb, salg og lign. Dette er vigtigt for at salgssystemet kan køre optimalt, og derfor denne prioritering.

Ordrestyring er vores tredje prioritet, da dette bl.a. kan give virksomheden et indblik i salgsfrekvensen for en vare, samt bedre kundeservice, da der kan følges ajour med lagerbeholdningen. Dette er vigtigt for at salgssystemet kan køre optimalt, og derfor denne prioritering.

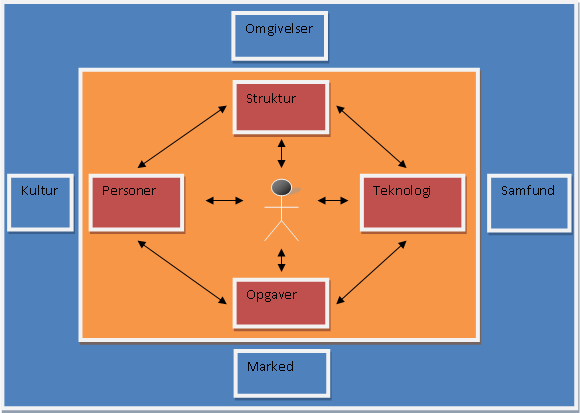
Kundehåndtering er vores fjerde prioritet. Denne applikation kan give et bedre serviceniveau for kunderne, samt bedre overblik over faste kunder for virksomheden. På den måde er oplysningerne om de enkelte kunder lagt ind i systemet i forvejen, og kan bruges fremover. Denne applikation er ikke nødvendig for driften af de andre højere prioriterede applikationer, men kan hjælpe, som nævnt, på serviceniveauet. Derfor denne prioritering.

Udlejningssystem er vores femte prioritet. Denne applikation kan give et bedre serviceniveau ved udlejning. Systemet er ikke nødvendigt for driften af virksomheden, og derfor denne prioritering.

E-handelssystemet er vores sjette prioritet. Denne applikation kan være med til at synliggøre firmaet, samt mulighed for at handle med grossister og håndværkere. Derudover kan det også forbedre serviceniveauet, da kunder på den måde kan bestille over nettet. Systemet er dog ikke vigtigt for firmaets strategi på nuværende tidspunkt, men er noget, som kan implementeres engang i fremtiden. Derfor denne prioritering.

### Implementering:

For at se hvorfor nogle afledte konsekvenser, forandringen i deres IT system, har for resten af organisationen, bruger vi Leavitt’s systemmodel. Den opdeler virksomheden i 4 delsystemer med centrum omkring en ændringsagent. Det den betyder, er at sker der en ændring i et delsystem, vil det have betydning for de andre.



Ændringsagenten er den der har ansvaret for implementeringen, det vil sige os. Så ser man på hvorfor et delsystem der primært bliver de største ændringer i. I det her tilfælde er det teknologien, vi med systemet får lavet mest om i. Teknologi er de redskaber der anvendes i forbindelse med opgavernes udførelse, såsom udstyr, arbejdsprocesser og lignende. Ændringen har så konsekvenser for de andre delsystemer. Struktur vedrører virksomhedens organisation, arbejdsdeling, kommunikationsstruktur, beslutningsstruktur osv. I det at virksomheden gerne vil have at det kun er ledelsen der kan ændre priser i systemet, betyder det en ændring i beslutningsstrukturen. Opgaver er salg og services og bliver ændret i den form de sælger deres varer på. Der bliver også ændringer for måden medarbejderne kommer til at skulle have deres daglige arbejdsgang, så derfor vedrører ændringen og Personer. Omgivelserne vil også altid stille krav til virksomheden. Kulturen har en indflydelse på nuværende og fremtidige medarbejderes kompetencer og adfærd. Samfundet skal også have et nogenlunde teknologisk niveau for at kunne bruge den teknologi en virksomhed gerne vil have implementeret.

Alt på en gang:

|  |  |
| --- | --- |
| Plus | Minus |
| * Der bliver ikke på et tidspunkt problemer mellem gammelt og nyt system. | * Kan være kompliceret * Kan være så tidskrævende, at hele virksomheden går i stå i en tidsbegrænset periode. |

Hver afdeling for sig:

|  |  |
| --- | --- |
| Plus | Minus |
| * Dele af virksomheden vil være i gang, imens implementering finder sted. * Mere overskueligt. | * Tager længere tid. * Dele af virksomheden vil i en tidsbegrænset periode, køre med forskellige systemer. |

Pga. plusser og minusser, sker implementeringen hver afdeling for sig, da det vil betyde en del for virksomheden, at den kan holde virksomheden i gang mens implementeringen finder sted.

### Brugerdeltagelse:

#### Forandringsstrategi i forhold til menneskesyn og organisationsopfattelse



Menneskeopfattelsen i virksomheden er Y-teori. Godt nok er direktøren fra den gamle skole, men han ved godt det er nye tider, så han har uddelegeret ansvar, og har ændret sig fra x- til y-menneskeopfattelse. Organisationsopfattelsen er harmonisk. De har en stabil struktur og der er velintegrerede enheder. Det vil sige at virksomheden med højst sandsynlighed har brug for en brugerdeltagelse på repræsentationsniveau. Det vil sige at beslutningstagerne har den mening at medarbejderne bør medvirke ved opbyggelsen af det nye system, det kan også ses i deres vision ”*Vore folk som skal bruge systemet skal involveres i beslutninger vedr. systemets udformning”*. De mener at deres medarbejdere har de samme mål og værdier.

## Iteration 2

### Aktivitetsdiagram



### Use-cases

#### Use-case diagram



#### Use-case beskrivelser

Use-case: createSale Id: UC1

Primær aktør: Sælger

Andre interessenter: Køber

Frekvens: 5 – 50 gange dagligt

Pre betingelser: Sælger er logget ind, køber der ønsker at købe en vare/varer og denne vare eksisterer og er på lager.

Post betingelser: En vare/varer bliver solgt

Basis succes flow:

1. Sælger scanner vare/varer ind i systemet
2. Systemet viser en liste med vare/varer og deres oplysninger og finder den/dem i lageret
3. Sælger afslutter salg og systemet fjerner varen/varerne fra lageret

Alternative flows:

1.a

1. Sælger leder i lager efter vare som kunde efterspørg
2. Systemet viser varens lagerstatus
3. Hvis varen ikke er på lager, registreres kundens tlf. nr.

1.b

1. Sælger leder efter vare på lager
2. Systemet returnerer vare
3. Sælger markerer vare til levering
4. Punkt 1 – 3 gentages efter hvor mange forskellige antal varer der ønskes
5. Sælger angiver konto id eller om der skal oprettes midlertidig leveringsadresse

1.b.1

1. Er konto oprettet angiver sælger konto id
2. Systemet returnerer valgt konto og vare/varer
3. Sælger bekræfter valg
4. Systemet udskriver følgeseddel og fjerner varen/varerne fra lageret
5. Sælger afslutter salg

1.b.2

1. Skal der angives leveringsadresse angiver sælger leveringsoplysninger
2. Systemet returnerer valgte leveringsoplysninger, varer og slutpris
3. Sælger angiver om hvor meget der bliver betalt
4. Systemet returnerer hvor meget der skal gives tilbage, udskriver kvittering og fjerner varen/varerne fra lageret
5. Sælger afslutter salg

Use case: Rent Id: UC2

Basis succes flow:

1. Sælger leder i liste over ledige værktøjer
2. Systemet returnerer fundne værktøj
3. Sælger angiver lånekriterier
4. Systemet returnerer lånekriterier og slutpris
5. Sælger angiver hvor meget der bliver betalt
6. Systemet returnerer hvor meget der skal gives tilbage, udskriver kvittering og fjerner værktøj fra listen af ledige værktøjer
7. Sælger afslutter udlejning

Use case: Handle Employee CRUD Id: UC3

Basis success flow: ”Create”

1. Bruger angiver medarbejder kriterier
2. Systemet returnerer kriterier
3. Bruger bekræfter valg

Use case: Handle Customer CRUD Id: UC4

Basis success flow: “Create”

1. Bruger angiver kunde kriterier
2. Systemet returnerer kriterier
3. Bruger bekræfter valg

Use case: Handle Item CRUD Id: UC5

Basis success flow: “Create”

1. Bruger angiver varens kriterier
2. Systemet returnerer kriterier
3. Bruger bekræfter valg

### Domænemodel



#### Domænemodel beskrivelse

##### Klasser

Account

Definition:

En konto kan tilknyttes til en kunde, idet kunder kan gemmes i systemet. Med klassen ”Konto”, kan vi se hvilken rabat, som skal gives, samt mulighed for at betale direkte uden at skulle oplyse kontonummer og lign., da disse er gemt i systemet.

Multiplicitet:

Kontoen kan være oprettet men skal ikke altid have et salg i gang, derfor 0 til mange. Men en konto vil altid have en kunde den er oprettet til og det kan være en familie f. eks. og derfor vil den kunne være tilknyttet flere personer, derfor 1 til mange.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| type | Kontotype |
| regNo | Kontoens registreringsnummer |
| cardNo | Kontonummeret |
| accountDiscount | Konto rabat |

Customer

Definition:

For at gemme kunder i systemet, har vi oprettet klassen ”Customer”. Ved oprettelse af en kunde, gemmes de personlige oplysninger om denne, såsom navn, adresse og telefonnummer. Desuden arver denne klasse arver fra superklassen ”Person”.

Multiplicitet:

Kunde er en generalisering af klassen person. En kunde kan kun have en enkelt konto. Kunden kan have flere forskellige salg i gang. Kunden kan have en levering i gang fra i går og lave et kasse køb i dag. En kunde kan også leje flere forskellige slags værktøjer samtidig.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| customerID | Kundeid |

Discount

Definition:

Ved salg, kan medarbejderen give rabat i forhold til betalingsbetingelser og rabatordningen med kunden. For at holde styr på dette, har vi oprettet klassen ”Discount”.

Multiplicitet:

Et salg kan have en rabat, men den behøver ikke at have et, det bestemmer medarbejderen.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| qualityDiscount | Angiver rabatten på et givet salg |

Employee

Definition:

Ledelsen skal have mulighed for at håndtere medarbejdere, derfor klassen ”Employee”. Denne klasse lagrer informationer om medarbejderen. Desuden arver denne klasse fra superklassen ”Person”.

Multiplicitet:

Medarbejder er en generalisering af klassen person. En medarbejder kan lave et salg og udleje værktøjer og kan gøre det mange gange dagligt, men behøver ikke altid at være i gang med det, derfor en multiplicitet på 0 til mange.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| employeeID | Medarbejderens ID |
| Password | En kode, som medarbejderen bruger til at logge på systemet med |

Item

Definition:

Ved oprettelse af en vare, skal der holdes styr på dens forskellige egenskaber, hvilket denne klasse sørger for.

Multiplicitet:

Item har en samling af eksemplarer, hvor der er en multiplicitet på 0 til mange pga. eksemplarer er de varer som har et serie nr. og det er ikke alle genstandene der skal laves med det. Til SalesLineItem er multipliciteten også 0 til mange, genstanden vil først sættes til der når der laves et salg. Da de har flere afdelinger vil genstandene kunne ligge flere steder, men vil altid ligge mindst 1 sted. En slags genstand vil også kunne være samlet på et sted og derfor en multiplicitet på 1 til mange.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| itemPrice | Varens pris |
| description | Beskrivelse af varen |
| maxInStock | Maksimumsbeholdningen |
| minInStock | Minimumsbeholdningen |
| itemInStock | Antal varer på lageret |
| type | Varetype |
| brand | Brand/mærke |
| place | Afdeling |

Order

Definition:

Ved bestilling, oprettes en ordre, hvilket denne klasse holder styr på.

Multiplicitet:

Hvis der er en ordre, vil der være et salg, men der kan ikke være flere ordre på det enkelte salg.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| deliveryDate | Leveringsdato på ordren |
| orderID | Ordreid |

Person

Definition:

Superklassen ”Person” håndtere de egenskaber som ”Kunde” og ”Medarbejder” deler, hvilket er navn, adresse og telefonnummer.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| name | Kunden eller medarbejderens navn |
| address | Kunden eller medarbejderens adresse |
| Phone | Kunden eller medarbejderens telefonnummer |

Rent

Definition:

Denne klasse holder styr på udlejning af varer.

Multiplicitet:

Et lån vil kun tilhøre en kunde og det er kun lavet af en bestemt medarbejder, det kan dog godt bestå af flere varer men der vil altid være mindst en.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| period | Den periode som en given vare er udlejet i |
| delivered | Angiver om en given vare er returneret |

RentableItem

Definition:

For at adskille almindelige varer fra varer, som kan udlejes, har vi oprettet klassen ”RentableItem” med nogle særattributter.

Multiplicitet:

Enten kan genstanden være udlejet eller så er den ikke.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| itemPrice | Varens pris |
| description | Beskrivelse af varen |
| itemInStock | Antal |
| statusUdlaan | Status for udlån |

Sale

Definition:

Ved et salg, skal der holdes styr på hvornår salget blev oprettet, samt den totale pris. Derudover skal vi også vide om denne er en ordre.

Multiplicitet:

Det vil være oprettet af en bestemt medarbejder. Det kan være en ordre og det kan have en rabat, men det kan også være uden de 2. Det kan være et løssalg eller det kan være over en konto og det består af en eller flere genstande.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| date | Datoen for salget |
| price | Den totale pris for salget |
| order | Angiver om salget er af typen ordre |

SalesLineItem

Definition:

For at holde styr på antallet af hvor mange, der bliver solgt af en given vare, samt den totale pris, har vi oprettet klassen ”SalesLineItem”.

Multiplicitet:

En salgs genstand vil høre til et bestemt salg og vil være af en bestemt slags type. Det kan være et med serie nummer, men det kan også være et uden.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| quantity | Antal |
| totalPrice | Den totale pris (antal \* pris pr. eksemplar) |

Lokation

Definition:

Denne klasse bruges i forbindelse med lagerstyring. Den håndterer virksomhedens varer, ved at gøre det muligt at sortere i forskellige varegrupper.

Multiplicitet:

En lokation vil altid have noget liggende pga. min- og max beholdninger af genstande.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| type | Angiver varegruppen |

Unit

Definition:

Ved oprettelse af en vare, vil der også være eksemplarer af denne, hvilket klassen ”Unit” holder styr på. Hvert eksemplar får et unikt serienummer.

Multiplicitet:

Hvis der er et eksemplar, vil det være af en bestemt genstand. Det kan være usolgt og ikke høre til et salg, men hvis det bliver solgt, er det kun med i det ene salg.

|  |  |
| --- | --- |
| Attributter | Beskrivelse |
| serialNo | serieNummer på et givet eksemplar |

### Systemsekvensdiagrammer

#### createSale



#### handleCustomerCRUD



#### handleEmployeeCRUD



#### handleItemCRUD



#### rent



### Operationskontrakter

Operation: createSale(employeeID, itemID, saleDate, itemQuantity)

Kryds reference: Use case: Sale

Prebetingelser: En kunde med en vare, en oprettet medarbejder, oprettede varer.

Postbetingelser: Et salg bliver gennemført og der bliver fjernet solgte varer fra lager.

Operation: createEmployee(String name, String address, String phone, String password)

Kryds reference: Use case: Handle Employee CRUD

Prebetingelser:

Postbetingelser: En medarbejder bliver oprettet.

Operation: updateEmployee(String password, String name, String address, String phone)

Kryds reference: Use case: Handle Employee CRUD

Prebetingelser: En medarbejder er oprettet.

Postbetingelser: Medarbejderen bliver opdateret med en af sine informationer.

### Designmodel

#### Interaktionsdiagrammer

#### cancelSale



Den starter med at få en liste salgslinier fra et salgsID. Herefter kørere der et loop så hvor den fjerner alle de salgslinjer der er blevet oprettet så længe salgslinje listen er større end index tallet, når alle salgslinjerne er fjernet fra salget bliver selve salget slettet.

#### updateCustomerName



Først finder den kunden, derefter kan navnet sættes – det samme virker for alle de andre attributter, vi har valgt at lave det på denne måde så man ikke behøver at opdatere alle attributterne hver gang, men bare kan opdatere en.

#### createSale



CreateSale() starter med UIet kalder en metode nede i inputUI der spørger efter employeeID, og derefter kørere de hele vejen ned modellaget og finder den employee vi indtastede idet på. Egentlig skulle denne funktion være automatiseret efter vores login system, så den automatisk ville tage den medarbejder der er logget ind. Derefter bliver der fundet et item ud fra et itemID. Derefter kalder den en metode der spørger hvor mange items der er med det itemID. Derefter er der et check på om det ønskede antal er større end itemsInstock, og så længe det er det bliver du bedt om at indtaste et nyt tal. Herefter bliver selve kaldet kaldet til at oprette salget nede i controlleren, med flere af parametrene fra oven. Hvor den først finder employee, derefter kaldes konstruktøren i Sale så salget oprettes, derefter bliver der oprettet et discount objekt som knyttes til salget, derefter bliver salget lagt over i salgs containeren, og til sidst bliver der tilføjet en salgslinie.

#### addSalesLineItem



Vi starter med at hente et item fra item containeren igennem controlleren. Derefter er der et loop som checker om inputQuntity er størrere end de items vi lige har hentet. Derefter bliver itemIDet hentet sammen med prisen og items på lageret, derefter bliver saleslineitemet oprettet med en pris, og et objekt af item, og hvor mange af dem der er i. Nu bliver SLi så lagt ind i et salg, efter den er lagt ind i salget bliver stock opdateret så den passer med virkeligheden. Herefter er det hvis det er et salg med units, - dette bliver checket i det enkelte salg, der er det lavet sådan at den altid tager de varer der har lagt længst tid på lageret, det bliver gjort ved at tage de første fra listen, derefter bliver de tilføjet. Til sidst fjernes de fra lageret.

#### removeSalesLineItem



Den starter med at finde salget. Derefter finder den de salgslinjer der er i salget, så finder den hvilke og hvor mange items der er, derefter lægger den alle de items der var i salgslinjen tilbage i lagret. Hvis det er et item der indeholder units bliver units lagt tilbage. Derefter sletter den salgslinjen og opdatere prisen.

#### endSale



Starter med at finde salget igennem salgsID. Derefter finder den ud af ud af om det er associeret en kunde til salget, hvis der er en kunde tilknyttet bliver der tilføjet konto discount, hvis ikke, er der mulighed for standard rabatterne. Til sidst bliver prisen udregnet og salget afsluttes.

### Designklassediagram

### 

### Singleton

Singleton-mønstret sikrer at kun en instans kan laves af en klasse, og giver global adgang (global synlighed) til denne. Dette er en fordel, hvis vi kun ønsker netop én instans af en klasse, hvilket vi har benyttet bl.a. ved alle vores container-klasser. De klasser hvor vi har benyttet singleton-mønstrer, er illustreret vha. metoden getInstance() på design klassediagrammet. I nogle tilfælde har dog singleton-mønstret ikke egnet sig fx ved klassen item. Item har nemlig en ArrayList over unit-objekter, hvilket klassificerer den som en slags container, men i dette system skal det være muligt at oprette mange item-objekter, og derfor vil det være logisk at undlade singleton-mønstret i denne her situation. Derimod egner fx singleton-mønstret til visse container-klasser fx EmployeeContainer til employee-objekter, idet vi ikke behøver mere end én til at samle vores medarbejdere.

Her er et eksempel på hvorledes Singlenton-mønstret er implementeret i programmet:

|  |
| --- |
| 14 public class SaleContainer {  15  16 private static SaleContainer instance; //en instans af selve klassen  17 private ArrayList<Sale> saleList;  18  19 /\*\*  20 \* Konstruktør  21 \* Initialiserer en tom database(saleList)  22 \*/  23 private SaleContainer() {  24 saleList = new ArrayList<Sale>();  25 }  26  27 /\*\*  28 \* En metode, der tjekker om der er oprettet en instans af klassen. Hvis dette er tilfældet, oprettet en ny.  29 \* @return En instans af klassen  30 \*/  31 public static SaleContainer getInstance() {  32 if (instance == null) {  33 instance = new SaleContainer();  34 }  35 return instance;  36 } |

### Beskrivelse af design klassediagrammet

Design klassediagrammet viser et overblik over det udviklede system. Det tager udgangspunkt i use-case diagrammerne, domænemodellen og interaktions/system sekvensdiagrammerne. Disse nævnte elementer giver udviklerne en idé om hvilke dele der kunne indgå i sådan et systemet, heraf klasser, attributter og metoder. På samme måde som med domænemodellen, skaber design klassediagrammet et udgangspunkt, som udviklerne af systemet kan forholde sig til.

### Arkitektur

Vores system er bygget i en lagdelt arkitektur med tre lag. Dette gør det langt nemmere at lave ændringer i systemet i fremtiden, hvis der skulle være behov for det. Udover det gør denne arkitektur også at det er nemmere at holde styr på hvilken vej vores kald går, blandt andet så vi ikke kalder nedefra og op.

Det øverste lag er et UI lag, i vores tilfælde er det et tekst baseret interface, dette lag står får at udskrive alt det visuelle som brugeren kommer til at se, såsom menuen fejlmeddelelser og lignende. Vores UI lag kender til begge de to lag under sig, dette kaldes åben arkitektur. I en lukket arkitektur ville alle kaldende gå som følgende UI --> Controller 🡪 Model. I Ui laget er det eneste sted man må lave Print metoder i de to nedenfor skal der altid returneres noget.

Controller Laget samler alle metoderne fra Model laget. F.eks. når vi skal oprette en person er det Controllaget der opretter personen og sætter den ind i containeren. Dette giver et godt overblik over de forskellige metoder, i forhold til hvis man lavede det hele i model laget, og skulle finde rundt i det fra UI laget.

Model laget er det nederste. Her bliver alle objekter og containers holdt, udover det ligger alle informationerne hernede. Alle get – set metoder ligger hernede så der kan returneres objekter eller attributter til de to ovenstående lag.

### Unittest

Unittest er foretaget vha. JUnit i NetBeans. Formålet med disse test er at sikre metoderne fungerer i henhold til hensigten. Vi har kodet testmetoderne manuelt ud fra skabelonerne som JUnit genererer. Her er et eksempel på test af metoden createCustomer:

|  |
| --- |
| 47 public void testCreateCustomer() {  48 System.out.println("createCustomer");  49 String name = "Gert Hansen";  50 String address = "Grønnegade 12";  51 String phone = "98352010";  52 CustomerCtr instance = new CustomerCtr();  53 int customerID = instance.createCustomer(name, address, phone);  54 Customer result = instance.getCustomer(customerID);  55 assertEquals("Gert Hansen", result.getName());  56 assertEquals("Grønnegade 12", result.getAddress());  57 assertEquals("98352010", result.getPhone());  58 } |

Vores fremgangsmåde for test af andre metoder, foregår på sammen måde.

## Konklusion

Vi skulle lave et system til Vestbjerg byggecenter, hvor vi først lagde en tidsplan, som vi faktisk har overholdt og endda nogle gange været foran. For at lave et system til byggecentret måtte vi lave en analyse af byggecentret for at finde ud af hvordan det egentlig var opbygget. Vi fandt ud at deres økonomi var helt udmærket, og de sagtens kunne ansætte en konsulent til at holde styr på deres IT fremover. Herefter var der en række af krav systemet skulle overholde, og vi skulle finde ud af hvilke ting der skulle prioriteres højst, og ved hjælp af Parker/Benson valgte vi at salg ville være den vigtigste use-case at få implementeret. Nu skulle der designes dette var der mange uenigheder over, om hvordan det ville være smartest, og vi har da også ændret på det et par gange løbende, men vi syntes vi er kommet frem til en rigtig fornuftig løsning. Så skulle der kodes, dette gik ret stærkt, og somme tider for stærkt, så diagrammerne ikke blev helt overholdt. Alt i alt syntes vi, vi har lavet et fornuftigt projekt, der er nogle nogle småting i kodningen der måske kunne være bedre f.eks. sådan noget som halvdelen af discounten bliver udregnet i endSale(), i stedet for at udregne det hele i vores addDiscount() metode og alle kunderne der har en account får automatisk ti procents rabat. Altså er alting er ikke lavet helt færdigt, men hvis byggecentret ønsker det skulle det være nemt at udvide systemet med de ting der mangler.

Bilag

Brugermanual:

Man starter i hovedmenuen, hvor man har 7 valgmuligheder.

1. Salg
2. Lagerstyring
3. Kundehåndtering
4. Medarbejderhåndtering
5. Udlejning
6. Nyt udlejningsudstyr
7. Exit

I (1) Salg har man 3 valgmuligheder:

1. Tilbage
2. Start salg
3. Find salg

Vælger man (0), kommer man tilbage til hovedmenuen, (1) starter man et salg og vælger man (2) kan man finde et tidligere salg.

I (2) Lagerstyring har man 6 valgmuligheder:

1. Tilbage
2. Find vare
3. Opret vare
4. Opdater vare
5. Slet vare
6. Vis en liste over alle varer

(0) kommer man tilbage til hovedvinduet. (1) kan man finde en oprettet vare. (2) kan man oprette en vare til at kunne sælges. (3) kan man opdatere en allerede oprettet vare, hvis den f. eks. har ændret pris. (4) kan man slette en vare hvis den er udgået fra sortimentet. (5) viser en liste over varer som er til salg.

I (3) Kundehåndtering har man 7 valgmuligheder:

1. Tilbage
2. Opret kunde
3. Slet kunde
4. Opdater kunde
5. Tildel konto
6. Find kunde
7. Print liste over kunder

(0) kommer man tilbage til hovedmenuen. (1) kan man oprette en kunde. (2) kan man slette en kunde. (3) kan man opdatere en allerede oprettet kunde. (4) kan man tildele en kunde en konto. (5) kan man lede efter en kunde der er oprettet. (6) kan man blive vist en liste over oprettede kunder.

I (4) Medarbejderhåndtering har man 6 valgmuligheder:

1. Create employee
2. Delete employee
3. Update employee
4. Find employee
5. List all employees
6. Tilbage

(1) kan man oprette en medarbejder. (2) kan man slette en medarbejder. (3) kan man opdatere en medarbejder. (4) kan man finde en allerede oprettet medarbejder. (5) får man en liste over medarbejderne. (0) går man tilbage til hovedmenuen.

I (5) Udlejning har man 3 valgmuligheder:

1. Udlej Item
2. Aflever Item
3. Tilbage

(1) kan man udleje en vare. (2) kan man sætte den som afleveret. (3) går man tilbage til hovedmenuen.

I (6) Nyt udlejningsudstyr har man 3 valgmuligheder:

1. Nyt item
2. Fjern Item
3. Luk

(1) kan man oprette en ny udlejningsvare. (2) kan man slette en udlejningsvare. (3) går man tilbage til hovedmenuen.

I (0) Exit lukker man programmet.