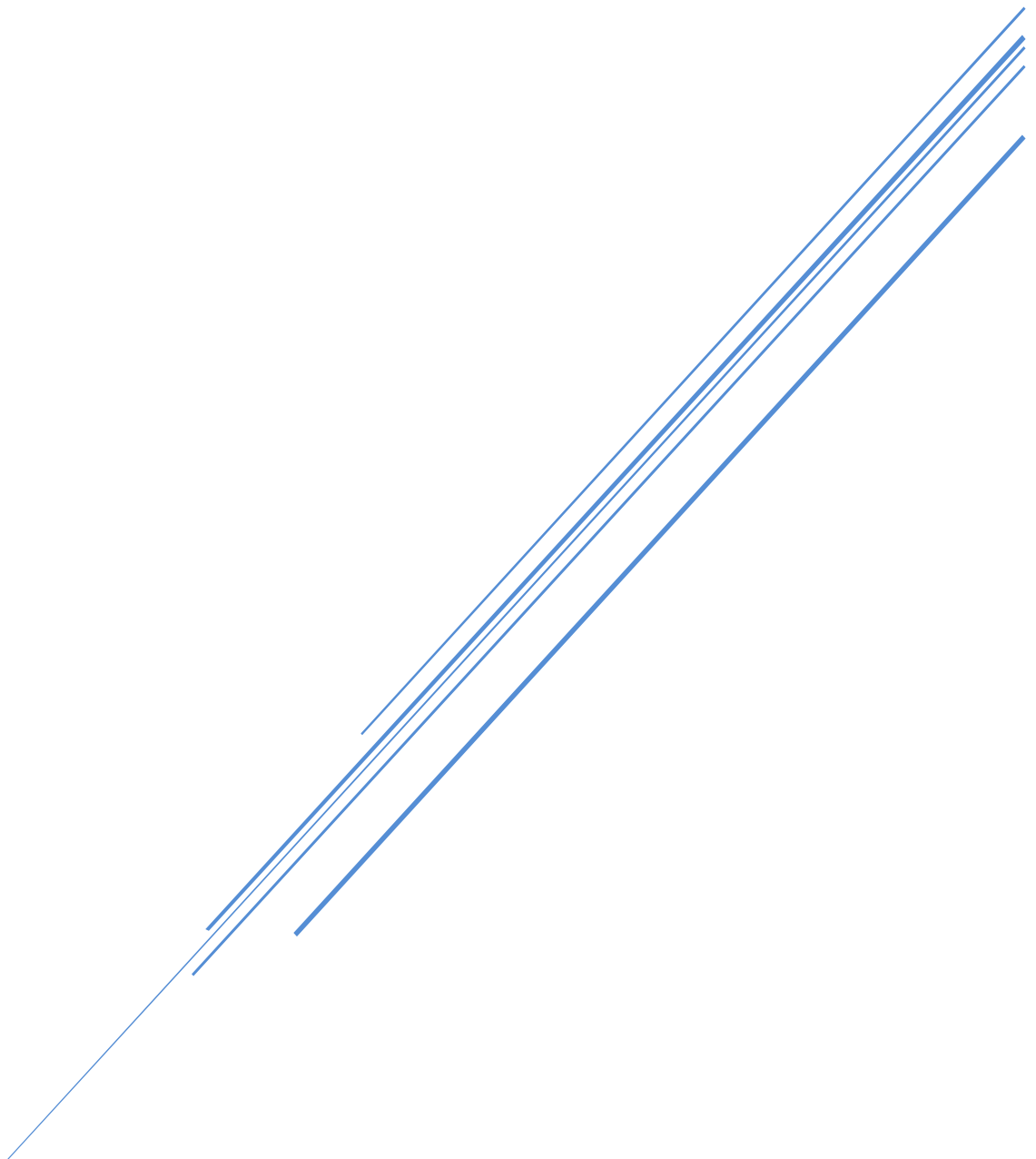


AUKTIONSHUS GRØN & OLSEN

ITA 4. Semester Projekt



Indhold

Målbillede for Auktionshus Grøn & Olsen	3
Formål	3
Motivation	3
Arkitektur Vision	4
IT-Principper og Kvaliteter	7
Egenskaber	8
Forretningsarkitektur	11
Foundation for Execution	11
Forretningskontekst, AS-IS	11
Forretningsprocesser	14
IT-Arkitektur egenskaber og principper	15
Governance & Management	17
Standarder	17
Brugerinddragelse	19
Brugerinddragelse i projektet	19
Interview format	19
AS-Is Brugerrejse: Analyse af den nuværende kundeoplevelse på platforme for online auktioner.	20
Interview med brugere: Identifikation af nuværende og potentielle kunders behov.	21
Design af Grænseflade	22
Hvorfor iterativ udviklings tilgang?	22
Fordele	22
Den iterative udvikling i brug	22
Overvejelser omkring POC-form, orientering og detaljeringsgrad	23
Refleksion og Konklusion	27
Refleksion over interviews	27
Refleksion mulige fremtidige design forberedelser i projektet	27
Evaluerings af PoC og dens rolle i projektets samlede løsning.	28
Konklusion på brugerinddragelsens bidrag til projektet.	28
Agil Udvikling	30
Valg af Agile Metoder og Processer	30
SCRUM Metodik	30
SCRUM Ceremonier:	30
Poker Planning	30

Sprint Planning	31
Daily Stand-ups.....	31
Sprint Reviews og Retrospectives.....	31
SCRUM-artefakter	31
Burndown Charts:	31
Extreme Programming (XP).....	32
Collective Ownership	32
Standardiseret Kodestruktur	33
Teststrategi	34
Test Driven Development (TDD)	35
API Tests i Postman	36
Konklusion af testresultater	36
Continuous Integration og Delivery (CI/CD)	36
Git Flow	36
Metodevalg og Tilpasninger.....	37
IT-arkitektur- og infrastruktur.....	39
Indledning	39
Designprocess (DDD)	39
Ønskede funktionaliteter	45
Drift	45
Logging.....	45
Hvorfor er logging godt?.....	46
Hvordan kan logging bruges i et forretningsmæssigt perspektiv?.....	46
Loki	46
Grafana.....	46
Aggregering	46
Docker.....	47
Authentication & JWT	48
Referencer	49
Bilag	52
Billeder brugt i Figma-modellen	57

Målbillede for Auktionshus Grøn & Olsen

Formål

Dette dokument har til formål at skabe en klar retning for Grøn & Olsens digitale transformation. Projektgruppen har udpeget en række hovedområder der vurderes vigtige at adressere, og slutproduktet er dette målbillede i version 1.0.

De udvalgte hovedområder er:

- **Identifikation og håndtering af eksisterende udfordringer:** Grøn & Olsens IT-infrastruktur er på nuværende tidspunkt relativt simpel, bestående af et økonomisystem og en ikke-specificeret måde hvorpå auktioner arkiveres. Med pres fra konkurrenter ønsker vi at analysere mulighederne for modernisering og digitalisering af så mange processer som muligt, så virksomheden kan følge med udviklingen i branchen
- **Tiltrække flere unge brugere:** Research viser, at der er et stort potentiale i kundegruppen 18 til 45 år, der traditionelt ikke har haft stor berøringsflade med auktionsindustrien. Der bør derfor implementeres funktioner og tjenester, der appellerer til denne aldersgruppe.
- **Bibeholdelse af den klassiske auktions-oplevelse:** Det anerkendes, at Grøn & Olsen har et eksisterende kundekartotek som er fortrolige med den fysiske auktionsform. Ambitionen er derfor på sigt at kunne køre auktioner både online og 'traditionelt', omend der i første omgang fokuseres på den nye online-platform

Gennem disse fokusområder vil Grøn & Olsen kunne tage de første skridt mod en digital transformation, hvilket vil styrke deres position i markedet.

Motivation

Projektbeskrivelsen afslører, at Grøn & Olsen oplever udfordringer i deres it-infrastruktur og forretningsprocesser. Deres hjemmeside er forældet, og der er næsten ingen brug af sociale medier i forbindelse med markedsføring. Begge dele hæmmer deres online synlighed, og

selvom auktionshuset tidligere har brændt sig i forsøg på digitalisering, er udviklingen i markedet nu så tydelig, at man ønsker at gøre forsøget igen. (bilag A).

Baseret på Grøn & Olsens egen SWOT-analyse, vurderer projektgruppen at den forestående transformation er en oplagt mulighed for at styrke virksomhedens nuværende, samt for at tilføje en række nye: (Bilag A)

- Udnyttelse af brugerdata til analyse og fastlæggelse af beslutningsprocesser
- Øget konkurrenceevne ifht. andre etablerede auktionshuse såsom Lauritz.com og DAB Auktioner
- Automatisering af organisationens mange manuelle processer
- Fremtidssikring af auktionshuset i form af en skalerbar løsning, så der på sigt kan udvides både nationalt og internationalt.

I de følgende afsnit præsenteres projektgruppens bud på hvordan en sådan transformation kan se ud, og for hvilke elementer der bliver nødvendige i det nye auktionssystem.

Arkitektur Vision

For at komme i mål med den ønskede transformation og derved opnå de ønskede forbedringer er der brug for en overordnet plan. I dette afsnit redegøres for den overordnede vision for systemet, herunder konkrete målsætninger, principper og kvaliteter. Strukturelt følges fase A fra TOGAS standarden, men da der ikke er fastlagt nogen egentlig forretningsstrategi som der kan tages udgangspunkt i bygger indholdet på informationer fra case oplægget og projektgruppens antagelser ud fra. (The Open Group, [n.d.]a)

Vision

På baggrund af Grøn & Olsens case oplæg, har projektgruppen udarbejdet en overordnet vision for projektet. Visionen fungerer som rettesnor for det videre arbejde i projektet, som sikrer at alle beslutningstagere har samme udgangspunkt, når der skal prioriteres, koordineres og planlægges.

Den foreløbige visions-sætning lyder således:

"En innovativ digital auktionsplatform, der muliggør bredere deltagelse og en mere engagerende auktion oplevelse for kunder, uanset geografisk placering eller tidspunkt, ved hjælp af en sikker, pålidelig og brugervenlig online infrastruktur"

I visionen udpeges en række vigtige kvaliteter for det kommende system:

- Det skal være sikkert, så brugerne kan stole på at deres data ikke bliver misbrugt eller stjålet.
- Det skal være brugervenligt, så platformen er intuitiv at anvende. Dette skal bl.a. gøres ved at sikre konsistens mellem andre af Grøn & Olsen løsninger og den nye platform. (Herunder print, logo, farver og fonte)
- Det skal være pålideligt, så Grøn & Olsen kan have tillid til, at auktionerne rent faktisk kan afholdes og at den enkelte brugers bud registreres.
- Til sidst skal systemet give anledning til at man kan deltage i auktioner, selvom man ikke befinder sig fysisk i Grøn & Olsens auktionshus i Aarhus.

Disse kvaliteter kan sidenhen anvendes når der skal opsættes mål for projektet, og den proof of concept leverance der skal afleveres efter første del af projektperioden.

Mål

Med afsæt i den førnævnte vision og motivation kan der formuleres en række mål. Herunder følger projektgruppens bud på konkrete målsætninger for den kommende løsning. Der tages udgangspunkt i opstillingen fra Medcoms 'Målbillede for meddelelse kommunikation'(Sundhedsdatastyrelsen, 2023), hvor hver identificerede målsætning sammenkobles med dennes ønskede gevinst:

ID	Målsætning	Gevinst
M.1	Alle Grøn & Olsens auktioner kan tilgås online	Fastholdelse af dedikerede stamkunder der er vant til fysiske auktioner, samtidigt med at der rækkes ud til en bredere brugergruppe via internettet

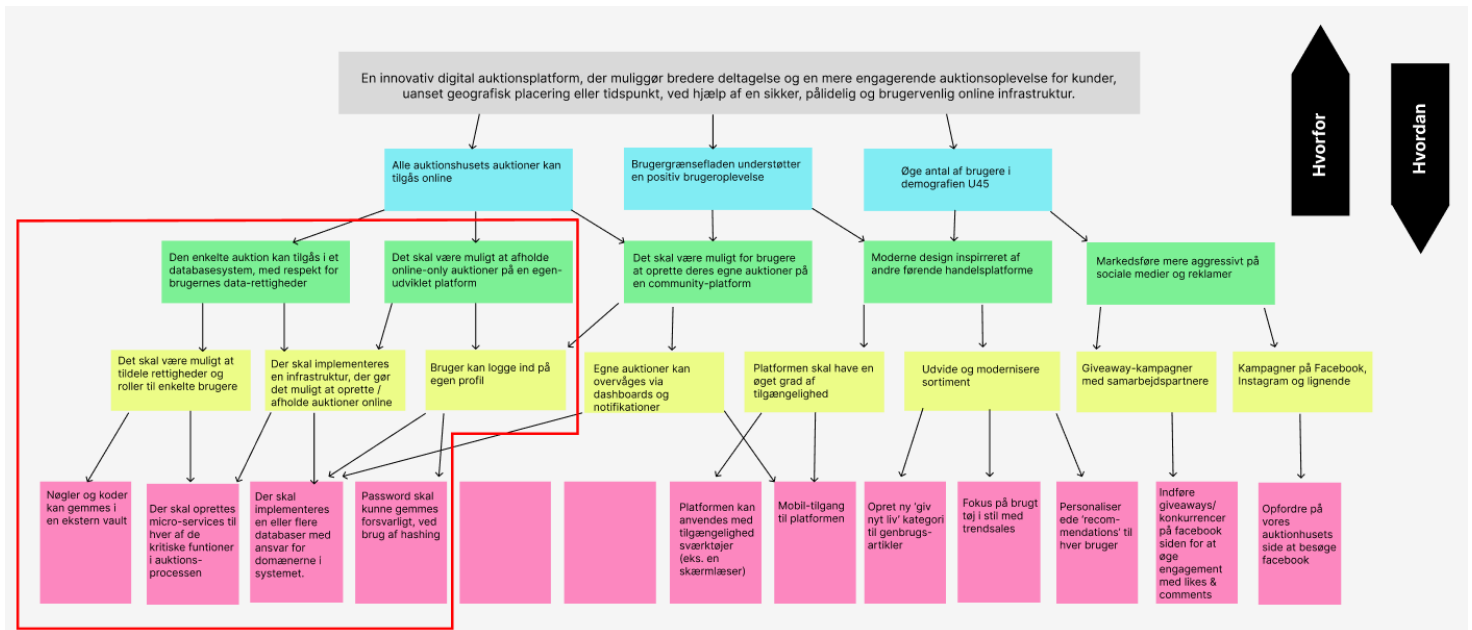
M.2	Der indsamles data i alle led af auktions processen, med henblik på rapportering og læring	Mulighed for kundeanalyse og kortlægning af adfærdsmønstre. Derudover på sigt mulighed for datasalg eller målrettede reklamer
M.3	Øget tilstedeværelse på sociale medier, med fokus på nye auktions-deltagere under 45 år	En bredere kundegruppe for mere budaktivitet. Derudover imødekommes tendenser fra udlandet, hvor unge har vist sig at være en målgruppe i vækst. [Henvis til kilde fra brugerinddragelse]
M.4	Moderne brugergrænseflade der er inspireret af lignende løsninger og overholder tidens design-standarder	Den kundesvendte del af platformen skal være så genkendelig som muligt så flest muligt kan interagere med den. Det leder igen til flere brugere, bud og højere hammerslag
M.5	Mulighed for skalering og udvidelse efter behov	Ved udvidelse til nye markeder kan der blive brug for flere ressourcer eller mere computerkraft. Jo nemmere dette er at tilføre, jo mindre bliver nede-tiden
M.6	Løsningen skal være sikret efter nuværende standarder.	Grøn & Olsens gode omdømme i markedet er værd at værne om. Et data eller sikkerhedsbrud ville være katastrofalt for dette, så det er vigtigt at vi gør alt hvad vi kan for at sikre brugernes data og platformens integritet.

Tabel 1: Målsætning for auktionssystemet

For at gøre det videre arbejde nemmere har projektgruppen valgt, at de ovenstående mål skal brydes ned i mindre bidder. Dette gøres både for at skabe overblik, og for at give en idé om, hvilke leverancer der skal til for at nå de enkelte mål.

Herunder ses et målhierarki, der indeholder visionen øverst. Ved at læse nedad i hierarkiet, besvares spørgsmålet "Hvordan opnår vi dette?". Læser man opad besvares spørgsmålet "Hvorfor implementeres dette"?

På den måde skabes der en rød tråd gennem målsætningerne indtil man nederst i hierarkiet har konkrete leverancer der skal stå klar, for at projektet får et solidt fundament.



Figur 1: Målhieraki. Mål indkranset af den røde boks indgår i POC leverancen

IT-Principper og Kvaliteter

Ifølge Dansk IT bør enhver virksomhed have konkrete IT-principper vedtaget i organisationen. Enten som del af en governance- eller overordnet it-strategi. (Fagrådet for Strategi og Ledelse, 2016)

Det har dog ikke været muligt i denne projektperiode at få indblik i disse hos Grøn & Olsen, så i stedet for er afsnittet herunder et udtryk for projektgruppens *forslag* til IT-principper og kvaliteter der skal indføres. Det vurderer projektgruppen til at være okay, dels fordi det ikke har været muligt at få indblik i de nuværende vedtægter, men i særdeleshed også fordi der er tale om en relativt stor transformation som naturligt vil føre forandringer med sig.

Egenskaber

For at fastlægge egentlige IT-principper der skal implementeres i virksomheden, kigger vi først, igen med afsæt i Medcom's målbillede, på de nøgleord der fremgår i visionen. Ud fra disse er det nemlig muligt at beskrive en række kvaliteter, som vi ønsker at systemet skal have. (Sundhedsdatastyrelsen, 2023)

Skalerbarhed	Vores oplæg til Grøn & Olsen lægger vægt på en fremtidig ekspansion udenfor Aarhus' bygrænse. Det vil på sigt betyde, at systemets ressourcer skal skaleres i takt med at flere brugere kommer til.
Sikkerhed	Ved omlægningen til online-auktioner vurderer projektgruppen, at der er afgørende for brugernes tillid og auktionshusets renommé, at løsningen er sikret med brugergenkendelse og adgangsbegrænsninger.
Pålidelighed	Der må ikke gå bud eller mod-bud tabt. Man skal som bruger kunne regne med at et validt bud gemmes, og at man får besked hvis et bud afvises. På samme måde skal auktionshuset kunne regne med, at al data gemmes i de korrekte systemer.
Brugervenlighed	Det vil være afgørende for brugernes adaption af den nye platform, at den er intuitiv at bruge. Herunder kommer mulighed for at anvende tilgængeligheds-værktøjer såsom skærmlæsere

Tabel 2: Egenskaber

Ud fra ovenstående vision, mål og egenskaber kan vi nu definere nogle konkrete IT-principper der skal implementeres i organisationen i forbindelse med systemets udvikling.

Princippernes form bygger på TOGAF 8 standarden, hvor hvert princip suppleres med rationale og implikationer. Dette er både for at sikre den bredest mulige forståelse, og for at forklare hvad der ligger til grund for implementeringen af det enkelte princip. (The Open Group, [n.d.]b)

ID	Princip	Rationale	Implikationer
P1	Alle processer, metoder og systemer designes med dataopsamling og anvendelse som fokuspunkt	Fremtiden er information. Ved at indsamle data i videst muligt omfang, kan vi blive ved med at forbedre vores metoder og vores forretningsmodel.	Et øget fokus på dataopsamling betyder, at kvaliteten af data bliver afgørende.
P2	Implementerede løsninger er fleksible og let skalerbare for at imødekomme ændringer i belastning og behov.	Det kan ikke vides præcis hvor hurtigt interessen for platformen vil vokse. Det kan derfor blive nødvendigt at tilføre flere ressourcer inden for kort tid.	Løsningen skal kunne vokse og krympe i henhold til ændringer i belastning og behov uden at påvirke ydeevnen negativt Der skal implementeres værktøjer til at overvåge systemets præstationer og belastning, så det kan skales proaktivt, når behovet opstår.
P3	Implementerede løsninger må ikke tilføjes features, der ikke direkte understøtter virksomhedens vision og målsætning	Udviklertimer er dyre, og det er vigtigt at sikre, at hver eneste feature der udvikles understøtter et forretningsbehov. På samme måde sikres det at der ikke opstår unødige komplikationer, gennem funktioner og metoder der aldrig rigtig anvendes.	Backloggen skal prioriteres baseret på værdien af de foreslåede features for virksomheden. Dette kunne for eksempel være gennem en vurdering op imod forretningsbehov og strategi.

P4	Den implementerede løsning skal være sikret, således den enkelte bruger kun har adgang til den data der er relevant for deres opgave	<p>En sikker løsning, der overholder regler og standarder for datasikkerhed (f.eks. GDPR), sikrer virksomhedens overholdelse af lovgivning</p> <p>Ved kun at give brugere adgang til de data, de har brug for, bliver systemet mere effektivt, da brugerne ikke skal filtrere unødvendige oplysninger.</p>	<p>Der skal etableres løbende kontrol og overvågning af brugeradgang og dataanvendelse for at identificere og afhjælpe sikkerhedsrisici.</p> <p>Der skal foretages en dataklassifikation for at identificere, hvilke data der er følsomme, og hvilke der er generelle, så sikkerhedskravene kan tilpasses derefter.</p>
P5	Monolitsstrukturer skal undgås og der lægges vægt på løsninger baseret på .NET microservices	<p>.NET indeholder adgange til mange frameworks og værktøjer der kan øge udviklingshastigheden</p> <p>Isolerede microservices betyder, at fejl i en service ikke nødvendigvis påvirker resten af systemet, hvilket øger robustheden og pålideligheden</p>	<p>Der skal etableres effektive processer og værktøjer til versionering og deployment af microservices for at sikre, at ændringer implementeres smidigt og sikkert</p> <p>Teams skal være mere autonome og ansvarlige for deres egne tjenester.</p> <p>Organisationen skal investere i passende værktøjer, såsom Docker, for at understøtte arbejdet med Microservices</p>

Tabel 3: IT-principper

Forretningsarkitektur

I dette afsnit præsenterer vi den strategiske plan for transformationen af Auktionshuset Grøn & Olsen.

Foundation for Execution

Målet er at definere den grundlæggende foundation for execution for virksomheden baseret på en dybdegående analyse af behovet for digitalisering. Foundation for execution defineres i bogen Enterprise Architecture as Strategy - Creating a Foundation for Execution som følgende:

"In short, a foundation for execution is the IT infrastructure and digitized business processes automating a company's core capabilities." (Ross, et al., 2006)

Dette er præcis, hvad vi ønsker, at Auktionshuset Grøn & Olsen skal opnå til slut, på baggrund af dette projektarbejde. Vi i projektgruppen mener, at en klar definition for og implementering af foundation for execution, vil være yderst gavnligt for den fremtidige udvikling og vision af virksomheden. Dette vil sikre, at virksomheden kan opretholde en konkurrencedygtig model, ved at automatisere og optimere kerneprocesserne efter overgangen til den udvidede digitalisering.

Forretningskontekst, AS-IS

For at kunne fastlægge virksomhedens foundation for execution og strategi, anbefales det, at virksomheden definerer en operating model:

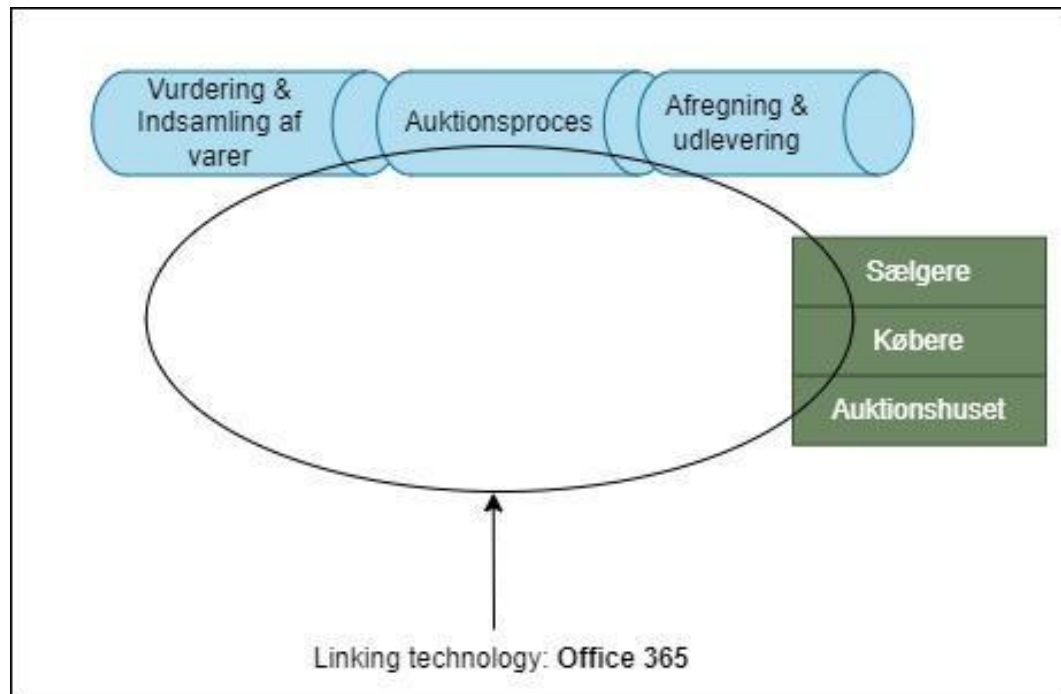
"To best support a company's strategy, we recommend that the company define an operating model. An operating model is the necessary level of business process integration and standardization for delivering goods and services to customers. An operating model describes how a company wants to thrive and grow". (Ross, et al., 2006)

Projektteamet har derfor undersøgt, hvilken operating model Auktionshuset Grøn & Olsen hidtil har opereret med. Denne undersøgelse bygger på en række antagelser som projektteamet har udarbejdet på baggrund af den udleverede projektbeskrivelse: (Bilag A)

	Antagelse	Baggrund
1	Lav standardisering	Grøn & Olsen har én lokalitet, hvorfra alle auktioner afholdes fysisk. Det vurderes derfor at der er stor frihed i den enkelte afdeling til at fastlægge processer og arbejdsmetoder, da de ikke skal replikeres andre steder.
2	Nuværende system: Office365	Der nævnes et økonomisystem, men ikke en måde hvorpå auktionshuset kan holde styr på kunder, auktioner og lager. Da der er tale om en relativt lille virksomhed, vurderes det derfor, at Office365 er et realistisk bud ud fra økonomi og kompleksitet.
3	Høj Business Process Integration	På trods af de enkelte afdelingers autonomitet, er der behov for en klar process for, hvilken rækkefølge aktiviteter skal foretages i. Baseret på antagelse 1 og 2 vurderes det derfor realistisk, at auktioner registreres på en form for fælles-drev, hvor den enkelte afdeling kan lave opdateringer når det bliver deres tur. Dermed arbejder alle i samme datakilde, og kan se status fra start til slut.

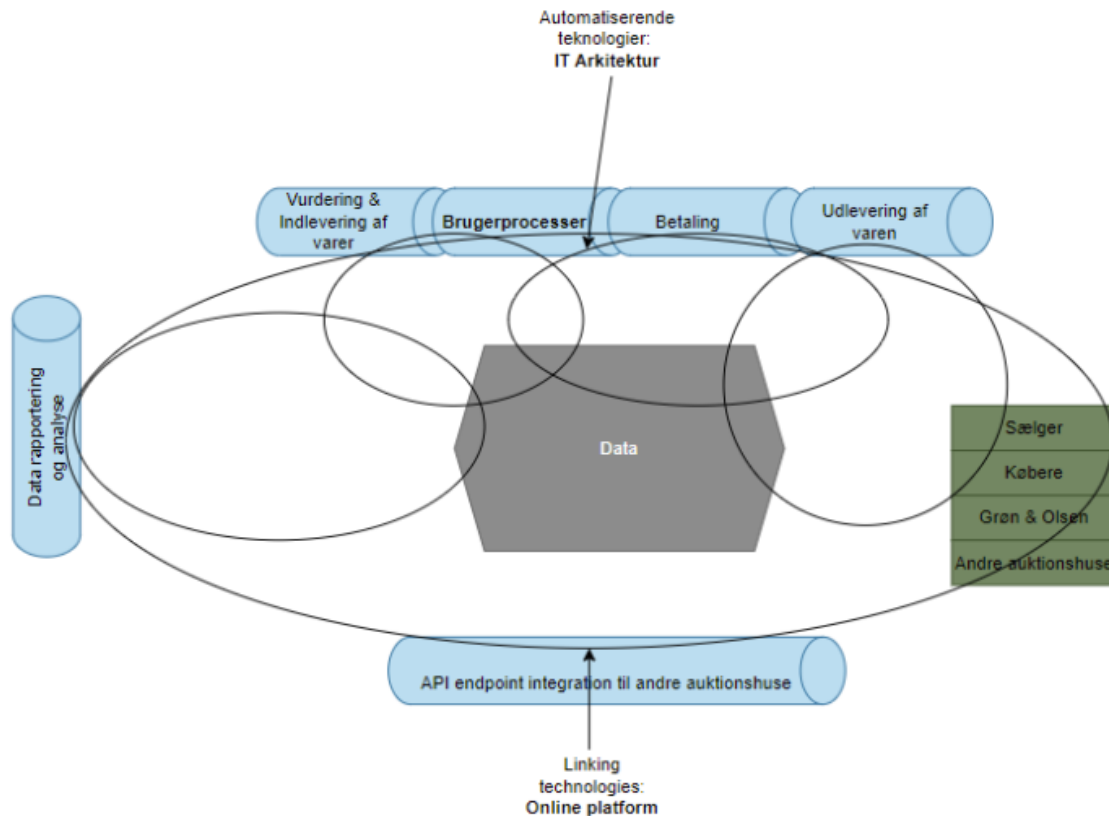
Tabel 4: Antagelser om Grøn & Olsen i forbindelse med udarbejdelse af Operating Model

I den nuværende tilstand (AS-IS) opererer auktionshuset Grøn & Olsen primært i en Coordination operating model. Dette indebærer en centraliseret styring af deres fysiske auktioner fra deres lokation i Aarhus. Der er behov for en høj grad af intern koordination mellem afdelingerne, hvilket er nødvendigt for at sikre en smidig daglig drift. Imens lider virksomheden under en lav grad af standardisering af forretningsprocesser, idet hver afdeling opererer efter deres egne metoder, hvilket er en stor begrænsning af effektiviteten på tværs af virksomheden. I nedenstående figur ses et mock-up af et Core Diagram AS-IS, der illustrerer de forskellige processer og operatører i auktionshuset, som er omgivet af et Office 365 system, der forbinder de forskellige afdelinger.



Figur 2 - AS-IS Core Diagram

I den kommende tilstand (TO-BE) vil auktionshuset Grøn & Olsen gennemgå betydelige ændringer i deres operating model og forretningsprocesser. Med den nye online platform vil Core Diagrammet blive omformet og udvidet for at afspejle virksomhedens fornyede fokus og udvidede kapabiliteter som online auktioner, dataanalyse, og automatisering af processer. Denne transformation sigter mod at løse de udfordringer, der er identificeret ved brugen af den nuværende operating model og at skabe en mere effektiv, standardiseret og integreret forretningsstruktur.



Figur 3 - TO - BE Core Diagram

Det ses af den nye model, at det i forbindelse med overgangen til det nye system er nødvendigt at justere den overordnede strategi. Der skal opbygges kapabiliteter til at automatisere processer, organisere data og afholde auktioner uden fysisk tilstedeværelse. Dette vil også være det reviderede Foundation for Execution, som vi gerne vil pålægge Auktionshuset Grøn & Olsen gennem vores online platform.

For at dette kan blive en realitet, vil det kræve en øget standardisering hos Grøn & Olsens nuværende forretningsprocesser. Dette vil også betyde, at Grøn & Olsen kommer til at bevæge sig fra en Coordination operating model til en Unification operating model, hvilket på lang sigt vil styrke virksomhedens effektivitet og konkurrenceevne ved at skabe en mere strømlinet forretningsstruktur.

Forretningsprocesser

Som tidligere nævnt vil der i forbindelse med implementeringen af det online auktionssystem være behov for ændringer i virksomhedens processer. I afsnittet herunder redegør vi for ændringen i et konkret eksempel, afholdelsen af en auktion. Andre processer, såsom

indlevering, vurdering eller afsendelse er ikke med i dette eksempel. De to processer findes under bilag hvor AS-IS findes i bilag B og TO-BE findes i bilag C.

TO-BE (Bilag C)

I den nye proces tilføjes en ny aktør i form af 'system' som skal stå for den foreslåede automatisering. Denne swimlane består af en række opgaver som tidligere har ligget hos auktionarius, herunder opstart af auktion, registrering af bud og notifikation af auktionens vinder. Det betyder at hele bud-processen kan laves online, så kunder kan deltage uden at skulle være i nærheden af auktionshuset eller en telefon. Kunden har stadig ansvar for at afgive bud, men er nu mere fleksibel i forhold til hvordan dette gøres.

Ved at digitalisere en stor del af auktionen får Grøn & Olsen også mulighed for at indsamle mere data end før. Der kan laves analyser på gennemsnitlige bud-hop, kundegruppers købsmønstre og effektiviteten af diverse markedsføringskampagner.

Andre processer i auktionshuset vil skulle omgå en lignende forvandling, men det er vurderet som uden for scope for denne version af målbilledet.

IT-Arkitektur egenskaber og principper

I de ovenstående afsnit er det beskrevet, hvordan projektgruppen ønsker at et fremtidigt system skal opereres. Herunder hvilke principper og egenskaber der er centrale for den endelige løsning. For at realisere dette system er det derfor nødvendigt at opbygge en teknisk platform, som kan facilitere netop disse krav og principper.

Da det ønskes at opnå størst mulig skalerbarhed, pålidelighed og sikkerhed, vurderes det at en 'Service Oriented Architecture' (SOA) vil være et solidt fundament for det videre tekniske arbejde. SOA beskrives af Sam Newman som et opgør med fortidens monolit-strukturer, som nu gør det muligt at bryde et system op i mindre dele:

"SOA emerged as an approach to combat the challenges of large, monolithic applications. This approach aims to promote the reusability of software; two or more end user applications, for example, could use the same services. SOA aims to make it easier to maintain or rewrite software, as theoretically we can replace one service with another without anyone knowing, as long as the semantics of the service don't change too much."
(Newman, 2021)

Med andre ord kan vi altså lave vedligehold og udskiftninger i dele af systemet, uden at det vil være nødvendigt at lukke det hele ned. Dette kan bidrage til bedre oppe-tid mens muligheden for at udskifte eller duplikere en enkelt service vil hjælpe os til en større skalerbarhed.

Det er tidligere nævnt at vi i udarbejdelsen af dette projekt har taget udgangspunkt i TOGAF-standarden, blandt andet i forbindelse med formuleringen af de indledende principper. Dette holdes der fast i, når der herunder fastlægges arkitekturprincipper. Disse er nemlig udvalgt fra en liste, som er en del af netop TOGAF. (The Open Group, [n.d.])b)

	Princip	Begrundelse
1	Data is an asset	Projektgruppen ser store muligheder for opsamling og brug af data i fremtiden, og det er vigtigt at man tilskriver sin data den rette status. Ved at se sin data som et aktiv i stedet for blot et analyseværktøj, bliver det nemmere at anvende aktivt. For eksempel i forbindelse med markedsføring, mersalg eller decideret salg af den opsamlede data.
2	Data is accessible	Det er et grundvilkår for systemet et det hele tiden har adgang til den relevante data. For eksempel duer det ikke at man vil afgive et bud, men at systemet så pludselig ikke kan se bud historikken og vurdere om man har budt over. Derfor skal der skabes en løsning, hvor kritisk data gemmes så tæt på de elementer, der skal anvende den.
	Requirements-based change	Som vi også har defineret i tabel 3, P3 er det vigtigt at sikre løsningen mod 'feature-creep' og spildte ressourcer. Derfor kan implementation af nye funktionaliteter kun ske på baggrund af dokumenterede forretningsbehov. Dette hjælper også med at opretholde en vis pålidelighed i løsningen.
	Interoperability	Eftersom SOA er valgt som grundstenen i projektet, er det vigtigt at de enkelte services kan kommunikere effektivt med hinanden. Dette vil også sikre at del-elementer i løsningen nemmere kan skiftes ud, og at det enkelte element kan håndtere sin egen del af forretningslogikken

Tabel 5: Arkitekturprincipper

Governance & Management

Standarder

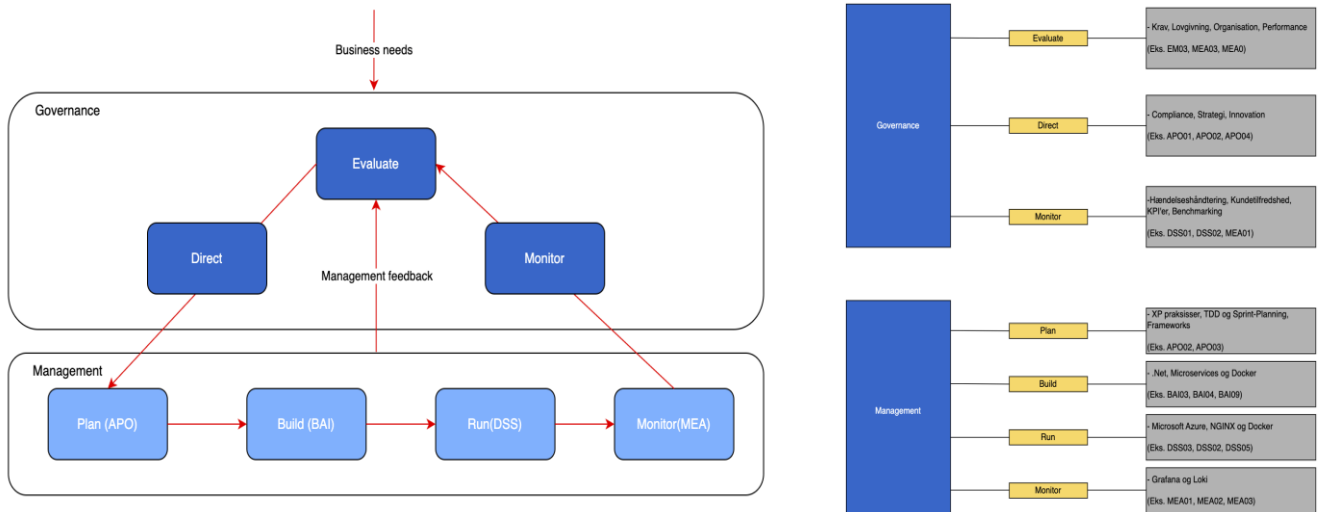
GDPR

Når projektgruppen arbejder på det nye IT-system, er det vigtigt for projektgruppen, at sikre brugernes og medarbejdernes personoplysninger bliver godt beskyttet mod udefrakommende adgang. For at projektgruppen kan opnå målet, har projektgruppen valgt at implementere nogle sikkerhedsforanstaltninger. Projektgruppen har valgt at indsamle data på en transparent måde, hvor man vælger, at informerer brugerne om, hvordan deres data kan blive indsamlet og hvad det vil blive brugt til. Dermed giver det kunderne og medarbejdere en sikkerhed, når de tilmelder sig hos Grøn og Olsen, og giver dermed en tryghed, når de vælger at dele deres oplysninger og give deres samtykke til Grøn og Olsen. (GDPR.DK, 2021)

Cobit

Vi ønsker Auktionenhuset Grøn og Olsen vælger at bruge principper fra COBIT 5. Cobit 5 består af en masse processer, som består af ISO-standarder, ud fra vores 'figur 4 - Overview of Cobit of Auktionenhuset Grøn og Olsen', kan vi gøre brug af de her processer til at forbedre deres IT-styring og sikre, at deres IT hjælper med at understøtte organisationens mål og behov. Da de lever op til forskellige best practice rammer og standarder, såsom ITIL, ISO 20000 og ISO 27001. De måder som de kan anvende Cobit på er, at man kan bruge det til at identificere nogle klare mål for deres IT-processer. Dette omfatter alt fra kontrolmål, der er relateret til datasikkerhed, systemets tilgængelighed og bruger adgangsstyring. Den er blandt andet med til at hjælpe med at identificere og vurdere risici relateret til deres IT-infrastruktur og processer. Derudover skal de have udviklet og implementeret klare politikker og procedurer overfor IT-styring såsom for eksempel datasikkerhed, bruger adgangsstyring, ændringsstyring og incident management. De skal etablere overvågnings- og evalueringssystem for deres IT-processer. Derudover skal auktionenhuset opfordres til en kontinuerlig forbedring inden for IT-styring ved at opfordre til løbende evaluering og optimering af processer og kontroller. (ITGovernance, [n.d.])

Ved at anvende COBIT, kan Grøn og Olsen auktionshus opnå større transparens, effektivitet og styring af deres IT-aktiviteter, hvilket kan bidrage til at minimere risici og sikre en pålidelig og sikker drift af deres auktionsplatform. (RISMA, [n.d.])



Figur 4 - Overview of Cobit of Auktionshuset Grøn og Olsen.

Brugerinddragelse

Brugerinddragelse i projektet.

Gruppen har afholdt 8 interviews med potentielle brugere inden for den ønskede demografi. Ud fra det har vi lavet detaljerede user stories og analyseret brugernes pains and gains med en AS-IS-skitse.

Interview format

I vores undersøgelse gennemførte vi en række strukturerede interviews for at indsamle indsigt i brugeroplevelser med online auktion platforme. Vores tilgang var styret af en interviewramme, som vi udviklede som det første. Formålet med strukturen er at fremkalde feedback fra deltagerne på forskellige stadier af deres engagement med auktionssystemer og at vedligeholde en systematik, så vi fik samme informationer fra flere forskellige interviewpersoner. Vi testede opsætningen gennem et demo-interview internt i gruppen, for at sikre at tidsestimatet var korrekt og at spørgsmålene føltes naturlige. Det var med til at hjælpe gruppen, da alle havde en idé om, hvordan interviewet kunne se ud, før medlemmerne tog ud og afholdt deres egne.

Start af Interview:

Vi startede med brede, åbne spørgsmål for at etablere en grundlæggende forståelse af deltagernes tidligere erfaringer med online shopping og auktioner, samt for at opvarme interviewdeltageren. (Anne-Sofie Thomsen, 2021)

Spørgsmål som "Hvad er det bedste køb, du nogensinde har lavet?" og "Kan du nævne platforme, du har handlet på i fortiden?" hjalp os med at vurdere deres kendskab til forskellige e-handelsmiljøer samtidig med at vi udforskede deres opfattelser og stigma forbundet med auktioner. For eksempel gav spørgsmålet "Hvilke mennesker tror du deltager i auktioner?" utrolig stor gavn, da vi opdagede et mønster med gentagende stigma omkring alder og social baggrund. (Bilag - interview opsætning)

Midten af Interviews:

For brugerne, der havde brugt en auktionsplatforme før, kunne vi stille mere specifikke spørgsmål, her begyndte vi at kortlægge deres brugerrejse (Anne-Sofie Thomsen, 2021). Vi spurgte kronologisk om, hvordan de fandt frem til platformen, hvilken platform de brugte, og hvad de købte. For de deltagere, der ikke havde anvendt en auktion, kompenserede vi

ved at spørge ind til deres anvendelse af kendte sider som DBA eller Facebook Marketplace, hvor vi undersøgte, om de havde solgt eller købt produkter før. Vi lavede en række spørgsmål specifikt til dem, der havde deltaget i en auktion, og et sekundært sæt spørgsmål til dem, der havde erfaring med de andre sider, hvilket gav os en bredere indsigt i brugernes forskellige handelsoplevelser. (Bilag - Interview struktur)

Slut Interview:

For dem, der havde deltaget i auktioner før, både de online men også de fysiske, fokuserede vi på deres oplevelser efter auktionen. Vi spurgte: "Hvad tænkte du, så snart der var hammerslag?" og "Hvordan var leveringsprocessen, hvis du havde vundet en vare på auktion?". Dette hjalp os med at forstå de følelsesmæssige aspekter af deres oplevelser og eventuelle bekymringer eller positive følelser. Vi afsluttede alle interviews med "har du andet du vil tilføje?", til hvis de har en sidste point de vil nå at få med. (Anne-Sofie Thomsen, 2021)

Ved at have aftalt interview strukturen på forhånd, kunne vi holde en kontinuerlig samtale og følge en rød tråd igennem hele interviewet. Det betød for os, at vi kunne indsamle indsigt fra brugere der havde haft en konkret oplevelse med online auktioner, og brugere som af den ene eller anden årsag havde fravalgt dem. Denne tilgang gav os en god forståelse af brugernes behov gennem deres oplevelser med forskellige platforme, og det blev klart at der er en række ting som afholdte visse af vores deltagere fra i det hele taget at overveje auktioner. Disse pains and gains, kunne vi bruge til at forbedre og streamline vores designproces samt sikre, at vi tog højde for en bred brugergruppes præferencer og udfordringer.

AS-Is Brugerrejse: Analyse af den nuværende kundeoplevelse på platforme for online auktioner.

I forbindelse med udviklingen af vores auktionsplatform, har vi skitseret en brugerrejse på baggrund af vores interviews. Dette er en fiktiv brugerrejse (AS-IS) for et auktionshus, lavet for at berøre alle touchpoints fra vores interviews. Her kom vi igennem en del pains and gains, som var aktuel når man designer et auktionshus.

Interview med brugere: Identifikation af nuværende og potentielle kunders behov.

Eftersom alle deltagerne i undersøgelsen er anonyme, har vi valgt at referere til dem med pseudonymer for at beskytte deres identiteter. Vi vil i denne rapport derfor referere til deltagerne som ID1-ID8 (Interviewdeltager 1-8).

Et centralt behov, der fremtrådte, var brugernes ønske om større sikkerhed og troværdighed i auktions-processen, hvilket blev fremhævet fra flere af deltagere, som f.eks. ID1, der nævnte behovet for at man skal forsikres, når et bud er lagt, da der typisk er en manglende bekræftelse, så der ikke er tvivl for brugeren om, at deres bud er gået igennem. Det gjorde personen usikker på at man kunne stole på hjemmesiden (*ID1, 12:21*). Dette går hånd i hånd med et andet væsentligt pain point, hvilket var det outdated layout og brugergrænseflade på mange eksisterende auktionsplatforme.

ID2 påpegede, at platforme, der ser gamle ud, og ikke følger den interaktive og "flotte" standard, som man forventer fra nye og smarte e-handels hjemmesider, kan skabe tvivl om platformens troværdighed (*ID2, 07:00*). Oplevelsen bliver mindre attraktiv for yngre brugere, som ID3 også nævner. Dette understøtter behovet for en mere moderne og brugervenlig grænseflade.

Yderligere peger brugerne på vigtigheden af detaljeret produktinformation for at kunne træffe informerede købsbeslutninger, hvilket ID4 understregede med sit ønske om mere detaljerede produktbeskrivelser (*ID4, s. 3*). Dermed kommer der klare forventninger til auktionshusets standardprocedure ved oprettelse af varer. En god pointe blev også nævnt af ID1, at forholdet mellem auktionstid og aktionsværdi skulle følge hinanden (*ID1, 20:29*). Gains, som brugerne oplevede, inkluderede følelsen af tilfredsstillelse og spænding ved at vinde en auktion eller bare det at deltage i en auktion. Dette blev fremhævet af flere deltagere. Derudover værdsatte brugerne notifikationer direkte på hjemmesiden i stedet for kun via e-mail, hvilket ID2 fremhævede som en forbedring, der kunne øge bruger engagementet (*ID2, 16:14*).

Hans pain fra hans egen oplevelse var, at han skulle refresh både platformen for at se, om der var nye bud, og sin e-mail for at se, om nogen havde budt over ham. Han forudser, at

man helt klart ville kunne imødekomme den pain og stress for brugeren, hvis alt den information og notifikationer brugeren havde behov for, altid var til stede inde på platformen.

Ved at have disse fokuspunkter in mente, når vi designede vores PoC, har vi kunnet bearbejde de funktionelle og emotionelle behov hos vores brugere. Resultatet er en mere brugercentreret og engagerende auktion oplevelse. hvilket i sidste ende vil øge brugertilfredsheden og den generelle brugerrejse.

Design af Grænseflade

Hvorfor iterativ udviklings tilgang?

Vores POC er bearbejdet i en iterativ proces, hvor løbende krav og ændringer kunne implementeres. Disse krav og ændringer kom fra vores desk research, interviews og teori.

Fordele

Den iterative design process har muliggjort evnen til at indføre løbende forbedringer. De forbedringer er fremmet af nye opdagelser af brugerkrav, udvikling af brugernes behov samt deres pains og gains. Det har medført en langt hurtigere responstid, det at starte helt forfra, samt identifikation af problemer end en traditionel tilgang ville.

Den iterative fremgangsmåde har også til fordel at imødekomme krav der kunne komme i fremtiden fra styregrupper eller Grøn og Olsen eller myndighederne. Som sagt, er vores produkt et PoC, så det er forventet at komme fremtidige ændringer eller krav.

Den iterative udvikling i brug

PoC blev konstant justeret gennem flere iterationer, med alt fra små ændringer af farve eller skriftstørrelse, til ændringen af funktionaliteter og sletningen eller oprettelsen af faner. Det overordnede design af hele grænsefladen blev lavet vha. vores interviews, brugerrejse og desk research. Vi har løbende kigget tilbage på vores research, vores målbillede og interviews for at implementere disse iterativt. Man kan hurtigt godt glemme alle disse punkter man først fandt frem til i f.eks. Desk research'en, så den iterative arbejdsmetode har hjulpet en del her.

For eksempel, i den første iteration var produktbeskrivelser og alle detaljer om vare, synligt på varelistens fane. Det blev klart for projektgruppen, da vi skulle dokumentere vores AS-IS brugerrejse, at vi havde overset ID4's (ID4, s. 3) vigtige pointe og behov, for at have en detaljeret produktbeskrivelse og at jo flere detaljer der var, jo bedre. Dette startede endnu en iterativ process, hvor vi oprettede en dedikeret fane, "Produktside" (*Figma*). En fordel ved at vi arbejdede iterative her, var at vi blot kunne tilføje denne fane kollektionen og lave minimale ændringer ved "Oversigt over vare" (*Figma*), fremfor at oprette et helt nyt sketch, som den traditionelle tilgang ville have krævet. Vi har raffineret vores design frem for at redesign, gennem rapid prototyping. (Labs, 2020)

Dette tillod brugerne at se flere billeder og detaljer uden at blive overvældet af information på hovedsiden. Denne beslutning blev også understøttet igen, da vi gennemgik vores interviews, og både ID3 og ID2 havde argumenteret, at simple, intuitive, moderne og letforståelige hjemmesider var mere brugervenlige.

Denne iterative udviklingsproces fremmede en naturlig tilgang, da spontan kreativitet og ide generation hurtigt og let kunne implementeres. Et andet eksempel på dette, var da projektgruppen browsede nogle forskellige salgsplatforme, og indså at vores grænseflade manglede noget meget basalt. Søgefelt og filtreringsmuligheder.

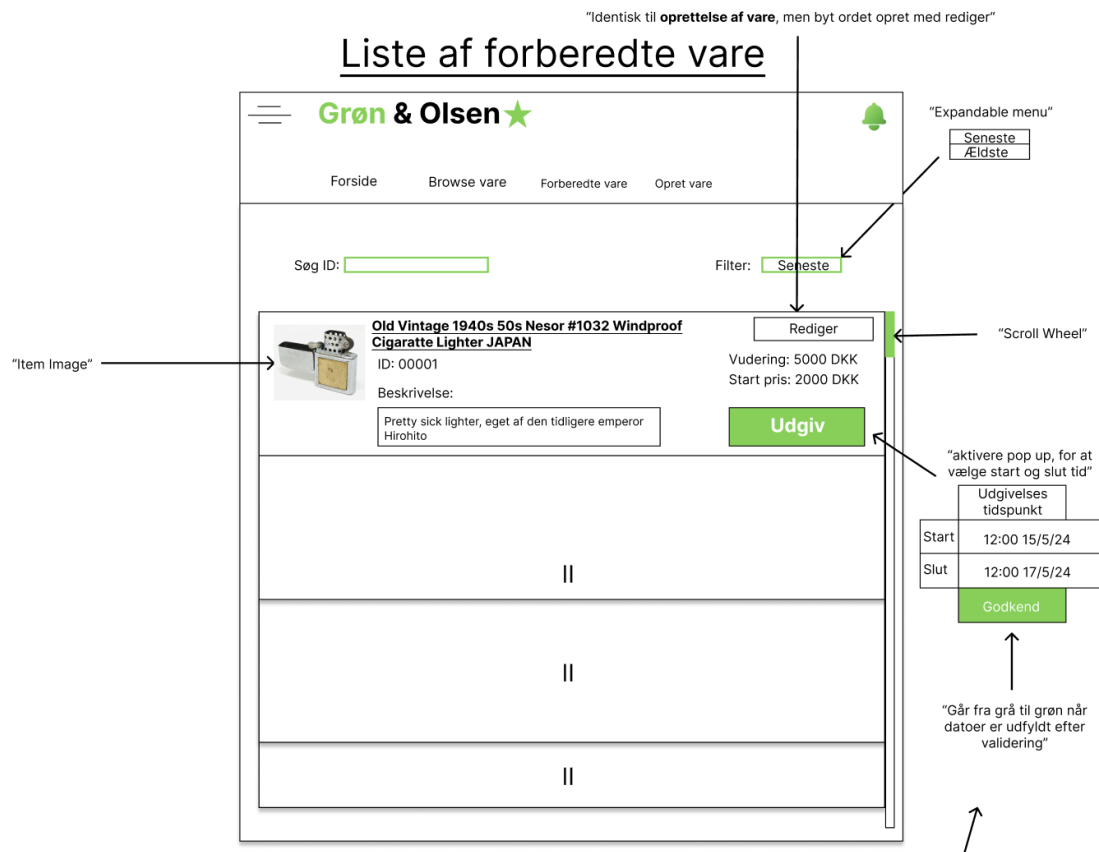
Overvejelser omkring POC-form, orientering og detaljeringsgrad.

Vi udviklede en detaljeret prototype med interaktive elementer som knapper, pop-ups og navigationsmenuer. For at illustrere interaktiviteten inkluderer vores *Figma*-design visuelle pile og tekster, der forklarer funktionerne af forskellige features vi vil have implementeret og hvordan de skal fungere. Disse visuelle hjælpemidler gør det lettere for vores interessenter at forstå og interagere med grænsefladen.

Gennem de afholdte interviews med potentielle brugere, har vi kunnet indsamle værdifulde indsigter om deres pains and gains. Dette har gjort os i stand til at designe en grænseflade, der ikke blot opfylder deres eksplicite behov, men også imødekommer deres implicite ønsker og forventninger. (Brown, 2008)

Et konkret eksempel på dette er under fanen "liste af forberedte varer" (*Figma*), som specifikt admins har adgang til. Hvert produkt på listen har sin egen knap med titlen "udgiv". Når denne knap trykkes, kommer der en pop-up, hvor admin skal vælge et udgivelsestidspunkt,

inklusive en start- og slutdato. Når de to validerede tidspunkter er valgt, bliver knappen "godkend" funktionel og grøn for at indikere til admin brugeren, at deres input er validt. Indtil da er knappen grå og ikke funktionel, hvilket forhindrer fejl og sikrer, at alle nødvendige oplysninger er så korrekte som muligt før udgivelsen. Ved hjælp af Jenny Davis' affordance of mechanisms-analyse kan vi forstå, hvordan disse designelementer fungerer i praksis. Se analysen af udgivelsestidspunkt, pop-up, funktionalitet på Figma, "Liste af forberedte vare"



Figur 5: Figma – Liste af forberedte vare

Jenny Davis nævner blandt andet de fire begreber 'Anmode', 'Kræve', 'Opfordre' og 'Tillade' som vigtige elementer af en designproces. De er en del af det domæne, hun kalder for affordance mekanismer, som beskriver hvordan et objekt eller en grænseflade kan anvendes. Altså, hvad kræver et objekt af mig som bruger, hvad anmoder det om, hvad opfordrer det til og hvad tillader det? (Davis, 2020)

Denne tilgang, med brug af affordance overvejelser og analyser, sikrer at administratorer og brugere nemt kan navigere grænsefladen og træffe korrekte beslutninger uden unødigt kompleksitet. Ved at integrere vores læring fra analysen af disse mekanismer har vi forsøgt at skabe en mere intuitiv og brugervenlig grænseflade, der reagerer på brugernes handlinger og behov. Dette reducerer muligheden for fejl, hvilket forbedrer den samlede

brugeroplevelse og øger platformens troværdighed i sidste ende.

Disse begreber har vi forsøgt at koble til vores POC, da vi har haft som krav at interfacet skal være 'intuitivt'. Det kan være en relativt subjektiv vurdering, men succeskriteriet i dette PoC bliver, at den overholder Davis' teorier om affordance. Specifikt for listen af forberedte varer er vores overvejelser som følger:

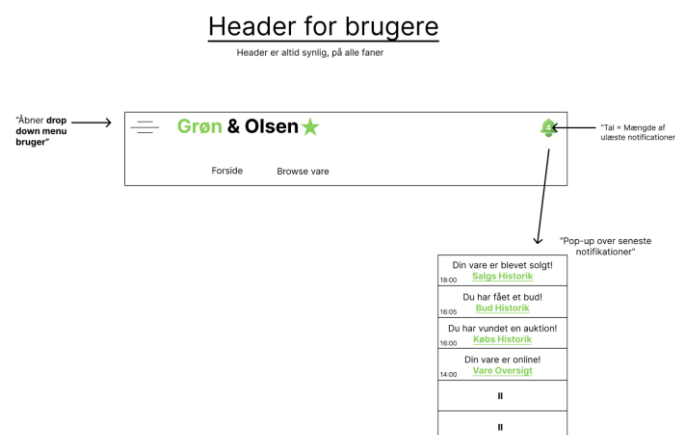
Anmode: Systemet anmoder om specifikke input fra brugeren ved at præsentere pop-up'en, hvor start- og slutdatoer skal indtastes

Kræve: Kravet bliver tydeligt gennem den ikke-funktionelle grå "godkend" knap, som ikke kan trykkes før validerede tidspunkter er indtastet. I fremtidige iterationer vil der komme en rød kant rundt om de manglende informationer, for yderligere at tydeliggøre kravet.

Opfordre: Når start- og slutdatoerne er korrekte, opfordrer den grønne "godkend" knap brugeren til at fortsætte, hvilket giver en visuel bekræftelse af korrekt input

Tillade: Når knappen bliver grøn og funktionel, tillader systemet brugeren at fortsætte med processen, hvilket mindsker risikoen for menneskelige fejl.

Vi har ligeledes lavet disse overvejelser for headeren på den brugervendte del af systemet. Her må man gå ud fra at der er større chance for at der kommer førstegangsbbrugere ind, og det er derfor ekstra vigtigt at grænsefladen tydeligt kommunikerer muligheder og begrænsninger.



Figur 6: Header og notifikations-menu på den brugervendte side

Anmode: Systemet anmoder om brugerens opmærksomhed ved at vise et nummer på notifikationsklokken, som indikerer ulæste notifikationer. Dette opfordrer brugeren til at klikke på ikonet for at se detaljer.

Kræve: Kravet om brugerens handling er tydeligt gennem det visuelle signal på notifikationsklokken. Brugeren skal klikke på klokkeikonet for at få adgang til de fulde detaljer om notifikationerne. Kravet om opmærksomhed kan på sigt udbygges ved at tilføje en animations-effekt.

Opfordre: Når brugeren klikker på notifikationsklokken, opfordrer pop-up'en med en liste over notifikationer brugeren til at interagere med de specifikke beskeder. Hver notifikation indeholder et hyperlink til den relevante side, hvilket gør det nemt for brugeren at navigere direkte til det område, der kræver opmærksomhed.

Tillade: Systemet tillader brugeren at følge op på notifikationerne ved at give direkte links og klare beskeder, hvilket mindsker behovet for at søge manuelt efter information. Dette skaber en mere strømlinet og effektiv brugeroplevelse.

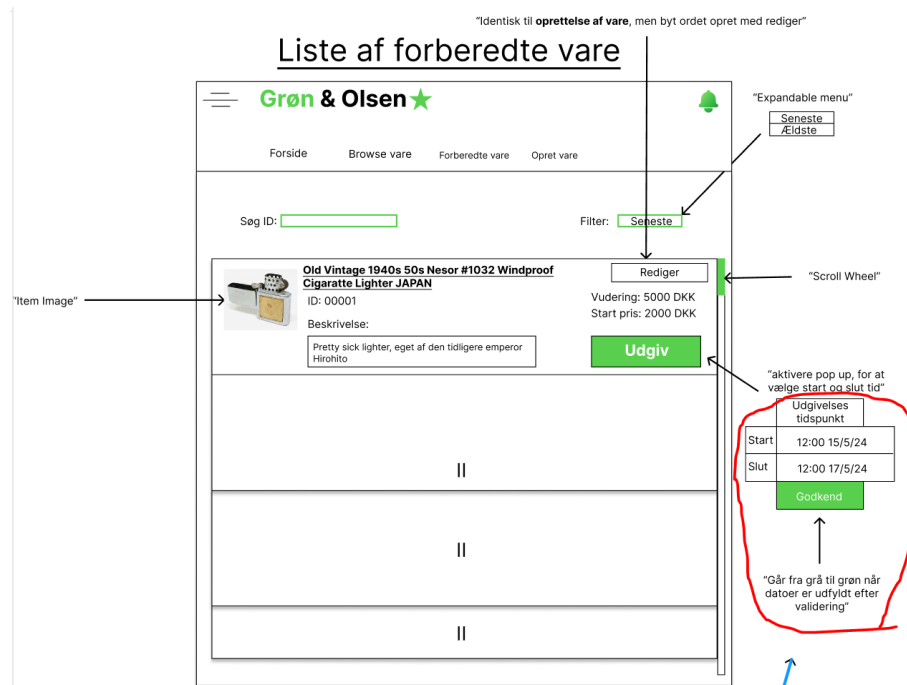
Denne analyse har også gavnet vores design process, ved at vide hvordan vores brugere integrere med notifications systemet, har det hjulpet os med at udvikle en funktionalitet der direkte tackle det problem som vi blev opmærksomme på ved analysen af ID2's interview, men også en pain der blev nævnt af ID3 (*ID3, 08:46*), at han havde prøvet at bruge en auktionsside hvor han kun fik at vide hvis han blev overbudt over email.

I header og notifikations-menu'en har vi også haft usability heuristik nr. 4 in mente "Consistency and standards". Header fremgår på alle sider, som gør det nemmere for brugeren at navigerer, hvilket ifølge NNGroup, gør brugernes liv nemmere.

"The fourth of Jakob Nielsen's ten heuristics — consistency and standards — is key to creating applications that make sense for users. [...] and that make users' lives easier".

(World Leaders in Research-Based User Experience, [n.d.])

Vi har også anvendt usability heuristik nr. 5 om "Error Prevention". Dette bliver anvendt når en auktionarius skal godkende en auktion, så den begynder. Her findes en knap med label "Godkend", som starter med at være grå, og bliver grøn når en start- og slutperiode er valid. Dette stopper en bruger for at begå fejl, før den overhovedet sker.



Figur 7: Eksempel på error prevention af knap.

Refleksion og Konklusion

Refleksion over interviews

Gennem vores interviews med brugerne har vi fået en værdifuld indsigt i deres oplevelser, præferencer og kommet godt i mål. Det blev tydeligt, at vores spørgsmål var effektive til at afdække en bred mængde af brugernes behov og udfordringer, dog kunne det godt blive for systematisk, der til tider gav en akavet effekt på samtalen. Interviewpersonerne var engagerede og gav deres ærlige meninger med detaljerede svar, hvilket var afgørende for at forstå deres perspektiver. En refleksion over processen i vores tilfælde viser, at det er vigtigt at skabe en afslappet og uformel atmosfære under interviewene for at få de mest ærlige og givende svar, hvilket demografien var med til.

Refleksion mulige fremtidige design forberedelser i projektet

Brugerinddragelsesmetoden, bruger interviews, som vi har anvendt i projektet, hvilket har været afgørende for at forme vores platform. Vores interviewdeltagere har fremhævet forbedringer på baggrund af en analyse af deres interviews, og i vores oprettelse AS-IS brugerrejsen. De har også givet os ideer, der har krævet udviklingstid, men som vi ville overveje til en fremtidig løsning. For eksempel kunne en dedikeret kategori fane gøre

navigationen mere intuitiv og brugervenlig, hvilket er efterspurgt af ID2 og ID3, men det vil kræve for mange ressourcer til at kunne inkludere i POC.

Endnu et forslag vil være en fusion af live webkameraer i live auktioner, hvilket ville gøre det muligt for budgivere at deltage fysisk i auktionen hjemmefra, hvilket ID4 (ID4, 22:39-23:40) stod meget indenfor. Denne innovation ville tiltrække særlig interesse fra den yngre demografi, der lægger stor vægt på bekvemmelighed. Som et sidste punkt vil vi overveje udviklingen af en mobilapplikation, der letter deltagelse i auktioner via smartphones; dette trin kan øge brugerinddragelsen betydeligt især hos unge og gøre auktionsplatformen mere tilgængelig for brugerne via deres telefoner.

Evaluering af PoC og dens rolle i projektets samlede løsning.

Proof of Concept (PoC) har spillet en central rolle i vores projekt ved at give os mulighed for at visualisere og demonstrere vores ideer i en tidlig fase. Selvom der ikke har været direkte brugertests af PoC, hvilket er et muligt forbedringspunkt, har gruppen løbende evalueret og justeret designet baseret på indsigter fra brugerinterviews, som blev gennemført før udviklingen af PoC. Disse interviews gav os feedback, som vi har valgt at bruge til at informere og finjustere vores design.

Ved at have startet på vores POC tidligt i processen, har det været med til at identificere potentielle problemer og muligheder, der kan bruges til forbedringer, før vi investerede ressourcer i udviklingen af den endelige løsning. Den iterative proces med at skabe og forbedre PoC løbende, har sikret, at vores endelige design opfylder de identificerede brugerbehov og forventninger. Vi mener, at PoC nu er klar til at blive præsenteret for virksomheden samt potentielle interessenter. Den vil give dem en konkret forståelse af, hvordan platformen vil fungere, og hvilke fordele den kan bringe dem.

Konklusion på brugerinddragelsens bidrag til projektet.

Brugerinddragelse har været en afgørende faktor i udviklingen af vores PoC, men kunne have været endnu mere integreret. Vi har gennem de afholdte interviews fået feedback som har givet indsigt i brugernes behov, præferencer og udfordringer. Disse oplysninger har været uundværlige i forbindelse med udarbejdelsen af vores designbeslutninger. En forbedring vi kunne have implementeret, ville være at have fået interviews efter vores POC

var udviklet. Hvis vi havde gennemgået PoC med brugeren, kunne vi have fået nye perspektiver til forberedelse og vi kunne have været tættere på participatory design.

Projektet har vist, at ved at fokusere på brugernes oplevelser, er der mulighed for at skabe mere effektive, tiltalende og brugervenlige løsninger. Brugerinddragelsen har således været en vejledning for vores designbeslutninger, men kunne have været brugt mere aktiv igennem hele forløbet for at skabe en mere brugercentreret og succesfuld platform.

Agil Udvikling

Valg af Agile Metoder og Processer

Vi har valgt at arbejde med SCRUM og Extreme Programming (XP) som vores primære agile arbejdsmetoder. Valget af disse metoder skyldes deres komplementære styrker: SCRUM har givet os en struktureret ramme for planlægning og evaluering af projektteamet, mens XP har hjulpet med praksisser såsom parprogrammering og continuous integration, der har øget kvaliteten i udviklingsprocessen.

SCRUM Metodik

SCRUM er blevet anvendt som den overordnede ramme for projektledelse, hvilket har inkluderet definerede roller, ceremonier og artefakter, der har styret vores daglige arbejde og overordnede planlægning. Der var stor enighed i gruppen om at SCRUM skulle benyttes til projektarbejdet, baseret på projektets tidligere erfaringer med lignende projekter.

SCRUM Rollen

Vi har i projektteamet valgt at have en SCRUM-master som har stået med det overordnede ansvar for, at daily standup meetings bliver afholdt dagligt. Samtidig har han sikret, at de ikke tager unødvendigt lang tid. For at sikre lige ansvarsfordeling er rollen gået på tur mellem gruppemedlemmerne i hvert sprint.

SCRUM Ceremonier:

Poker Planning

Backloggen blev udarbejdet i starten af projektet med alle de forskellige user stories, som projektteamet skulle færdiggøre inden deadline d. 31. maj. Disse user stories blev herefter estimeret ved hjælp af Poker Planning, hvor alle projektmedlemmer vurderede, hvor mange story points de enkelte user stories skulle tildeles. Projektteamet vurderede, at dette ville være nyttigt at have på plads inden det primære arbejde begyndte.

"When asking everybody to estimate a story, we often discover discrepancies where two different team members have wildly different estimates for the same story. That kind of stuff is better to discover and discuss earlier than later." (Kniberg, [n.d.])

Sprint Planning

Projektteamet startede sprint planning efter, at de forskellige user stories på backloggen var estimeret. Det var udfordrende at estimere user stories relateret til microservices, da det var første gang, teamet arbejdede med denne arkitektur-type, der skulle kunne kommunikere sammen i dette omfang. Dette medførte, at nogle user stories blev højt estimerede, og vi arbejdede på dem løbende igennem sprint 1-3. Dette afspejledes i vores burndown charts, hvor vi løbende var bagud med udførte story points. Der var dog en forståelse i teamet for, at disse services skulle være færdige i løbet af sprint 3, hvilket ville frigøre mange storypoints. Derfor vidste vi i projektteamet, at selvom det så ud til at vi var bagud på vores burndown chart, ville vi hurtigt indhente det i løbet af sprint 3.

Daily Stand-ups

Vi har afholdt daily stand-ups, hvor hvert medlem skiftevis forklarede, hvad de havde opnået dagen forinden, hvilke problemer, tanker og frustrationer de stod med, og hvad de planlagde at arbejde videre med.

Sprint Reviews og Retrospectives

Projektteamet har opdelt projektperioden i fire forskellige sprints, ét for hver uge siden kickoff af projektets anden del den 2. maj. Efter hvert sprint, har vi haft en kort gennemgang af projektteamets indsats, fremgang og samarbejde i form af Sprint Reviews og retrospectives. Her har vi diskuteret hvad der fungerede godt, og hvad der kunne forbedres til de næste sprints.

SCRUM-artefakter

Backlog Administration i GitHub:

Vores produkt backlog er blevet administreret i GitHub, hvor alle user stories og tasks er blevet oprettet, prioriteret og tildelt. Dette har givet et klart og tilgængeligt overblik over projektets status og fremdrift. Her har vi også kunne tildele de forskellige sprint tasks, så vi har et overblik over, hvem der står med det primære ansvar for de forskellige user stories.

Burndown Charts:

Vi har brugt burndown charts til at følge fremskridt i vores sprints. Dette har hjulpet os med at planlægge løbende for at sikre, at vi holder os inden for tidsrammen for projektet. Her har vi i starten af hvert sprint estimeret de forskellige user stories ved hjælp af Planning Poker.

Vi har undervejs været lidt bagud på vores burndown charts, men det skyldes primært at vi har haft vores forskellige services i programmeringsdelen estimeret meget højt, som vi vidste med høj sandsynlighed ikke kunne færdiggøres på ét sprint. Dem havde vi vurderet til at blive færdige med på samme tid undervejs i Sprint 3.

Extreme Programming (XP)

I projektet kombineres SCRUM med XP-praksisser som f.eks. parprogrammering, hvilket har været meget positivt for udviklingsprocessen. Gennem brug af parprogrammering, har vi haft tæt samarbejde om komplekse opgaver, og opnået en dybere forståelse for systemets struktur.

I udviklingsfasen har vi benyttet parprogrammering flere gange, og det har især været gavnligt til at etablere forbindelse mellem de forskellige services, og deployment til Azure Bicep.

Selvom parprogrammering har været en værdifuld praksis i vores projekt, er der enighed i projektteamet om, at visse dele af udviklingsprocessen sagtens kan udføres uden denne tætte form for samarbejde. Dette har været særligt tydeligt i oprettelsen af nye service repositories og tilføjelsen af generelle metoder i disse.

Under udviklingen af helt nye service repositories og ved tilføjelse af de generelle http-metoder, har vi valgt at arbejde individuelt. Ved mange af disse opgaver som ikke havde den højeste kompleksitet, var dette en bedre fremgangsmåde for teamet. Ved at arbejde individuelt på disse opgaver har vi kunnet opnå en hurtigere opstart af kodegrundlaget og sikret effektivitet i udviklingen af de mere grundlæggende elementer i vores system. Koden er herefter blevet gennemgået fælles under code-reviews.

Collective Ownership

I projektteamet har vi arbejdet aktivt for at opnå kollektivt ejerskab af kodebasen. Dette har vi gjort ved at implementere og følge en række strategier, som sikrer, at hele teamet har kendskab til og ansvar for koden.

Standardiseret Kodestruktur

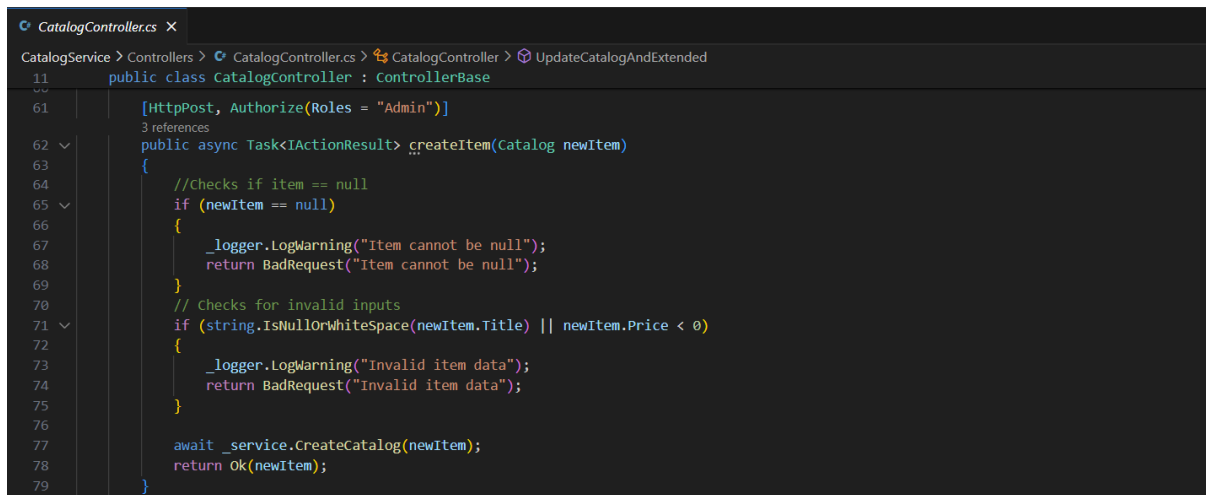
Vi har forsøgt at opretholde en standardiseret kodestruktur på tværs af alle vores services. Dette har hjulpet os med at sikre, at koden er konsistent og letforståelig, uanset hvem fra projektteamet som har skrevet det. For at opnå dette har vi brugt:

- **Parprogrammering:** Gennem løbende parprogrammering har teammedlemmer samarbejdet omkring de forskellige microservices. Dette har ikke kun forbedret kvaliteten af koden, men også øget fælles forståelse og ejerskab. Ved at skiftevis rotere projektmedlemmer ind i parprogrammering, opnår alle en større forståelse for forskellige dele af kodebasen.
I bogen *Scrum and XP from the trenches* beskrives det, hvordan parprogrammering med hyppig rotation af par, fører til et højt niveau af collective ownership: *"We've found that pair programming with frequent rotation of pairs automatically leads to a high level of collective code ownership. Teams with a high level of collective code ownership have proven to be very robust"* (Kniberg, [n.d.])
- **Code Reviews:** Vi har gennemført fokuserede code reviews, hvor ny kode bliver gennemgået og diskuteret. Dette har hjulpet os med sikre, at alle kodningsstandarder og best practices bliver fulgt. Under disse reviews, har vi også delt viden og erfaringer mellem teammedlemmer.
- Vi har fulgt specifikke kodningsstandarder og best practices, herunder SOLID-principperne og refactoring. Disse retningslinjer har hjulpet os med at skrive robust og vedligeholdelsesvenlig kode. Under code reviews har vi fokuseret på at sikre, at ny kode lever op til disse standarder så vidt som muligt.

Anvendelse af Single Responsibility Principle (SRP)

I vores projekt har vi nøje fulgt Single Responsibility Principle (SRP) ved at opdele ansvarsområderne mellem forskellige services. Dette kan ses i POST-metoderne i vores Catalogservice og Auctionservice, hvor Catalogservice kun står for at oprette et produkt og tilføje det til vores MongoDB i Atlas, og Auctionservice herefter kan finde dette produkt og oprette det som en auktion.

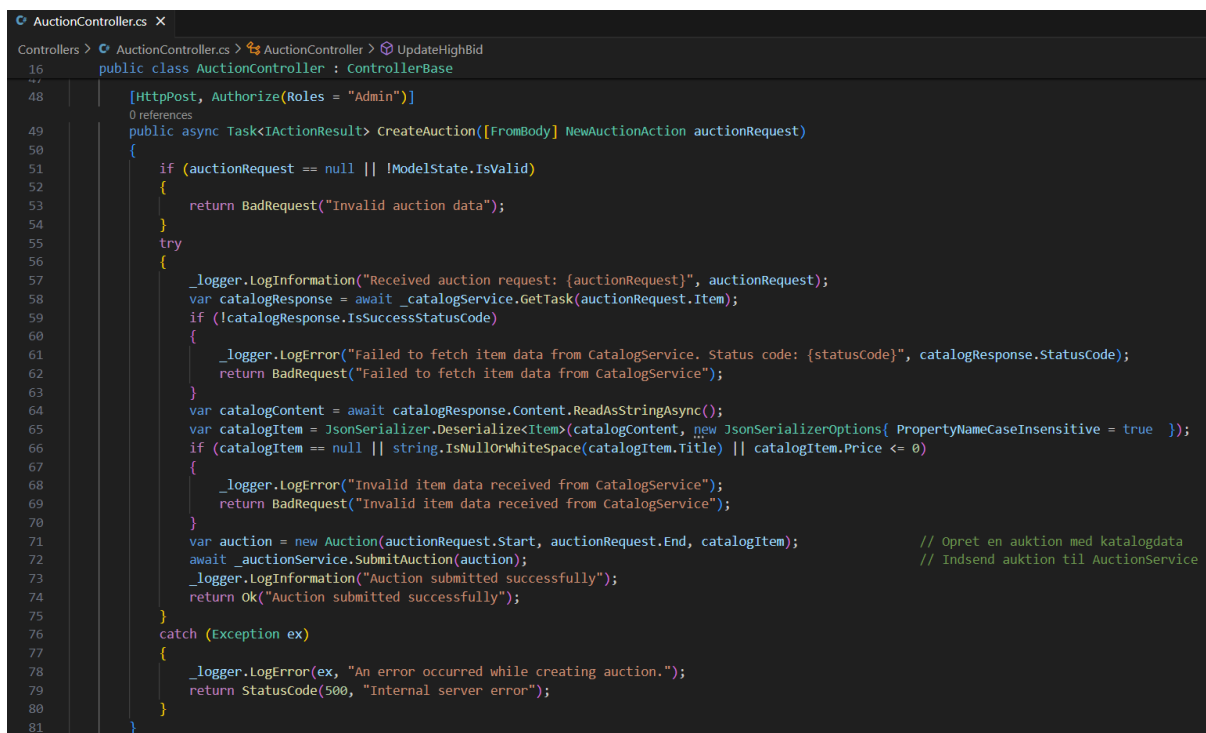
Catalogservice er ansvarlig for at håndtere produktinformationer. Her er et eksempel på en POST-metode, der opretter et nyt produkt:



```
11 public class CatalogController : ControllerBase
61 [HttpPost, Authorize(Roles = "Admin")]
62 public async Task<IActionResult> createItem(Catalog newItem)
63 {
64     //Checks if item == null
65     if (newItem == null)
66     {
67         _logger.LogWarning("Item cannot be null");
68         return BadRequest("Item cannot be null");
69     }
70     // Checks for invalid inputs
71     if (string.IsNullOrEmpty(newItem.Title) || newItem.Price < 0)
72     {
73         _logger.LogWarning("Invalid item data");
74         return BadRequest("Invalid item data");
75     }
76
77     await _service.CreateCatalog(newItem);
78     return Ok(newItem);
79 }
```

Figur 8: Screenshot fra kodebasen – CatalogService – CatalogController.cs - createItem

Auctionservice er ansvarlig for at håndtere auktioner. Her er et eksempel på en POST-metode, der opretter en ny auktion for et produkt på baggrund af et catalog-produkt.



```
16 public class AuctionController : ControllerBase
48 [HttpPost, Authorize(Roles = "Admin")]
49 public async Task<IActionResult> CreateAuction([FromBody] NewAuctionAction auctionRequest)
50 {
51     if (auctionRequest == null || !ModelState.IsValid)
52     {
53         return BadRequest("Invalid auction data");
54     }
55     try
56     {
57         _logger.LogInformation("Received auction request: {auctionRequest}", auctionRequest);
58         var catalogResponse = await _catalogService.GetTask(auctionRequest.Item);
59         if (!catalogResponse.IsSuccessStatusCode)
60         {
61             _logger.LogError("Failed to fetch item data from CatalogService. Status code: {statusCode}", catalogResponse.StatusCode);
62             return BadRequest("Failed to fetch item data from CatalogService");
63         }
64         var catalogContent = await catalogResponse.Content.ReadAsStringAsync();
65         var catalogItem = JsonSerializer.Deserialize<Item>(catalogContent, new JsonSerializerOptions{ PropertyNameCaseInsensitive = true });
66         if (catalogItem == null || string.IsNullOrEmpty(catalogItem.Title) || catalogItem.Price <= 0)
67         {
68             _logger.LogError("Invalid item data received from CatalogService");
69             return BadRequest("Invalid item data received from CatalogService");
70         }
71         var auction = new Auction(auctionRequest.Start, auctionRequest.End, catalogItem);
72         await _auctionService.SubmitAuction(auction);
73         _logger.LogInformation("Auction submitted successfully");
74         return Ok("Auction submitted successfully");
75     }
76     catch (Exception ex)
77     {
78         _logger.LogError(ex, "An error occurred while creating auction.");
79         return StatusCode(500, "Internal server error");
80     }
81 }
```

Figur 9: Screenshot fra kodebasen – AuctionService – AuctionController.cs - CreateAuction

Teststrategi

TDD og MSTest er lavet i vores CatalogService repository, det kan hentes og testes herfra:

<https://github.com/Hablas-Si/CatalogService.git>

Test Driven Development (TDD)

Vi startede i projektet med TDD på vores CatalogService. Dette betyder, at vi har lavet tests til de forskellige HTTP-metoder, hvor de startede med fejl, da vi ikke havde lavet metoderne. Vi programmerede derefter vores metoder til vores test, hvor vi lavede mock data, som er tilpasset vores metoder, for at se om testene virkede. På baggrund af dem, var vi sikre på, at de indeholdte diverse checks og logging.

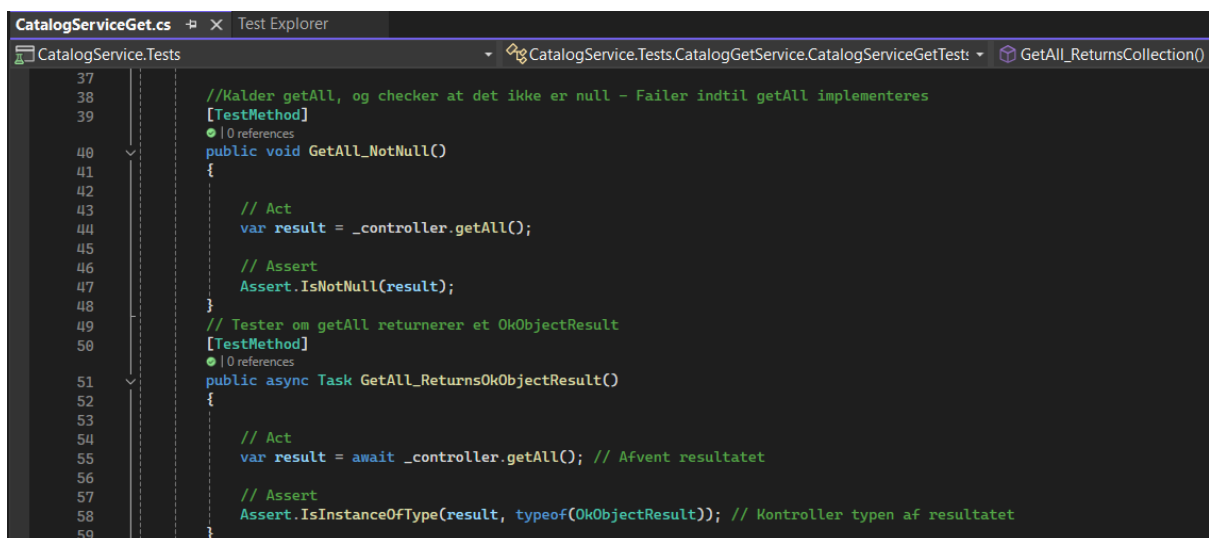
Et eksempel på vores brug af TDD i projektet, kan ses i eksemplet herunder fra CatalogService. Disse tests blev lavet forinden vores første HTTP-GET metode, og fejler derfor indtil metoden blev lavet, og tester derefter med mock data, og om de har nødvendige checks.

Den første test GetAll_NotNull, tester at resultatet fra getAll metoden ikke er null.

Den anden test GetAll_ReturnOkObjectResult, tester at responsen fra resultatet er af typen OkObjectResult.

Med disse TDD tests på plads, kunne vi forme getAll metoden på baggrund af disse krav, så den fra start af har meget af den ønskede funktionalitet.

Efter metoden blev indført, bestod disse tests, og vi var sikre på at de ønskede checks var tilstede.



```
37
38
39 //Kaldet getAll, og checker at det ikke er null - Failer indtil getAll implementeres
[TestMethod]
40 public void GetAll_NotNull()
41 {
42
43     // Act
44     var result = _controller.getAll();
45
46     // Assert
47     Assert.IsNotNull(result);
48 }
49
50 // Tester om getAll returnerer et OkObjectResult
[TestMethod]
51 public async Task GetAll_ReturnsOkObjectResult()
52 {
53
54     // Act
55     var result = await _controller.getAll(); // Afvent resultatet
56
57     // Assert
58     Assert.IsInstanceOfType(result, typeof(OkObjectResult)); // Kontroller typen af resultatet
59 }
```

Figur 10: Screenshot fra kodebasen – CatalogService – CatalogService.Tests – CatalogServiceGetTests

API Tests i Postman

Vi har haft særligt stort fokus på tests af vores API-endepunkter i Postman, da det giver et godt overblik over vores fejlhåndtering af ugyldig data input og succesfulde API-kald.

Konklusion af testresultater

Vores teststrategi har bidraget til at sikre, at vores applikation opfylder de krav, vi har defineret. Ved hjælp af tests i Postman, har vi identificeret og rettet fejl samt sikret, at vores applikation opfører sig som forventet.

Continuous Integration og Delivery (CI/CD)

Projektteamet har benyttet sig af Continuous Integration og Continuous Delivery (CI/CD) i projektet ved hjælp af GitHub Actions Pipelines samt Git Flow-init. Denne opsætning har hjulpet os med at automatisere vores builds og tests, hvilket har bidraget til en mere effektiv og pålidelig udvikling. Det har været meget brugbart med GitHub Actions løbende under udviklingen, eftersom man hurtigt kan se, hvis et build der bliver pushet op fejler, hvorefter vi har prioriteret at fikse buildet så hurtigt som muligt. Dette nævnes også som god praksis i artiklen 'Continuous integration' af Martin Fowler:

"Continuous Integration can only work if the mainline is kept in a healthy state. Should the integration build fail, then it needs to be fixed right away. As Kent Beck puts it: "nobody has a higher priority task than fixing the build"." (Fowler, [n.d.])

Git Flow

Alle services har hvert sit repository på GitHub, som er samlet i en organisation. Hver service benytter Git Flow til udviklingen, så alle nye features, tilføjelser eller ændringer udarbejdes på feature branches. Når en feature er færdig, oprettes en pull request for at merge den ind i vores development branch. Denne proces omfatter følgende trin:

Feature Branches: Udviklingen af nye funktioner sker på separate feature branches. Dette sikrer, at hovedbranchen forbliver stabil og kun indeholder gennemtestet kode.

Pull Requests: Når en feature branch er klar, oprettes en pull request for at merge koden ind i development branchen. Pull requests bliver gennemgået af teammedlemmer, hvilket hjælper med at sikre kodens kvalitet og finde fejl tidligt i processen.

Code Reviews: På GitHub kan vi gennemgå disse pull requests, inden de bliver pushet til development branchen. Dette giver et bedre overblik over den kode, der bliver pushet, og muligheden for at opdage fejl, inden de bliver merged ind i vores udviklingsmiljø.

Parprogrammering og Code Reviews: Ofte, når vi har lavet parprogrammering, er det de samme to medlemmer, som reviewer og godkender pull requesten. Vi mener, at dette stadig er en god praksis, eftersom de to projektmedlemmer sammen kan vurdere, om koden er god nok. Til større ændringer inddrages et tredje projektmedlem for at gennemgå koden, før den lægges op.

Skalering af Projektet

Ud fra omfanget af det tildelte projekt, har projektteamets størrelse på 5 personer været meget passende. Vi har haft et tæt samarbejde og benyttet XP og SCRUM, i så stor grad det har været gavnligt. Alle i teamet har haft et holistisk indblik i projektet og har været en del af alle arbejdsopgaverne.

Hvis vi skulle skalere projektet op til 30 personer og samtidig bevare den agile tilgang, ville vi opdele teamet i mindre, specialiserede teams. Disse teams kunne inkludere:

- Et frontend- og UX-team
- Et backend-team
- Et team fokuseret på virksomhedens vision og målbillede
- Et SCRUM-team, der koordinerer den agile udvikling

Dette ville have mange fordele, såsom øget leverance og en mere grundig udførelse på de forskellige områder, men teammedlemmerne ville ikke have det samme holistiske overblik over projektet, som vi mener er en klar fordel på det aktuelle projekt.

Metodevalg og Tilpasninger

Kombinationen af SCRUM og XP har været meget passende i forhold til projektets omfang. Selvom vi har været et relativt lille team på fem medlemmer, har det været nyttigt at følge en overordnet SCRUM-struktur for at opretholde et overblik undervejs i projektet.

Grundet projektets omfang, og størrelsen på projektteamet, har nogle af SCRUM-elementerne, såsom daily standups, været nedprioriteret. Vi har stadig afholdt dem dagligt, men eftersom vi altid har siddet samlet og løbende diskuterer igangværende arbejdsopgaver, bliver de hurtigt gennemgået. Til gengæld har vi i høj grad benyttet os af XP og par-programmering, da udviklingen af systemet har været meget kompliceret, og alle gruppemedlemmer gerne vil have et indblik i kodebasen og strukturen. Disse metodevalg og tilpasninger har mødt projektteamets forventninger, og der er klar enighed om, at det har været en passende fremgangsmåde for projektet.

IT-arkitektur- og infrastruktur

Indledning

Baseret på de tidligere afsnit i denne rapport står det klart, at der skal udvikles et microservice-system, der kan håndtere auktioner online. For at sikre det bedst mulige resultat anvendes Design Driven Development til at få et overblik over domænet, og til at udvælge den bedst mulige løsning. I dette dokument redegøres der for processen og sammenhængen af den endelige løsning.

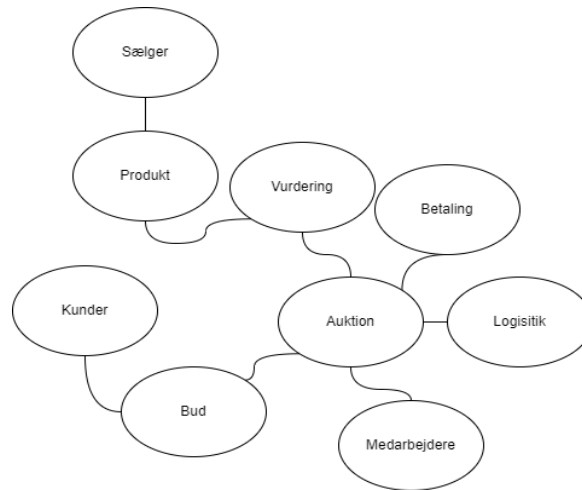
Designprocess (DDD)

På baggrund af vores analyser i Målbilledet er det blevet besluttet, at auktions-løsningen skal opbygges som en microservice-arkitektur. Dette er af hensyn til skalerbarheden og resiliensen i systemet, men introducerer samtidigt en række komplikationer.

Når man arbejder med microservices, er den store udfordring ikke bare at definere de enkelte services, men også at opsætte klare afgrænsninger for dem. Til det formål anvendes Domain Driven Design (DDD), da det kan bruges som et framework for at komme ordentligt fra start. Det centrale er, at én service som udgangspunkt skal gøre én ting, og det kan kun lade sig gøre hvis man har en ordentlig forståelse for domænet. (martinekuan, 2022)

Analyse af Domæne

Som værktøj til analyse af domænet anvendes en domænemodel. Domænemodellen giver et overordnet overblik over forretningsdomænet og viser sammenhænge mellem eventuelle sub-domæner. (martinekuan, 2022)



Figur 11: Domænemodel

Entiteter

Med en domænemodel i hånden er næste step at identificere de enkelte entiteter. Ud fra domænemodellen kan vi opstille en række scenarier som systemet skal kunne håndtere, og herudfra gives bud på entiteter. (martinekuan, [n.d.])

Scenarier:

- En sælger kommer ind til auktionshuset og får sin vare vurderet
- Varen tildeles et label og sættes på lageret indtil en auktion kan finde sted
- Kunder ankommer til auktionshuset og afgiver bud på varen
- Kunden med højeste bud orienteres via mail og opråb
- Auktionshuset opkræver betaling og salær, og pakker varen
- Kunden tager varen med hjem eller den sendes med posten

På baggrund af de scenarier er følgende entiteter identificeret:

- Auktion
- Bud
- Kunde
- Medarbejder
- Logistik
- Vare (eventuelt 'katalog')
- Vurdering
- Notifikation (eventuelt blot 'email')
- Betaling

Aggregater

Det næste man kigger på er, om nogle af de identificerede entiteter kan samles i såkaldte aggregater. Aggregater består af en eller flere entiteter som modellerer virkelighedens sammenhænge. For eksempel skal en auktion kunne bindes sammen med nogle bud, og bud skal kunne spores til den bruger, der har indsendt det. Aggregaterne er oftest gode kandidater til egentlige microservices. (martinekuan, [n.d.])

Identificerede aggregater:

Root entity - Auktion: Har bud og vare som child-entiteter.

Root entity - Bud: Har kunde som child-entitet

Root entity - Vare: Har vurdering og sælger som child-entiteter

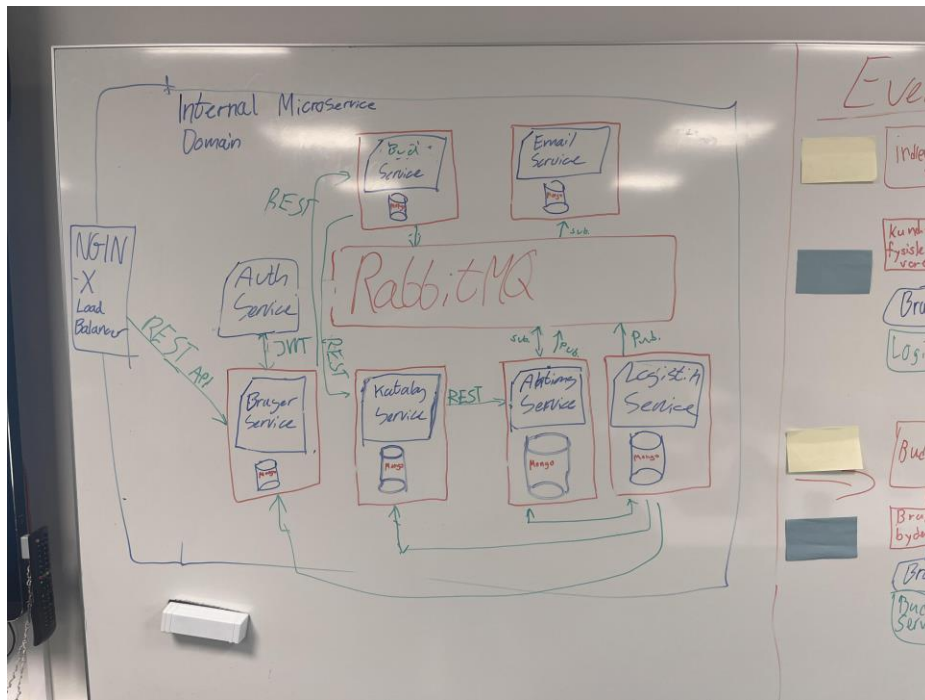
Root entity - Betaling: Har kunde og Auktion som child-entiteter

Root entity - Logistik: Har auktion og betaling som child-entiteter

Det vurderes, at de identificerede aggregater håndterer en afgrænset del af forretningslogikken, og at de ikke behøver lave konstante kald mellem hinanden. Hvis man lægger validering i bud-aggregatet skal der kun opdateres i auktionen når der kommer et validt bud igennem. Dette kan gøres yderligere robust ved brug af shared cache (redis) og rabbitMQ. Derfor fortsættes projektet med ovenstående services, som repræsentanter for forretningslogikken.

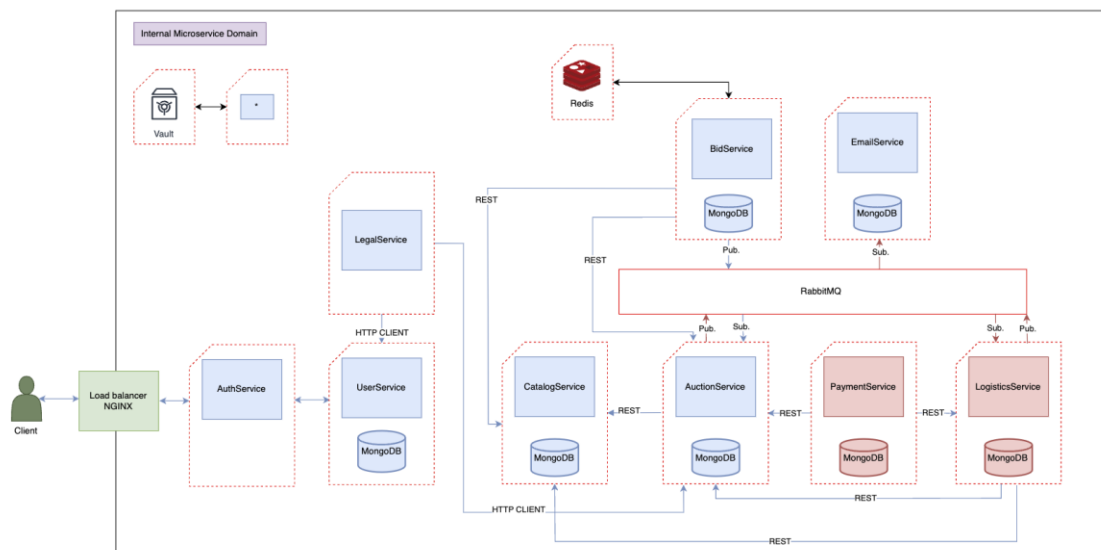
Deployment

I dette afsnit beskrives deployment af vores bud på et Proof of Concept for auktionssystemet. Der tages udgangspunkt i et deployment diagram som viser de enkelte services samt strømmingen af data imellem dem. Der er tilføjet flere administrative services i forhold til analysen herover, blandt andet for at opretholde sikkerhed og minimere API-kald mellem services. Vi starter i sprint 1 at udforme et arkitektur-diagram som så således ud:



Figur 12: Version 1 af deployment for Auktionshuset's POC

Der kom flere services, som UserServices på og fjernede Logistik og Email, som en del af implementeringen, da de ikke vægtede så højt og var mere et "nice to have" i vores prioritering af vores PoC. Der blev også rettet fejl som f.eks. kaldende imellem services. Vi havde først en vision om at samle UserService-database og logik i en service, nemlig AuthService, men det ville ikke give den løse kobling vi ønskede ved microservice-arkitekturen. Senere sprints senere kom vores arkitektur til at se sådan her ud:

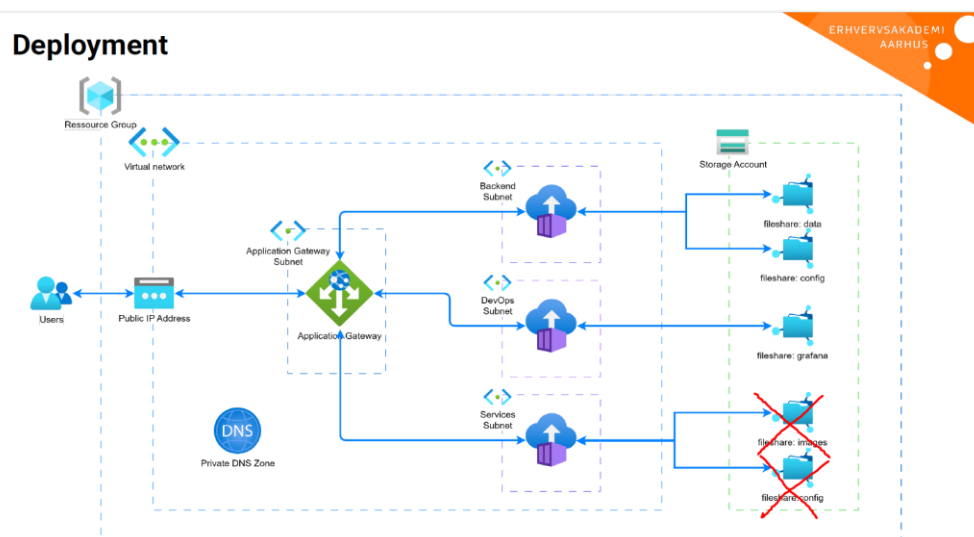


Figur 13: deployment diagram for Auktionshuset's POC

Som vores PoC ser ud lige nu, har vi vores services, en NGINX konfiguration, som er en load balancer som styrer trafikken i vores service system/domæne, derudover har loki og grafana, som står for vores monitorering og overvågning (DevOps). Loki indsamler data gennem vores NLog, og bliver derfra vist inde på grafanas UI.

Det vi har fået implementeret i vores PoC lige nu, består af en AuthService, som er en godkendelses service, som giver os en JWT token, der gør at vi kan tilgå de forskellige services, vi har derudover to roller for en bruger og en admin som er inkorporeret med JWT role claims, da de to roller skal have adgang til de forskellige funktioner/kald i andre services. UserService har CRUD funktioner, som at man fx kan oprette sig og se brugere mm. Oprettelse kan alle gøre uden jwt-token, men sådan noget som at se brugere er begrænset til roller med admin jwt-token. Det samme er gældende for vores Catalog, bare med produkter.

Derudover har vi en BiddingService, hvor man som bruger, efter man har fået tildelt en token, kan sende et bud igennem servicen med den betingelse af, at den er højere end det nuværende bud. BiddingService har derudover Redis, som er et højhastighedsdata lager, som gemmer en delmængde af data, så fremtidige anmodninger bliver fremsendt hurtigere, end det er at hente fra MongoDB. (Savaş, 2022) Vores auktionsservice består udelukkende af at kunne oprette en auktion, ved at den kan hente data fra biddingservice og catalogservice. Derudover har vi en HashiCorp vault, som alle services tilgår og bruger, alt er der for at skjule følsomme oplysninger, og som kun kan tilgås af projektteamet.



Figur 14: Deployment med Azure Bicep

Vi har brugt bicep scripts til at deploye vores microservices på Azure. Den består af en App. Gateway og 3 subnets - Backend, DevOps og Services. Services har en nginx i sig, som fungerer som en proxy til kald til de forskellige services. I stedet for at map alle portene på App. Gateway'en så alle services får deres egen port, kan vi vha. nginx map dem alle til port:4000. Så App. Gateway kalder på nginx som sender forespørgslerne videre ud fra nginx.conf filen. I nginx.conf filen har vi mappet dem videre til alle services med:

“:4000/api/<navn på servicecontroller>/<endepunkt>”

I vores VNET er der også en DNS, som sørger for at oversætte vores loki, services, backend og devops til de rigtige subnets/containerer. Vi bruger også Azure Storages til grafana, nginx.conf og vault certifikater. Vores bicep scripts kan findes i vores Deployment repository på GitHub.

Interoperabilitet

For at sikre interoperabiliteten mellem services i løsningen, anvendes der modelklasser ved dataoverførsler. Modelklasserne agerer skemaer som sikrer, at service 1 afsender præcis det service 2 skal bruge, hverken mere eller mindre.

Som eksempel kan kommunikationen mellem AuctionService (AS) og CatalogService (CS) nævnes. For at en ny auktion kan oprettes i AS, kræver det reference til en vare der ligger i CS. Derfor er oprettelse af en auktion et POST endepunkt, hvor CS sender en modelklasse af typen 'item' til AS, hvor det indsættes i en identisk modelklasse. Da samme modelklasse anvendes på begge sider af kommunikationen elimineres risikoen for fejlkommunikation.



Figur 15: Modelklasser i Catalog- og AuctionService der agerer schema

Vi har lavet en LegalService som står for interoperabiliteten mellem auktionshuse og statsmyndighederne med online aktiviteter. Tjek LegalService repositories readme.md fil for mere info.(Hablas Si# GitHub, 2024)

LegalService API Endpoints

Auction Endpoints

GET /api/legal/auctions

Henter en liste over auktioner.

Sikkerhed: Kræver Admin JWT-token.

GET /api/legal/auctions/{auctionId}

Henter detaljer om en specifik auktion.

Parametre:

- `auctionId`: UUID for den auktion, der skal hentes. Tjek Database.

Sikkerhed: Kræver Admin JWT-token.

User Endpoints

GET /api/legal/users/{userId}

Henter detaljer om en specifik bruger.

Parametre:

- `userId`: UUID for den bruger, der skal hentes. Tjek Database.

Sikkerhed: Kræver Admin JWT-token.

Auth Endpoints

POST /api/AuthService/login

Brugerlogin og JWT-token generering af en user token.

Body-indhold:

```
{
  "username": "JohnDoe",
  "password": "secret123"
}
```

Figur 16: LegalService API Endpoints

Ønskede funktionaliteter

- Sequentiel Convoy i RabbitMQ
- Vinderen skulle have fået tilsendt en mail gennem EmailService, når man har vundet sit produkt.
- Vi ønskede at lave en logistikservice, hvilket betød at når man havde vundet skulle man udfylde en forsendelses formular, hvor man kunne følge sit produkt i hvor langt den var.
- Når der bliver oprettet en bruger, skal deres password hashes af password med salt.
- Kunne overvejes at implementere noget eventbased med azure funktions - f.eks. til dokumentation.

Drift

Logging

Logging i programmering er en proces der er med til at registrere hændelser, aktiviteter og meddelelser fra et softwareprogram. Det er alt fra fejlmeddelelser, advarsler og brugerinteraktioner. Projektgruppen bruger dermed logging til at registrere og gemme information, hvilket kan være nyttig for fejlfinding og optimering af systemet.

Hvorfor er logging godt?

Logging bruges for hurtig og effektiv fejlfinding, fordi det giver detaljerede oplysninger om fejl og kan hjælpe udviklere med at spore tilbage til deres services, hvis der opstår problemer, hvilket kan være med til at spare og minimerer nedetid. Ved logging bliver det muligt at overvåge ydeevnen i systemet og kan være med til at hjælpe med at identificere og optimere processer. Hvis man kigger på et sikkerhedsmæssigt perspektiv, kan logging hjælpe med at opdage adgangsforsøg og opfylder lovgivningsmæssige krav til databeskyttelse.

Hvordan kan logging bruges i et forretningsmæssigt perspektiv?

Logging giver et overblik i brugeradfærd og systemets funktionalitet, hvilket kan hjælpe virksomheder med at forstå kundernes behov og optimering af processor. Logging giver en fordel, da det sikrer en høj kvalitet, og kan være med til at øge kundetilfredsheden og understøtte sikkerheden. Ved at integrere logging i deres it-system kan det dermed fremme en forretningsmæssig succes. (Stefano Vardalos, 2017)

Loki

Loki er et log management system udviklet af Grafana Labs, designet til at være enkelt og skalerbart. I hver service har vi en NLog fil, der indsamler alle logs til de enkelte services. Dette gør det lettere og billigere at håndtere store mængder logdata. Ved hver service har de fået tildelt labels, så udviklingsholdet hurtigt kan søge og analysere specifikke logdata. (Grafana Labs, [n.d.])

Grafana

Grafana er en open-source platform, der anvendes af projektgruppen til visualisering af data, herunder logs fra Loki. Med Grafana kan der oprettes dashboards, der viser logs i realtid gennem visualiseringer. Den brugervenlige grænseflade gør det muligt for både tekniske og ikke-tekniske brugere at forstå og analysere logdata nemt, hvilket kan være med til at forbedre systemovervågning og beslutningstagninger. (Grafana Labs, [n.d.])

Aggregering

Dette er et eksempel på, hvordan loki indsamler det ved hjælp af NLog og bliver derefter visualiseret i grafana. Der bliver logget, når der bliver kaldt på endepunkterne.

```
[HttpGet("all"), Authorize(Roles = "Admin")]  
public async Task<IActionResult> getAll()  
{  
    _logger.LogInformation("Fetching all items from catalog");  
    var catalogList = await _service.getAll();  
    return Ok(catalogList);  
}
```

Figur 17: Screenshot fra kodebasen – CatalogService – CatalogController

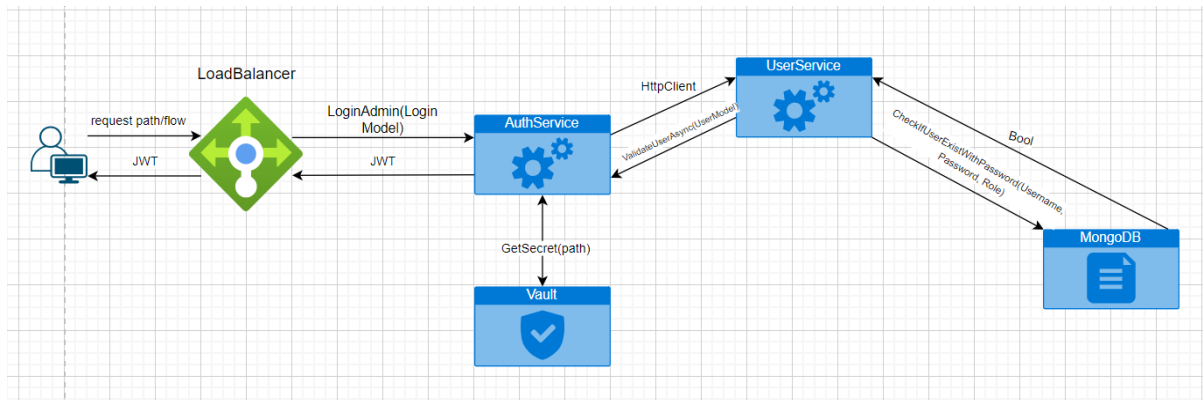
Docker

Der er blevet anvendt docker til udviklingsmiljøet. Alle Services har et bygget docker-image oppe i dockerhub, som endnu videre bliver testet i en composefil med en nginx. Det har vi valgt at gøre for at opnå fordelene:

- **Isolation:** Hver enkelte service er isoleret fra hinanden, så hvis systemet går ned, kan nogle dele af systemet stadig virke også til testing/development.
- **Skalérbarhed:** Man behøver kun at duplikere en service for at skalere systemet.
- **Portabilitet:** Alle kan hente compose-filen og teste det på deres egen maskine alle steder og maskiner.
- **Konsistens:** Ved at bruge containere sikrer vi, at applikationen kører på samme måde i alle miljøer, hvilket reducerer "it works on my machine" problemer.

Vi har også brugt GitHub Actions som vores Continuous Integration (CI) pipeline til at bygge og pushe docker-image til dockerhub hver gang der bliver pushet på main branch i hver service.

Authentication & JWT



Figur 18: Oversigt over authentication

Vores uddeling af JWT bliver lavet vha. AuthService, UserService, HashiCorp Vault og en mongo database. En client har et request flow til AuthService igennem nginx loadbalancer og kalder på metoden "LoginAdmin()" eller "LoginUser()" med en body med Username og Password. Den kalder først en anden metode som er ValidateUserAsync() som går ind og laver en efterspørgsel " /api/User/login/validate" i UserService Controlleren vha. httpclient.

Inde i UserService bliver der så kaldt en "CheckIfUserExistWithPassword()" som tjekker body'en om den findes i mongo databasen og returnerer så en bool. Hvis bool'en returnerer true, vil LoginAdmin() metoden så tjekke om den pågældende bruger har rollen "Admin" i User.Role. Hvis ja, kalder den på GenerateJwtToken() som tager et username og "isAdmin" sat til true i parameterene.

Den vil så udskrive en JwtToken med et Role Claim baseret på hvad isAdmin parameteren er true eller false. True = Admin Role og False = User Role.

Denne JwtToken er aktiv i 15 min. og vi bruger Role Claims i alle vores Services for sikkerheden. Hver af disse Services har forbindelse til en Vault med en getSecret metode der udtrækker secrets fra vault som bl.a. JwtSecret og Issuer til JwtToken til at identificerer den indkommende jwt-token. Disse bliver brugt til authorization på diverse endpoints.

Referencer

Anne-Sofie Thomsen, 2021. *Øjenåbnende brugerrejser - trin for trin*. 1. udgave udg. Metoder trin for trin. Frederiksberg: Samfundslitteratur.

Brown, Tim, 2008. Design Thinking. *Harvard Business Review*. [online] 1 jun. Tilgængelig hos: <<https://hbr.org/2008/06/design-thinking>> [Set 31 maj 2024].

Fagrådet for Strategi og Ledelse, 2016. *Anbefalinger om it-strategi*. Tilgængelig hos: <<https://www.dit.dk/-/media/Files/PDF/Gratis%20publikationer/Anbefalinger%20om%20it-strategi.ashx>> [Set 1 apr. 2024].

Fowler, Martin, [n.d.]. *Continuous Integration*. [online] martinowler.com. Tilgængelig hos: <<https://martinfowler.com/articles/continuousIntegration.html>> [Set 23 maj 2024].

GDPR.DK, 2021. De nye GDPR regler forklaret på 5 min. | GDPR.DK. Tilgængelig hos: <<https://gdpr.dk/>> [Set 30 maj 2024].

Grafana Labs [n.d.]. *Grafana Loki OSS | Log aggregation system*. [online] Grafana Labs. Tilgængelig hos: <<https://grafana.com/oss/loki/>> [Set 16 maj 2024].

Hablas Si# GitHub, 2024. *LegalService/README.md at main · Hablas-Si/LegalService*. [online] Tilgængelig hos: <<https://github.com/Hablas-Si/LegalService/blob/main/README.md>> [Set 31 maj 2024].

ITGovernance [n.d.]. *What is COBIT 5? Definition & Explanation*. [online] Tilgængelig hos: <<https://itgovernance.co.uk/cobit>> [Set 14 maj 2024].

Kniberg, Henrik, [n.d.]. *Scrum and XP from the trenches*. 2. Edition udg. [online] InfoQueue. Tilgængelig hos: <<https://leanagiletraining.com/wp-content/uploads/2020/03/Scrum-and-XP-from-the-Trenches-2nd-edition.pdf>> [Set 23 maj 2024].

Labs, Engineering Department | Pacific Research, 2020. 5 Advantages of Iterative Design and Prototyping. *Pacific Research Laboratories*. Tilgængelig hos: <<https://www.pacific-research.com/5-advantages-of-iterative-design-and-prototyping-prl/>> [Set 31 maj 2024].

Loreno Ramonda 2016. *Git Flow: how it works*. [online] Bitbull. Tilgængelig hos: <<https://www.bitbull.it/en/blog/how-git-flow-works/>> [Set 13 maj 2024].

martinekuan, 2022. *Domain analysis for microservices - Azure Architecture Center*. [online] Tilgængelig hos: <<https://learn.microsoft.com/en-us/azure/architecture/microservices/model/domain-analysis>> [Set 30 maj 2024].

martinekuan, [n.d.]. *Using tactical DDD to design microservices - Azure Architecture Center*. [online] Tilgængelig hos: <<https://learn.microsoft.com/en-us/azure/architecture/microservices/model/tactical-ddd>> [Set 30 maj 2024].

Newman, Sam, 2021. *Building microservices: designing fine-grained systems*. Second Edition udg. Beijing: O'Reilly Media.

RISMA, [n.d.]. Hvad er governance, og hvorfor er det vigtigt? [online] Tilgængelig hos: <<https://www.rismasystems.com/da/ressourcer/artikler/hvad-er-governance>>.

Ross, Jeanne W., et al., 2006. *Enterprise architecture as strategy: creating a foundation for business execution*. Boston, Mass: Harvard Business School Press.

Savaş, Mert, 2022. In-Memory Cache: Use of Memory Caching In .NET Core. *Berkut Teknoloji*. Tilgængelig hos: <<https://medium.com/berkut-teknoloji/in-memory-cache-use-of-memory-caching-in-net-core-41d99153ebd0>> [Set 30 maj 2024].

Stefano Vardalos 2017. *An introduction to logging for programmers*. [online] freeCodeCamp.org. Tilgængelig hos: <<https://www.freecodecamp.org/news/you-should-have-better-logging-now-fbab2f667fac/>> [Set 15 maj 2024].

Sundhedsdatastyrelsen 2023. *Målbillede for meddelelseskommunikation på sundhedsområdet*. [online] sundhedsdatastyrelsen.dk. Tilgængelig hos: <<https://sundhedsdatastyrelsen.dk/-/media/sds/filer/rammer-og-retningslinjer/referencearkitektur-og-it-standarder/referencearkitektur/maalbillede-for-meddelelseskommunikation.pdf?la=da>> [Set 29 maj 2024].

The Open Group [n.d.]a. *Phase A: Architecture Vision*. [online] <https://opengroup.org/>. Tilgængelig hos: <<https://pubs.opengroup.org/architecture/togaf8-doc/arch/chap05.html>> [Set 29 maj 2024].

The Open Group [n.d.]b. *The TOGAF Standard, Version 9.2 - Architecture Principles*. [online] opengroup.org. Tilgængelig hos: <https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap20.html#tag_20_06> [Set 29 maj 2024].

World Leaders in Research-Based User Experience [n.d.]. *Maintain Consistency and Adhere to Standards (Usability Heuristic #4)*. [online] Nielsen Norman Group. Tilgængelig hos: <<https://www.nngroup.com/articles/consistency-and-standards/>> [Set 31 maj 2024].

Bilag

Bilag A - Uddrag af "Projektbeskrivelse for 4. semester projekt - fuld beskrivelse":

Auktionshuset Grøn og Olsen er et 150 år gammelt auktionshus i Aarhus, der med stor succes placerede sig som auktionshuset for high-end produkter indenfor møbler, porcelæn, sølv, guld, smykker osv.

Rasmus Grøn er den karismatiske chef for auktionshuset og 4. generation fra Grøn-Olsen familien. Deres auktioner er kendt for at være fyldt med humor, men stadig seriøse og tiltrækker altid en større folkemængde i auktionshuset i Grønnegade. Auktionshuset er samtidig kendt for at opnå priser over vurderingen og er derfor eftertragtet blandt samlere med unikke effekter.

Indlevering af effekter foregår ved at sælger møder op i Grønnegade, hvor vurderingen foretages, baseret på de ansattes ekspertise. Auktionerne er fysiske, men man kan deltage per telefon. Under Corona-nedlukningen har der været forsøg med at købere kunne deltage via en online chat, men forsøget gik under forventning og initiativet er ikke blevet gentaget.

Virksomhedens IT-systemer begrænser sig til et system, hvor effekter registreres med basale oplysninger omkring sælger og produktet, en vurdering og en mindstepris. Efter en auktion registreres salgsprisen og sælgers informationer. Dette overføres til et økonomisystem, der står for afregning (faktura til køber og betaling til sælger).

Virksomheden har en simpel hjemmeside sat op i WordPress. Her annonceres auktionsdatoer, samt udvalgte effekter i en forældet template. Siden indeholder en kort beretning om virksomhedens historie samt adresse. Virksomheden har en Facebook side, men bruger den ikke.

Auktionshuset Grøn og Olsen fik i efteråret 2020 en henvendelse fra en større international investor, der ønsker at være med til at vækste virksomheden og sikre forretningen mange år frem. Desværre trak investoren sig igen, da den overvejende manuelle drift og de forældede it-systemer, vurderes at hæmme digitaliseringspotentialet.

Rasmus' datter Trine Grøn blev i 2020 færdiguddannet med en PBA i It-arkitektur og sammen med sin fætter Jan Olsen, der er uddannet i England som auktionarius, erkendte de ved en familiefest at der skulle ske radikale ændringer. Selvom auktionshuset stadig er en god forretning, så ved Trine at man til stadighed skal forbedre sig. Derfor besluttede de sig

for at undersøge muligheden for at tage dele af auktionshusets handler online. Efter et par glas af den dyre rødvin lavede de en brainstorm omkring SWOT og kom frem til følgende uredigerede og ikke prioriterede SWOT:

Strength

- Kendt auktionshus med godt omdømme
- Velkonsoliderede forretningsprocesser (manuelle)
- God styring på (relativt få) data

Weakness

- Meget manuel
- Dårlig udnyttelse af IT

Opportunities

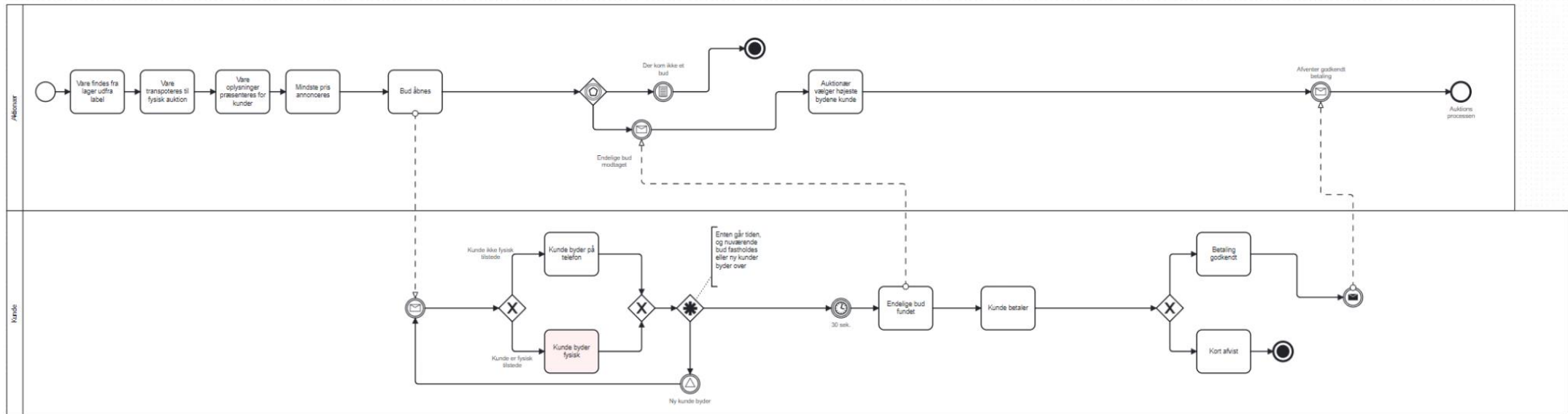
- At flytte auktionshusets brand til en ny form for aktioner

Threats

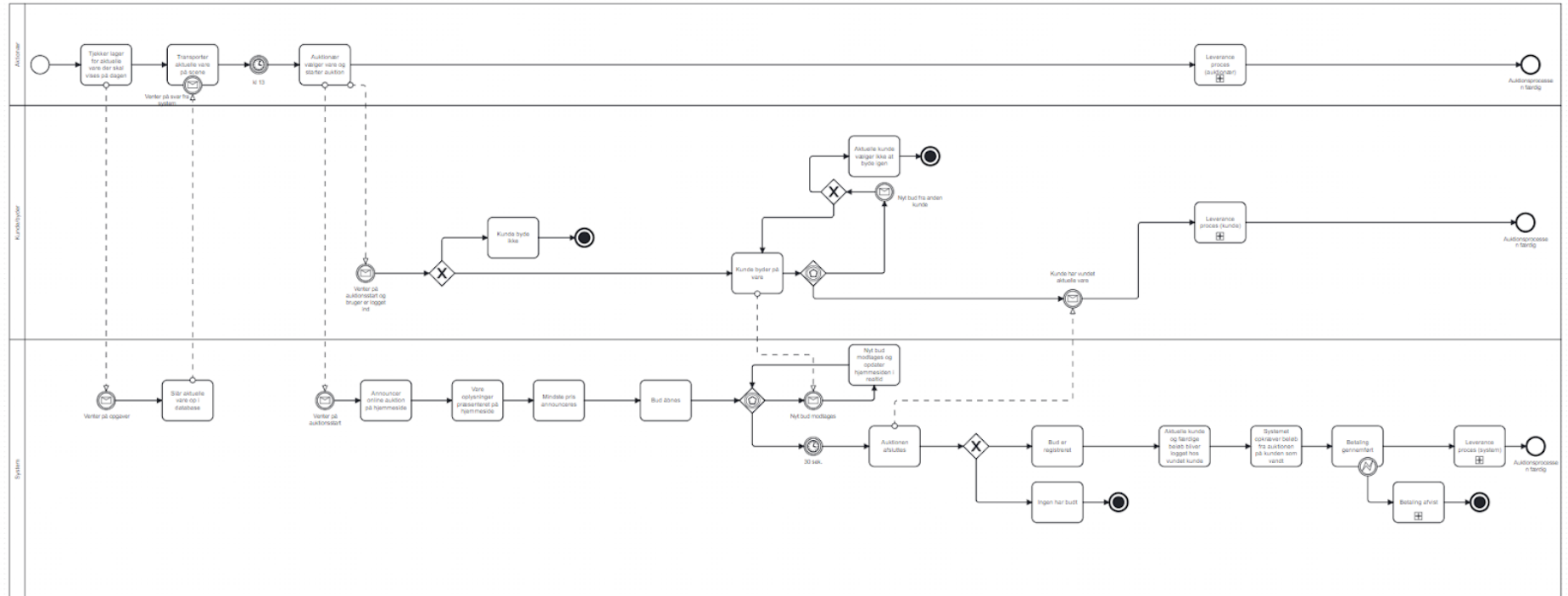
- Konkurrence fra Lauritz.com, DBA, "Gul og Gratis" og Facebook marketplace

Trine og Jan erkendte at auktionshuset manglede en ny strategi og forretningsmodel, hvor man centrerer sig omkring digitalisering og blive digital.

Bilag B - Forretningsprocesser AS-IS:



Bilag C - Forretningsprocesser TO-BE:



Interviews

Interview struktur (spørgsmål)

https://docs.google.com/document/d/1S1m5o5ypxh9MpBh2gm6_WxL284TCd2qm4rq3gdAtFvk/edit?usp=sharing

Interview Deltagere:

ID1

<https://docs.google.com/document/d/1Kg7r54V2GQemylrfokmkMf1OTW09En8-Mg09VuEZWj4/edit?usp=sharing>

ID2

<https://docs.google.com/document/d/1ydsKy2CYBbVg7Dk0FKOKF-aP8xgUAoSXpA9w4m7LwHQ/edit?usp=sharing>

ID3

https://docs.google.com/document/d/1rQv1_0ZEErixv9M6RI5H9vaSdrZK-Cii8z2AAgEGdEw/edit?usp=sharing

ID4

<https://docs.google.com/document/d/1CGx-QB-WLPHCZ99XD3pQbkNeLy6MjVCNZ5AU4kNsdY8/edit?usp=sharing>

ID5

<https://docs.google.com/document/d/1RWDXwFF4GzOvt2YEe3egOlqcPsK6KukltbliPoLaU5Y/edit?usp=sharing>

ID6

https://docs.google.com/document/d/1Si73TVGTqeHj_FZ3jBbe1BdhVPUIcAL1Blmv28UONY/edit?usp=sharing

ID7

<https://docs.google.com/document/d/18xDE2UPxNaXs3YNtLtOYf3IP-0wV4Yer59h14fJHTu8/edit?usp=sharing>

ID8

<https://docs.google.com/document/d/15-AZfNoOux-HtyBcyNy4hMoBZXQevbJXQymfjCQNUd0/edit?usp=sharing>

Skitser & Grænseflade

AS-IS

<https://docs.google.com/document/d/1GMPHnHn8TN39wNAyfjFPVOsGBGG8na5n8NHrVjxrr6A/edit?usp=sharing>

User stories

<https://docs.google.com/document/d/1bsMkx4yb6sjaABd3jWTRJlaNEruKnDmWd8vZZSus3Xw/edit?usp=sharing>

Figma Grænseflade:

<https://www.figma.com/file/SIGsSWIoLKqFOjANJXH0Zk/Bruger-gr%C3%A6nseflade?type=design&node-id=0%3A1&mode=design&t=FB7zq3Nk5unEaZnw-1>

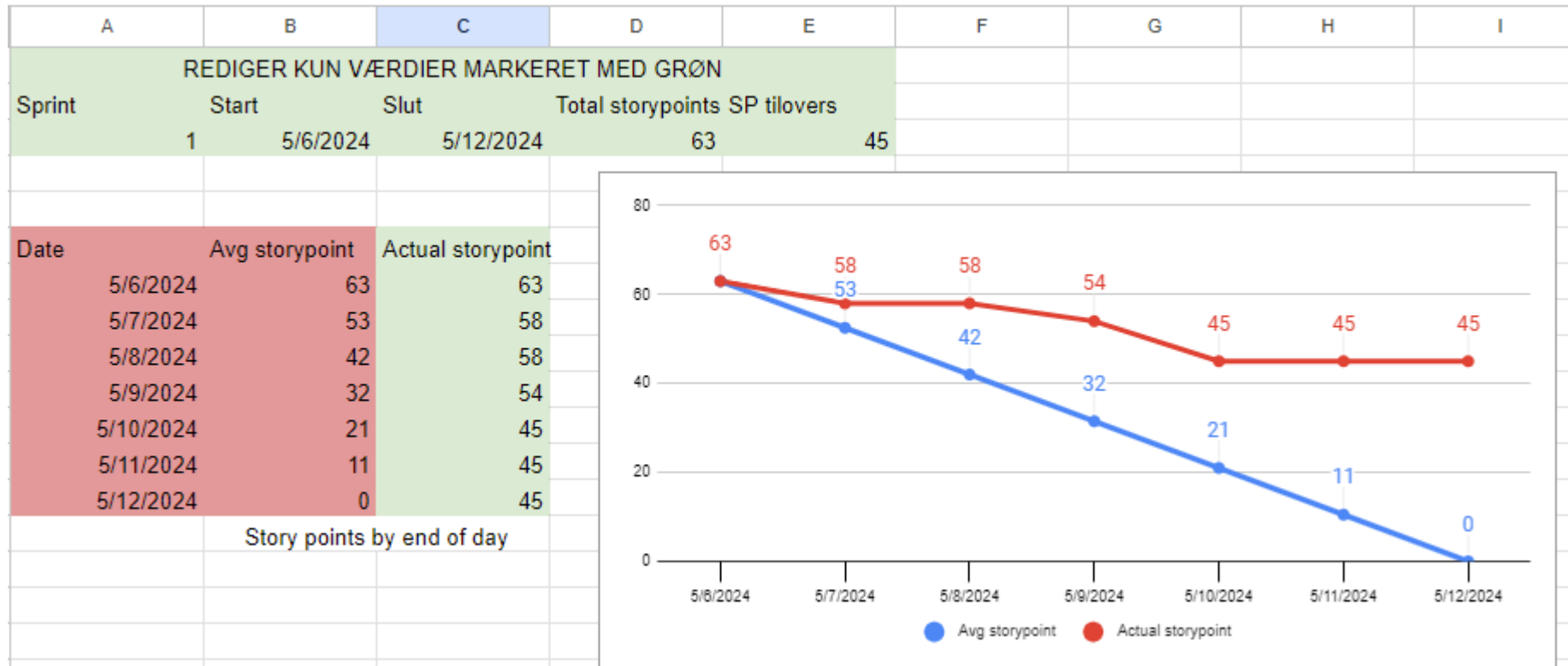
Billeder brugt i Figma-modellen

"Betalings kort" Au2wheels, <https://au2wheels.dk/shop/guide-til-2-13621c1.html>. Accessed 22 May 2024.

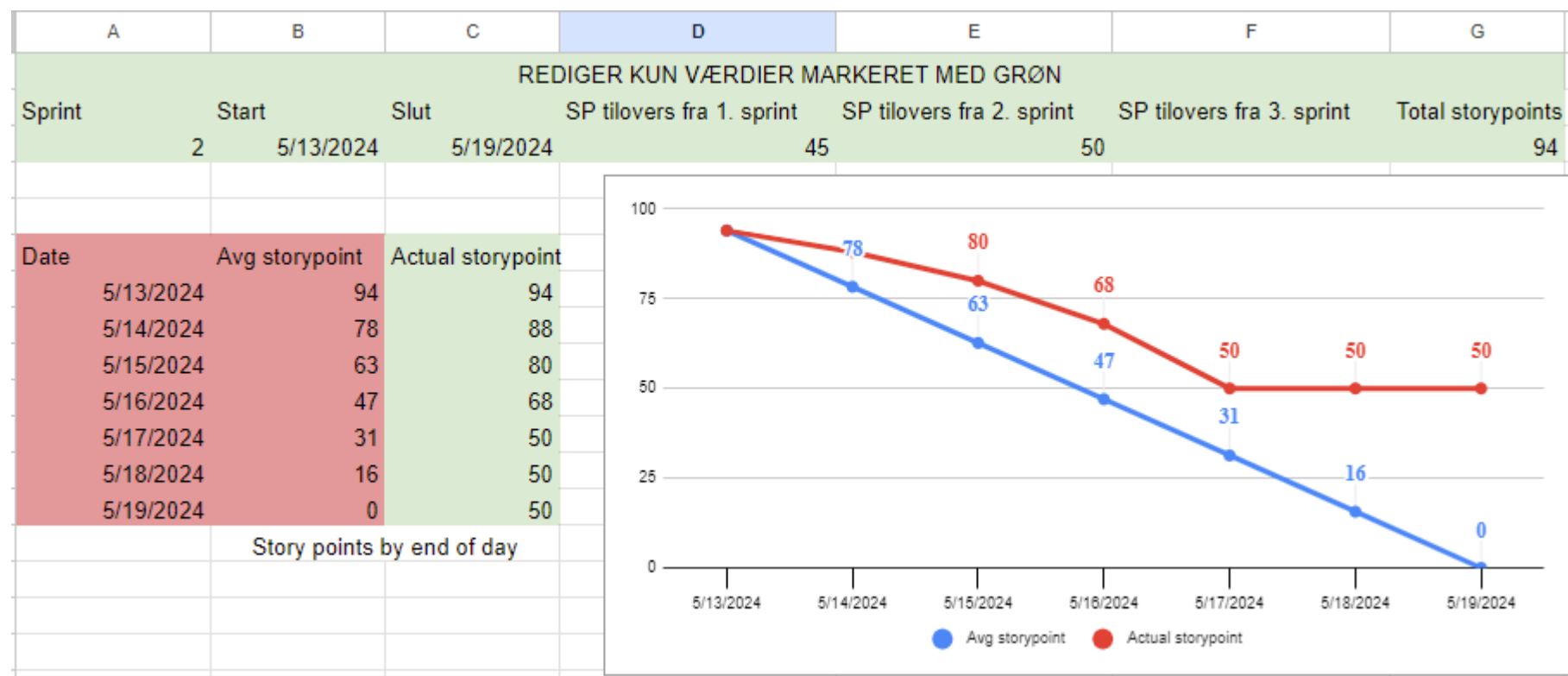
"Vintage Watch." eBay, <https://www.ebay.com/itm/154566360841>. Accessed 15 May 2024.

"Winston Churchill's Half-Finished Cigar Sells at Auction." Independent, 16 Oct. 2017, <https://www.independent.co.uk/news/uk/winston-churchill-cigar-smoked-half-finished-paris-auction-a8000751.html>. Accessed 22 May 2024.

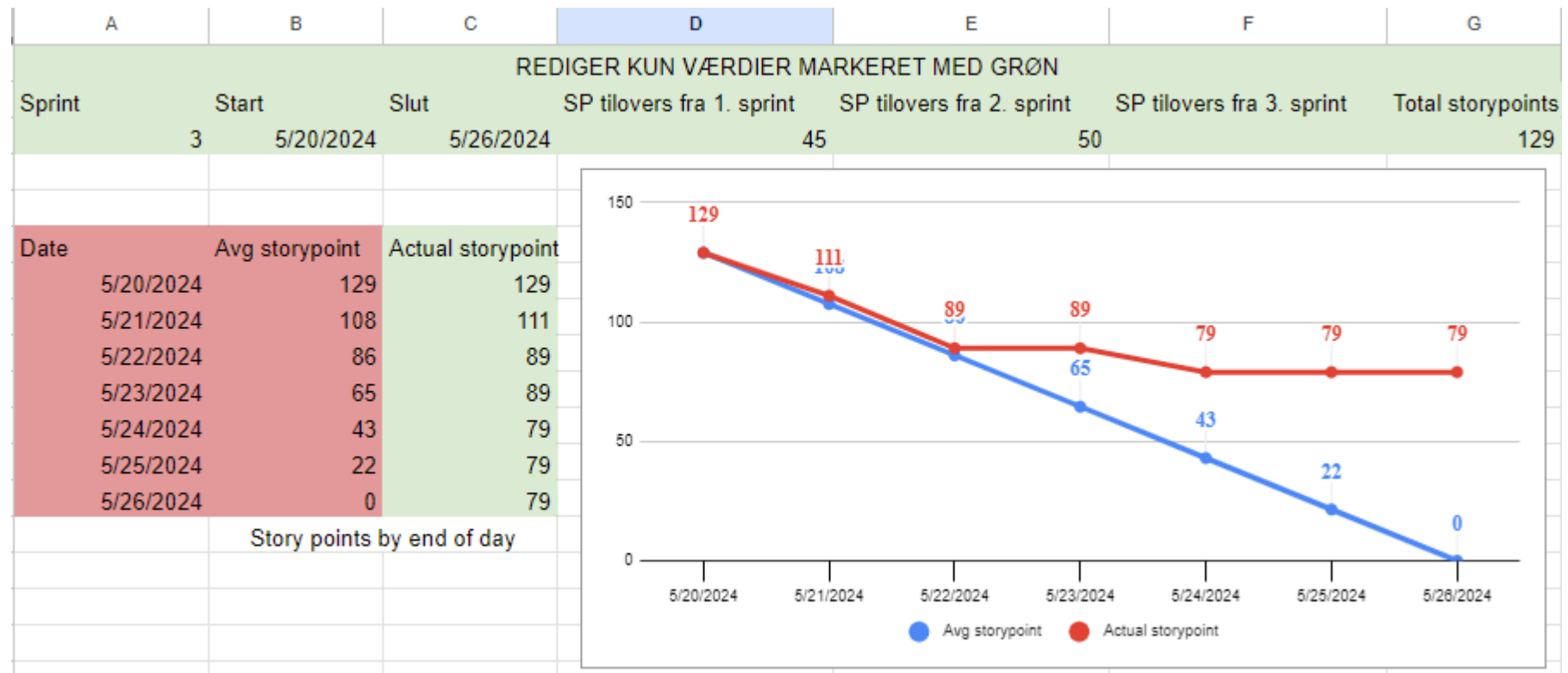
"Volvo P210 Duett B18." Bilweb Auctions, <https://bilwebauctions.se/en/marsauktion-2021/volvo-p210-duett-b18-2788>. Accessed 22 May 2024.



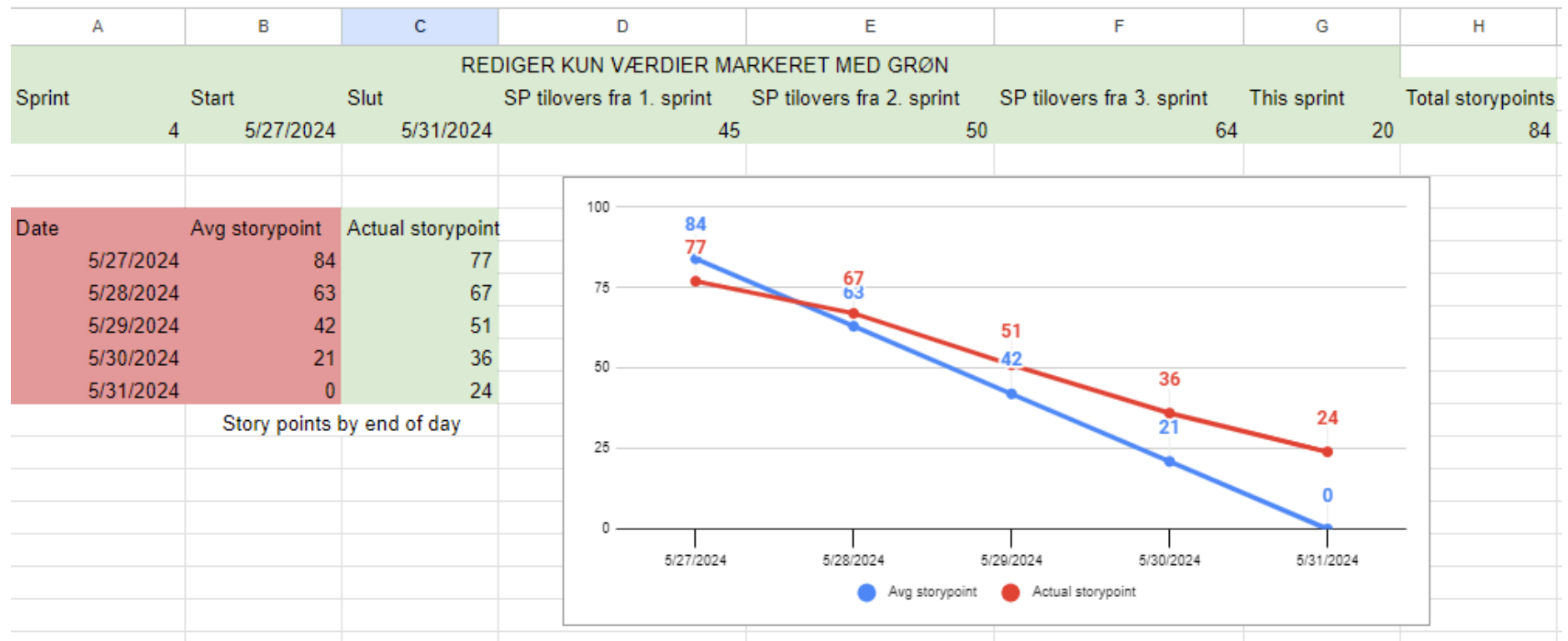
Bilag D – Burndown chart, sprint 1



Bilag E – Burndown chart, sprint 2



Bilag F – Burndown chart, sprint 3



Bilag G – Burndown chart, sprint 4