



UNIVERSITETET I AGDER

IS-304: 2024

Tittel: Bachelorprosjekt

Emnekode	IS-304
Emnenavn	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen
Veileder	Rania Fahim Hassan Ibrahim Elgazzar
Oppdragsgiver:	Capgemini AS

Studenter:

Etternavn	Fornavn
Ømer	Eziz
Nedland	Eirik Skogland
Tveit	Bastian Ege
Zuoza	Dominykas
Huynh	Dung Katrine My
Fiske	Eivind Hustad

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeid uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA <input checked="" type="checkbox"/>	NEI <input type="checkbox"/>
Kan besvarelsen brukes til undervisningsformål?	JA <input checked="" type="checkbox"/>	NEI <input type="checkbox"/>
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA <input checked="" type="checkbox"/>	NEI <input type="checkbox"/>

Gruppe 1



Vår 2024

EcoSpectra

Visualisering av miljø-
fotavtrykk fra byggeplasser

Skrevet av:

Dominykas Zuoza

Eziz Ømer

Eirik Skogland Nedland

Bastian Ege Tveit

Dung Katrine My Huynh

Eivind Hustad Fiske

Capgemini 

Forord

Vi ønsker å takke alle som har bidratt til vårt bachelorprosjekt hos Capgemini. En spesiell takk går til konsulentene som hjalp oss underveis. Tusen takk til Vegard Sætre som ga oss god veiledning innen prosjektgjennomføring. Vi vil også gjerne takke Øyvind Mjølund for hans hjelp med oppsett av møter, brukertesting og presentasjonsteknikk, sammen med Lars Busetth. En stor takk til Kim Ommundsen for hans tekniske veiledning, til Anne Cecilie Rueness for hennes hjelp med oppsett av møter med kunden og opplæring i tidsestimering. Til slutt vil vi takke konsulentene hos Capgemini som var med på intervjuer og tester.

Vi vil også takke Capgemini Kristiansand og deres kunde, Rental Group, for samarbeidet og muligheten til å jobbe med en sentral problemstilling i dagens samfunn. Deres bidrag og veiledning har vært uvurderlig, og vi setter stor pris på den innsikten de har delt med oss. En spesiell takk går også til vår veileder, Rania Elgazzar, for hennes støtte og tilbakemeldinger gjennom hele bachelorprosjektet. Hennes veiledning har vært hjelpsom, og vi verdsetter hennes engasjement.

Dette prosjektet har vært en lærerik og utfordrende reise, og vi er takknemlige for muligheten til å jobbe med en så spennende oppgave. Gjennom dette prosjektet har vi fått praktisk erfaring med teknologi, prosjektstyring og kvalitetssikring. Vi håper at vårt arbeid vil være til nytte for Capgemini og deres kunde.

Sammendrag

Denne rapporten er skrevet i samarbeid med Capgemini Kristiansand og deres kunde, Rental Group. Prosjektets mål er å utvikle en webapplikasjon som skal visualisere miljødata fra utleide anleggsmaskiner. Applikasjonen skal bidra til å identifisere områder hvor Rental Group kan veilede kundene sine til å redusere utslipp og effektivisere arbeid, samt vise frem miljøvennligheten av de forskjellige anleggsmaskinene.

Rapporten dekker gruppens håndtering og utførelse av prosjektet med fokus på de forskjellige fasene som inkluderer analyse, design og utvikling. For å kunne vise framgang og utvikling av produktet underveis, har gruppen gjennomført prosjektet ved å ta i bruk Scrum i en hybrid waterfall-agile arbeidsmetodikk.

Capgemini lot gruppen selv bestemme utvalg av verktøy, og dermed har gruppen vært gjennom flere teknologier i løpet av prosjektet. Rapporten vil gi en detaljert gjennomgang av de verktøyene som ble benyttet i design- og utviklingsprosessen, og forklare hvordan disse bidro til å skape et brukersentrert produkt.

Prosjektet har resultert med en webapplikasjon som viser innsikt, miljøutslipp og effektivitet av Rental Group sine anleggsmaskiner. I tillegg inneholder systemet et detalje-visning både på kunde-, prosjekt- og maskinnivå.

Innholdsfortegnelse

Forord	3
Sammendrag	5
1. Introduksjon	8
1.1 Bakgrunn	9
1.2 Oppgavebeskrivelse og Systemdefinisjon	9
1.3 Problemdomene og Personlig Portal	10
1.4 Proof of Concept (PoC)	11
2. Analyse og Design.....	12
2.1 Analyse	12
2.2 Design	18
2.3 Systemarkitektur	22
3. Teknologi og Verktøy.....	28
3.1. Administreringsverktøy	28
3.2 Backend og Frontend Teknologier	30
4. Prosjektgjennomføring.....	34
4.1. Scrum i en Hybrid Waterfall-Agil metodikk.....	34
4.2 Tidsestimering	37
4.3. Gjennomføring av Scrum.....	41
4.4. Pre-Sprint 1 & Sprint 2 – Analyse, Planlegging og Designstart.....	43
4.5. Sprint 3 & Sprint 4 – Designslutt & Utviklingsstart.....	44
4.6. Sprint 5 til Sprint 8 – Utvikling.....	45
4.7 Kvalitetssikring.....	47
5. Endelig produkt	50
5.1 Funksjonalitet	50
5.2 Veien videre.....	53
6. Refleksjon	54
6.1 Sosialt	54
6.2 Prosjektgjennomføring.....	54
6.3. Utvikling.....	56
6.4. Har gruppen oppnådd kravene?	57
Referanser	58
Appendiks.....	62

Figurliste

Figur 1: FACTOR-tabell	10
Figur 2: PACT-analyse	12
Figur 3: Brukerhistorie 1.....	13
Figur 4: MoSCoW method av TechTarget (Brush, 2023).....	14
Figur 5: Risikoanalyse	15
Figur 6: Risikomatrix	15
Figur 7: Risikonivå fargekoder	15
Figur 8: Risiko tiltak og kommentar	15
Figur 9: Suksesskriterier	16
Figur 10: Eksempel på en PBI med akseptkriterier	17
Figur 11: Wireframe	20
Figur 12: Endelig navigasjonskart.....	20
Figur 13: Mockup - hovedsiden.....	21
Figur 14: Prototypen - hovedsiden.....	21
Figur 15: Tabell for prioritering av designkriterier	23
Figur 16: Systemarkitektur	25
Figur 17: Mappedstruktur	26
Figur 18 - Eksempel på en "@code" blokk	27
Figur 19: Sprint Board.....	29
Figur 20: Azure Repository	29
Figur 21: Infoboks for Maskin Aktivitet - tabell for maskiner for prosjekt.....	31
Figur 22: Miks-graf som viser tidslinje for maskintimer (driftstimer og tomgangstimer).....	32
Figur 23: Visning av antall problemer i koden med Qodana før code cleanup.....	33
Figur 24: Opprinnelig plan av sprintoversikt	36
Figur 25: Gruppens Waterfall-Agile prosjektstruktur.....	37
Figur 26 - Planning poker.....	38
Figur 27: Burndown-chart fra Sprint 5	40
Figur 28: Timeliste per gruppelem for prosjektet	41
Figur 29: Sprint backlog fra Sprint 4.....	42
Figur 30: Daily Scrum fra 9. januar 2024	44
Figur 31: Innsikt-side	50
Figur 32: Miljøfotavtrykk-side	51
Figur 33: Effektivitet-side	51

Figur 34: Kundedetaljer-side	52
Figur 35: Prosjektdetaljer-side	52

Appendiks

Appendiks 1: Risikomatrise	62
Appendiks 2: MoSCoW RG API-er	64
Appendiks 3: Brukertest og undersøkelse.....	66
Appendiks 4: Brukerhistorier med MoSCoW	66
Appendiks 5: Prosjektdagbok	68
Appendiks 6: Møtereferat Styringsmøte 1.....	68
Appendiks 7: Møtereferat Styringsmøte 2.....	69
Appendiks 8: Møtereferat Styringsmøte 3.....	70
Appendiks 9: Endelig wireframe	71
Appendiks 10: Link til Prototype	71
Appendiks 11: Navigasjonskart første utkast.....	72
Appendiks 12: Individuell wireframe Dominykas.....	72
Appendiks 13: Individuell wireframe Eziz.....	73
Appendiks 14: Individuell wireframe Katrine.....	73
Appendiks 15: Individuell wireframe Eirik.....	73
Appendiks 16: Individuell wireframe Bastian.....	73
Appendiks 17: Individuell wireframe Eivind.....	73
Appendiks 18: Mockup	74
Appendiks 19: Prototype i Figma	75
Appendiks 20: Intervjuer med Capgemini.....	76
Appendiks 21: Known issues	82
Appendiks 22: Gruppekontrakt	83
Appendiks 23: Attest fra Capgemini.....	84

Introduksjon

Dette kapitlet introduserer bakgrunnen til prosjektet, interessentene under prosjektet og oppgavebeskrivelsen.

1.1 Bakgrunn

I dette semesteret gjennomførte gruppen et prosjekt i samarbeid med en bedrift. Formålet med dette prosjektet er å bruke kunnskaper og erfaringer de har lært fra studieløpet og anvende det i arbeidslivet. Gruppen skal ha planlagt, estimert, gjennomført, testet, fulgt opp og dokumentert prosjektet (Universitetet i Agder, n.d.).

Gruppen er i samarbeid med Capgemini, som er et av de ledende konsultentselskaper globalt innen informasjonsteknologi. Det er en mangfoldig bedrift som sprer seg over 50 land, og har som hensikt «å frigjøre menneskelig energi gjennom teknologi for en inkluderende og bærekraftig fremtid»

- Aiman Ezzat, CEO Capgemini (Capgemini, u.d.).

I Norge har Capgemini rundt 1600 ansatte fordelt på flere kontor i landet. Avdelingen i Kristiansand ble etablert i 2021, og har som mål å bli en av de ledende konsulentvirksomhetene på Sørlandet (Mjølund, 2024).

Capgemini samarbeider tett med Rental Group AS, heretter referert som RG, som er en sentral aktør innen utleie av maskiner og utstyr til anleggsbransjen. RG ivaretar alle behov, fra både små og store anleggsprosjekter (Rental Group, n.d.). Konsernet består av spesialiserte selskaper med høy kompetanse og stort volum av tunge anleggsmaskiner, tårnkraner og spesialutstyr til anleggsbransjen (Rental Group, n.d.).

RG har satt seg et ambisiøst mål som er å redusere sitt utslipp med 50% innen 2025 og bli utslippsfrie innen 2050 (Rental Group, 2023). Derfor ønsker ledelsen å få en klar oversikt over sitt miljøfotavtrykk for å identifisere områder hvor de kan redusere forbruket. Dette er viktig, siden hvis man ikke kan kartlegge problemet, hvordan kan man løse det?

1.2 Oppgavebeskrivelse og Systemdefinisjon

Oppgaven var opprinnelig å utvikle et Proof of Concept som kunne presenteres for RG. Prosjektet ble lagt frem åpent, slik at det var opp til gruppen å identifisere anvendelsesområder, vise kreativitet og

utvikle løsningen. Gruppen skulle utvikle en Minimum Viable Product, heretter MVP, som skulle vise kreative visualiseringer av RG sitt miljøfotavtrykk.

En systemdefinisjon er en presis beskrivelse av et system uttrykt i et naturlig språk (Mathiassen et. al., 2018, s. 24). Dette var viktig for gruppen å definere for å identifisere kravene i systemet som gruppen skulle utvikle. Formålet med systemdefinisjonen var å holde den forståelig for gruppen, samt interessentene, uten å ha for mange tekniske begreper.

Under utviklingen av systemdefinisjonen tok gruppen i bruk FACTOR-tabellen. Tabellen består av funksjonalitet, applikasjonsdomene, betingelser, teknologier, objekter og ansvar (Mathiassen et. al., 2018, s. 40). Gruppen brukte FACTOR-tabellen for å utvikle en systemdefinisjon som dekker disse områdene.

Factor tabell		
FACTOR	Definisjon	Kriterier
Functionality	Beskriver hvilke funksjoner systemet består av, og hvordan de støtter opp om de oppgavene som skal løses i applikasjonsdomenet	Et system visualiserer, sorterer og filtrerer sensordata hentet fra databasen fra RentalGroup
Application Domain	Beskriver den delen av systemet som administrerer, monitorerer, og kontrollerer et problemomene.	Ansatte (økonomiansvarlig, miljødirektør, ledere etc.) i Rental Group
Conditions	Omhandler hvilke forhold systemet blir utviklet og brukt under.	Utviklet av et team som har kontakt med en produkteier fra Capgemini og RG som sluttbrukere
Technology	Omhandler hvilke teknologier som brukes til å utvikle systemet, men også teknologien som systemet vil kjøre på.	ASP.NET, BLAZOR, C#, MYSQL, .NET, Mudblazor, Docker, Syncfusion, Visual studio, PC-enhet
Objects	Problemomene vil bestå av alle systemets objekter.	Sensordata fra maskiner, statistikkhenvisninger, grafer, tabeller, bilder
Responsibility	Representerer det overordnede ansvaret som systemet skal utføre.	Systemet skal bidra med å identifisere områder hvor RG sine kunder kan redusere sitt forbruk og utslipp. Systemet skal visualisere dette på en forståelig måte.

Figur 1: FACTOR-tabell

I tillegg til dette, hadde gruppen interne diskusjoner og møter med produktansvarlig fra Capgemini, hvor følgende systemdefinisjon ble fastsatt:

«En webapplikasjon som bruker sensordata fra anleggsmaskiner. Applikasjonen skal illustrere statistikk og data som ansatte i RG kan benytte for å ta bedre beslutninger som reduserer utslipp og fremmer utslippsfrie byggeplasser.»

1.3 Problemomene og Personlig Portal

Som nevnt tidligere i 1.1 har RG som mål å redusere sitt utslipp med 50% innen 2025 og i tillegg oppnå utslippsfrie byggeplasser innen 2050. Fra og med 2023 har de startet å tilby elektriske maskiner med arbeidstid på opptil 12 timer per ladeperiode (Rental Group, 2023). Tilbudet er i en kontinuerlig utviklingsfase, med en stadig utvidelse av sitt utvalg av elektriske maskiner. Capgemini har allerede utviklet en kundeportal for RG sine kunder, kalt IDA.

Gjennom IDA får RG sine kunder tilgang til detaljert informasjon om maskinenes brukstid, utslipp, lokasjon og annet relevant data (Rental Group, u.d.). Per dags dato er portalen eksklusivt tilgjengelig for kundene, noe som har resultert i et behov for en intern portal for å overvåke utstyret som er utleid.

Derfor er det ønskelig med en overordnet oversikt over CO2-utslipp fra både kunder og prosjekter, samt sammenligninger mellom diesel- og elektriske maskiner for å fremme en mer bærekraftig anleggsbransje. Dette arbeidet er satt i gang av gruppen i utgangspunktet som en Proof of Concept, et konsept som er nærmere forklart i neste underkapittel.

1.4 Proof of Concept (PoC)

Proof of Concept, heretter PoC, er en demonstrasjon for å evaluere realiserbarheten av en produktide (Gillis, 2023). PoC fokuserer på å utforske gjennomførbarheten og skalerbarheten av en idé, i motsetning til å være det mest effektive eller beste produktet. Oppgaven gitt av Capgemini var å utvikle en PoC med fokus på RG sitt miljøfotavtrykk. Teoretisk sett kunne gruppen ha levert en Figma-prototype, forklart i underkapittel 3.4.2, med konsepter som møtte oppdragsgiverens krav. Gruppen valgte å ikke begrense seg til minstekravene, siden de mente at dette ikke ville gi tilstrekkelig læringsutbytte.

Interne diskusjoner med produkteier fra Capgemini ble gjennomført i Pre-Sprint og delvis i Sprint 2. Gruppen besluttet at det å både designe og utvikle systemet ville demonstrere konseptene, gi bedre læringsutbytte, og legge bedre grunnlag for videreutvikling dersom kunden ble fornøyd med produktet. Dette innebar krav om å opprettholde høye kodestandarder, god dokumentasjon, og en kritisk tilnærming til valg av programvarepakker. Med økte krav til prosjektet fulgte også behovet for solid prosjektstyring og bruk av smidige metodikker.

1. Analyse og Design

Dette kapittelet tar for seg analyse og design. Gjennom prosessene ble det oppnådd en mer omfattende forståelse av prosjektet og potensielle risikoer.

2.1 Analyse

Gruppen konkluderte at en tydelig oppgavebeskrivelse var nødvendig for å oppnå en effektiv fremdrift. Ved å få en fellesforståelse om oppgavebeskrivelsen hjalp det gruppen med å forstå hva som var forventet, dette førte både til mer sikkerhet og effektivitet.

2.1.1 Intervjuer

Gruppen hadde jevnlig møter med produkteier for å komme frem til en oppgavebeskrivelse hvor det ble avklart hvilke forventinger og ønsker produkteier hadde for prosjektet. Ut fra disse møtene kom det frem at brukerorientert design for prosjektet hadde en høy prioritering. Det går ut på at man tar i bruk sluttbrukere i designprosessen for å sikre direkte dialog og innspill angående brukerens behov (Lowdermilk, 2013).

Under analysefasen ble det brukt ustrukturerte intervjuer for å samle inn data. Ustrukturerte Intervjuer bruker ikke forhåndsdefinerte spørsmål, og gjør det mulig å stille åpne spørsmål (Zhao, 2024). Ved å gjennomføre ustrukturerte intervjuer benyttet gruppen seg av kvalitativ datainnsamling, noe som tillot dem å stille varierte og åpne spørsmål (Utdanningsdirektoratet, n.d.). Dette bidro til å gi gruppen et bredere perspektiv på brukernes behov.

2.1.2 PACT-analyse

Under analyseprosessen av systemet, har gruppen bestemt å gjøre en PACT-analyse. Analysen er definert som et rammeverk for å designe brukersentrerte og interaktive systemer. Det er “en essensiell del av vår framgangsmåte for brukerdessign er at man putter brukerne først” (Benyon, 2019, s. 25). Gruppen valgte å ta i bruk PACT for å utvikle et system rettet mot brukeren. Analysen består av People, Activities, Context og Technology.

PACT hjalp gruppen å forstå hvilke type mennesker som skal bruke systemet, konteksten for aktivitetene, og selve aktivitetene som brukerne skal utføre på systemet. Til slutt går dette gjennom hvilke teknologier som brukeren utnytter for å utføre aktivitetene på systemet.

People	Ledelsen, økonomi, kontormedarbeidere
Activity	Se miljø og bærekraft data, hente grafer til presentasjoner, se ytelsen til maskinene
Context	For å få informasjon til å begrunne beslutninger
Technology	PC, mobil, web browser, webapplikasjon

Figur 2: PACT-analyse

2.1.3 Brukerhistorier og MoSCoW

Ut fra PACT-analysen lagde gruppen brukerhistorier, som ble brukt for å forklare egenskapene til et system fra brukerens perspektiv. Brukerhistoriene hjalp gruppen med å forstå brukerne sine behov og forventninger (GfG, 2023). Figur 3 viser et eksempel på hvordan gruppen satt opp brukerhistoriene.

#1	Prioritering: Must have
Brukerhistorie	Som en miljødirektør i Rental Group vil jeg se gjennom en sortert liste over de maskinene som lager størst utslipp, for å vite hva den beste måten å redusere vårt utslipp på er.
Funksjon	Se gjennom og sortere maskiner etter utslipp. (Liste med sortering)
Kriterier	Visning av maskiner, og sorteringsmuligheter etter utslipp.
Test-scenario	<ol style="list-style-type: none">1. Logge inn2. Navigere til statistikk siden3. Sortere maskiner etter utslipp
Argumentasjon	Dette er svært viktig for å kunne vite hva det beste steget er for å redusere utslipp, er det viktig å vite de mest forurensende maskinene.

Figur 3: Brukerhistorie 1

Videre tok gruppen i bruk MoSCoW for å prioritere de forskjellige brukerhistoriene. MoSCoW-analyse er et prioriteringssystem hvor funksjoner blir rangert som *must have*, *should have*, *could have* eller *won't have*. Funksjoner som er kritiske for at systemet skal fungere blir rangert som *Must have*. *Should have* er funksjoner som er viktige, men ikke kritiske for at systemet skal fungere. *Could have* er funksjoner som er mindre viktige. *Won't have* er funksjoner som systemet ikke trenger (Benyon, 2019, ss. 148-149).

MoSCoW prioritization



Figur 4: MoSCoW method av TechTarget (Brush, 2023)

Forklaringen av MoSCoW er vist i figur 4 og oversikten over MoSCoW-analysen av brukerhistoriene er vist i appendiks 4. Dette hjalp gruppen med en klar prioritering som førte til at de viktigste funksjonene ble implementert først (Atlassian, 2023). MoSCoW effektiviserte beslutningsprosessene videre i utviklingen etter som at gruppen tok beslutninger basert på denne prioriteringen (Atlassian, 2023).

2.1.4 Risikoanalyse

En risikoanalyse er en prosess hvor en kartlegger mulige risikoer angående en aktivitet ved å beskrive mulige hendelser, årsaken til dem, konsekvenser, og sannsynligheten for at de skal skje (Aven, 2023). Ved å ta i bruk en risikoanalyse fikk gruppen oversikt over mulige risikoer i prosjektet. Med dette kunne gruppen sette definerte tiltak for å begrense både sannsynligheten at en risiko forekommer, og negative konsekvenser som følge av den.

De forskjellige risikoene ble vurdert ved å se på sannsynlighet og konsekvens på et nivå fra 1 til 5. Deretter konverterte gruppen nivået til det samsvarende risikonivået, hvor nivå 1 var svært lav og nivå 5 var svært høy som illustrert i figur 5.

Risikoanalyse før tiltak					
Nr.	Usikkerhet	Type	Sannsynlighet	Negative konsekvenser	Status
1	Får ikke kontakt med bedrift	Risiko	Lav	Høy	Betydelig risiko
2	Prosjektet tilfredstiller ikke behovene til produkteier	Risiko	Lav	Svært lav	Ingen betydelig risiko
3	Lav kodekvalitet	Risiko	Høy	Høy	Høy risiko
4	En uengasjert gruppemedlem bidrar lite	Risiko	Lav	Lav	Har lite risiko
5	Flere uengasjerte gruppemedlemmer bidrar lite	Risiko	Svært lav	Høy	Har lite risiko
6	Data fra databasen lekker ut	Risiko	Svært lav	Svært høy	Betydelig risiko
7	Klarer ikke å implementere noen funksjoner	Risiko	-	Lav	-
8	Klarer ikke å implementere alle funksjoner	Risiko	-	Svært lav	-
9	Dårlig tidsestimering	Risiko	Lav	Middels	Betydelig risiko
10	Prosjektet blir kansellert midt i semesteret	Risiko	Svært lav	Høy	Har lite risiko
11	Dårlig kommunikasjon	Risiko	Lav	Høy	Betydelig risiko
12	Dårlig optimalisering av kode	Risiko	Middels	Middels	Betydelig risiko
13	Systemet er for ressurs-intensiv	Risiko	Middels	Middels	Betydelig risiko

Figur 5: Risikoanalyse

Ut fra disse ble risikoene lagt inn i en matrise, som illustrerte hvilke statuser de forskjellige risikoene gikk under som ga mulighet til å bestemme tiltakene som skulle gjøres mot dem. En slik matrise vises nedenfor i figur 6.

Risikomatrix før tiltak						
Sannsynlighet	Svært høy					
	Høy					
	Middels					
	Lav					
	Svært lav					
		Svært lav	Lav	Middels	Høy	Svært høy
Konsekvenser						

Figur 6: Risikomatrix

Avhengig av hvilken farge de forskjellige risikoene endte opp på, skulle gruppen vurdere å sette inn tiltak. Tiltakene som skulle iverksettes ble bestemt ut fra matrisen nedenfor i figur 7. Målet var at det skulle være ingen risikoer på Lys Rød eller over etter tiltakene.

Tiltak	
Grønn	Ingen betydelig risiko
Lys Grønn	Har lite risiko (vurder tiltak)
Gul	Betydelig risiko (Risikoreducerende tiltak iverksettes hvis praktisk mulig)
Lys Rød	Høy risiko (start å iverksette tiltak)
Rød	Uakseptabelt (Risikoreducerende tiltak er påkrevd)

Figur 7: Risikonivå fargekoder

Tiltakene ble loggført i en tabell som er illustrert under i figur 8. Effekten av disse tiltakene ble vanligvis vurdert etter en uke, for å bedre kartlegge hvordan tiltaket har påvirket risikoen. Dette er fordi gruppen synes de hadde lite kjennskap til hvordan et tiltak skulle forandre risikobildet umiddelbart.

Risiko før	Tiltak	Kommentar	Risiko etter
	Kontakte flere personer innad bedriften for svar	X	
	Holde tett kontakt med produkteier og kontaktperson	Grunnet tiltakene, vil sannsynligheten til at produktet ikke tilfredstiller kundens behov gå ned, og samtidig vil konsekvensene være mindre fordi gruppen kan være sikker at de i det minste oppfyller noen av kravene.	
	Sette kode standarder, kvalitetsjekk før merge og Code review. I tillegg til refaktorering vha. Analyseverktøy	I det siste har vi gjort en kodeanalyse med forskjellige verktøy og sett at kodekvaliteten er mye dårligere enn det vi hadde sett før. Derfor ble resten av den gjeldende sprinten satt av til bugfixes og forbedring av kode	
	Tilpasse oppgaver slik at vedkommende jobber med noe de liker å gjøre.	X	
	Teambuilding med gruppa skaper tilhørighet, skaper åpenhet og tillit, samt øker moral	X	

Figur 8: Risiko tiltak og kommentar

I starten av hver sprint tok gruppen en ny risikovurdering, la inn tiltak på risikoer som var uakseptable, og vurderte hvor effektive tidligere tiltak var for å legge dette inn i en ny risikomatrix rundt slutten av hver sprint. Ved å revidere risikoanalysen ofte med å oppdatere, fjerne eller legge til nye risikoer, sikrer gruppen at kvaliteten blir opprettholdt gjennom hele prosjektet.

2.1.5 Kvalitet og Kvalitetssikring

Her definerer gruppen kvalitet gjeldende for prosjektet, samt definerer gruppen områder som krever kvalitetssikring.

Definere kvalitet

Siden det er flere måter å definere kvalitet på, er det viktig å knytte begrepet til noe spesifikt for å gi mening (Frich, 2011). Etter flere intervjuer med Capgemini understrekte de viktigheten av at produktet eller tjenesten skal oppfylle brukerens krav og forventninger. Kvalitetssikring refererer til de planlagte aktivitetene og tiltak som utføres for å sikre at et produkt eller en tjeneste oppfyller de kravene som er satt av produkteieren (Halbo, 2020). Basert på dette, mener gruppen at kvalitet oppnås ved at produktet oppnår ønskede krav fra produkteieren og at kundene er fornøyde med produktet. Gruppen vil også vektlegge god kodekvalitet i systemet siden produktet skal overleveres til Capgemini. Med hensyn til dette, har gruppen definert to områder for kvalitet og kvalitetssikring;

1. Produktet tilfredsstiller både produkteier og brukerne/kunden
2. God kodekvalitet i systemet

Produktet tilfredsstiller både produkteier og brukerne/kunden

I dette prosjektet er interessentene Capgemini og RG. Gruppen definerte suksesskriterier tidlig i prosjektet. Dette ble gjennomført for å sikre at produktet dekker ønskede krav fra produkteieren i Capgemini.

Beskrivelse av suksesskriterier
Framvise miljødataen på flere ulike grafer på en forståelig måte
Grafene skulle vise data om utslipp, forbruk av strøm/drivstoff
Sammenlikne utslipp mellom kundene for Rental Group
Sammenlikne utslipp mellom ulike maskiner – se på lasteeffektivitet, effektivitet per liter drivstoff forbruk, effektivitet per tonn CO2
Få til minst en Minimum Viable Product (MVP), som f.eks. en prototype med hardkodete elementer

Figur 9: Suksesskriterier

Ved å ha definerte suksesskriterier, kunne gruppen fokusere på områder som sikrer at produktet dekker forventningene fra produkteier. I tillegg benyttet gruppen systemdefinisjon og brukerhistorier slik at gruppen hadde retningslinjer de kunne følge gjennom prosjektet. Systemdefinisjonen gav gruppen et overordnet bilde av produktets formål, mens brukerhistoriene gikk grundigere gjennom funksjonaliteten knyttet til brukernes behov og ønsker i systemet. Dette bidro til å sikre at gruppen utviklet et system som er brukersentrert, samt dekker kravene fra produkteieren.

Gruppen baserte suksesskriteriene og akseptkriteriene fra brukerhistoriene, og lagde skreddersydde features og PBI-er vist i figuren under. Ved å gjøre dette, sikret gruppen at oppgavene dekket suksesskriteriene for prosjektet, samt unngikk manglende funksjonalitet i systemet.

PRODUCT BACKLOG ITEM 279*

279 Maskin inaktiv/aktivInfoboks

Dung Katrine My Huynh 1 Comment Add Tag Save and Close Follow

State: Done Area: EcoSpectra Reason: Work finished Iteration: EcoSpectra/Sprint 7 Updated by Dung Katrine My Huynh: Mar 25

Details 7 0

Vise maskinaktivitet for prosjekter
Sammenheng:

- Timer leid, timer i bruk, prosent av tid leid ut som maskin har vært i bruk

Acceptance Criteria

- Hente ut data; Timer leid, timer i bruk, prosent av tid leid ut som maskin har vært i bruk
- Pop-up for valg av prosjekter
- Mulighet for å filtrere etter prosjekt
- Mulighet for å filtrere etter kunde

Priority: 2
Effort: 1
Business Value: 1
Value area: Business

To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Development

Add link

Pull Request

44 Made machine activity info...
1 Inlared 1 Completed

Figur 10: Eksempel på en PBI med akseptkriterier

Siden RG er sluttbrukerne av systemet, har gruppen forsøkt å involvere dem gjennom hele utviklingsprosessen. Det at brukerne er fornøyde med produktet, bidrar til god kvalitet, uansett om produktet mangler funksjonalitet. Selv om et produkt er teknisk solid, innovativt og sikkert, vil det være meningsløst hvis brukerne ikke er fornøyde med produktet. Gruppen gjennomførte flere intervjuer med ansatte i Capgemini hvor intervjuobjektene understrekte hvor viktig dette var. Intervjuene er i appendiks 20.

God kodekvalitet i systemet

I begynnelsen av prosjektet diskuterte gruppen hvordan de tenkte å utvikle et system med best mulig kode – kode som er intuitivt, effektivt, skalerbart og med god dokumentasjon. Gruppen satte dårlig kodekvalitet som en middels risiko tidlig i prosjektet og lagde tiltak for dette. Ettersom at gruppen kom videre inn i prosjektet opplevde de at kodekvaliteten ble verre grunnet økende progresjon i utviklingsprosessen. Selv om alle gruppemedlemmene hadde samme opplæring fra dette studiet, hadde alle ulik kodestruktur. Dette gjorde at dårlig kodekvalitet økte fra middels til høy risiko, og da innså gruppen at tiltak måtte iverksettes for å opprettholde og ivareta god kodekvalitet i systemet.

2.2 Design

Gruppen sin designprosess gikk ut på å forstå, visualisere og definere prosjektet. Dette førte til et visuelt eksempel som gruppen kunne presentere til produkteier og få tilbakemelding på. Systemets design var iterativt, noe som gjorde at gruppen ofte gikk tilbake til design senere i utviklingsprosessen etter tilbakemeldinger basert på brukertester.

2.2.1 Designprinsipper

Designprinsippene er retningslinjer som skal hjelpe utviklere ved å sette standarder som kan bli brukt for å evaluere og kritisere designet (Benyon, 2019, s. 116). Fordelene ved å følge generelle designprinsipper er at de vil hjelpe gruppen med å etablere en god start på prosjektet, og gjøre det lettere for brukerne å lære seg systemet (Derrick, 2020). Gruppen har tatt utgangspunkt i Benyons 12 designprinsipper.

Basert på funnene i analysefasen har gruppen gjort et valg på hvilke prinsipper som bør prioriteres høyere enn de andre. Neste del av dette kapitlet vil forklare prioriteringen av prinsippene.

Veldig viktig

Visibility er sentralt for å forbedre brukerinteraksjonen, og legger vekt på klar fremstilling av funksjoner og pågående prosesser (Benyon, 2019, s. 117). Mangel på synlige designelementer kan betydelig redusere brukervennligheten.

Affordance handler om å gjøre det intuitivt for brukeren om hva formålet med et element er, for eksempel at en knapp logger deg ut av nettside, og knapper fungerer som knapper (Benyon, 2019, s. 117). Prinsippet prioriteres høyt fordi det er viktig å lette læringskurven for brukerne.

Navigation spiller en avgjørende rolle i brukerengasjement, og veileder utforskning gjennom ulike systemkomponenter som retninger, navigasjonsbar og trykkbare knapper (Benyon, 2019, s. 117). Effektiv navigasjon er fremhevet som en nøkkelprioritet for å sikre en god brukeropplevelse.

Viktig

Consistency i språk, designfunksjoner og systemets overordnede logikk er avgjørende for brukerforståelse. Ved å opprettholde sammenheng, unngår brukere forvirring og fremmer et konsekvent miljø. Dette gjør læringsprosessen enklere for brukerne (Benyon, 2019, s. 117).

Familiarity handler om å ta i bruk kjente begreper og symboler for å gjøre systemet mer velkjent til brukerne (Benyon, 2019, s. 117). Dette prinsippet ble prioritert som viktig siden dette er med på å skape trygghet hos brukeren og gjør det lettere for brukerne å lære systemet.

Flexibility refererer til systemets mulighet på å gjøre samme funksjon på ulike måter (Benyon, 2019, s. 117). Hvis brukeren ikke husker prosjektnavnet, men kjenner kundenavnet, bør systemet tilby alternative navigasjonsmetoder, som å bruke en liste over kundens prosjekter for å velge prosjektet og vise tilknyttet informasjon.

Style setter hensyn til systemets estetikk. Prinsippet går ut på å imøtekomme forskjellige brukerpreferanser og forbedre den samlede brukeropplevelsen (Benyon, 2019, s. 117).

Mindre viktig

Control fokuserer på å definere hvem eller hva som har kontroll, og gir brukerne mulighet til å gjennomføre denne kontrollen (Benyon, 2019, s. 117). I prosjektet er kontroll ansett som mindre viktig ettersom det er valgt å benytte én enkel rolle, nemlig ansatte hos RG.

Feedback handler om å gi tilbakemelding fra systemet til brukerne slik at de vet om det de har gjort er bra eller dårlig (Benyon, 2019, s. 117). Det er ikke mange muligheter til å få tilbakemelding i systemet med unntak av når noe gikk gale med å laste inn data, eller at en knapp som er klikket har ført til en handling som er synlig.

Conviviality innebærer at systemet skal være hyggelig å bruke. Det vil si at tilbakemeldingene fra systemet skal være lett å forstå, ikke inneholde unødvendig språk eller legge byrden av feilen på brukeren (Benyon, 2019, s. 118). **Recovery** går ut på å kunne gjenopprette eller gå raskt tilbake hvis en bruker har gjort en feil (Benyon, 2019, s. 117). **Constraints** er med på å sette tydelige rammer på systemet slik at brukeren ikke gjør noe som ikke er deres intensjon (Benyon, 2019, s. 117).

Conviviality, recovery, og constraint ble ansett som mindre relevante for denne applikasjonen på grunn av at brukerne har ingen tilgang til å endre data på applikasjonen, og sjansen for at de begår feil som fører til irreversible skader er liten.

2.2.2 Wireframes

Low-fidelity prototype er en type prototype som fokuserer på de underliggende detaljene som oppsettet til sidene, strukturen, og navigering. Fordelen med low-fidelity prototyper er at de er kjappe å lage, gir grunnleggende forståelse av hvordan produktet kommer til å se ut, og kan endres raskt (Benyon, 2019, s. 199). Wireframe er en type low-fidelity prototype som fokuserer på hvordan strukturen til en applikasjon vil se ut ved å fokusere på oppsettet, istedenfor å fokusere på mer detaljerte sider ved designet som for eksempel fargevalg og funksjonalitet (Benyon, 2019, ss. 194-195).

Hvert gruppelem lagde en wireframe, som vises i appendiks 12-17. Til slutt ble det laget en endelig wireframe basert på de elementene som gruppen likte best. Grunnen til dette var fordi

gruppen designet webapplikasjonen helt fra bunn av, og derfor fikk hvert gruppemedlem vist sin ide av hvordan de så for seg systemet uten innflytelse fra de andre gruppemedlemmene. Den endelige wireframen som kan bli sett i figur 11 og appendiks 9, ble grunnlaget for oppsettet til sidene i første utkast av mockupen.

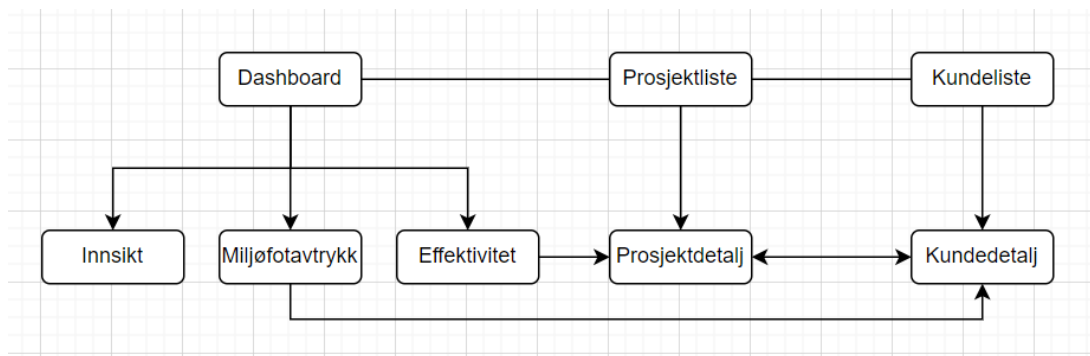


Figur 11: Wireframe

2.2.3 Navigasjonskart

Navigasjonskart er et diagram som viser navigasjonsstrukturen til et system (Benyon, 2019, s. 191). Første utkast av navigasjonskartet som kan sees i appendiks 11, ble laget samtidig som den endelige wireframen. Navigasjonskartet ble endret etter hvert som mockupen ble endret.

Siste utkast av navigasjonskartet kan sees i figur 12 og er en forenklet versjon. Basert på tilbakemeldinger fra brukertester var det et ønske om at det skulle være mulig å klikke på grafene for å se mer kontekst til dataen som vises frem. Dette førte til at det er mulig å navigere seg frem til prosjektdetalj eller kundedetalj siden via enkelte grafer som er i Dashboard.



Figur 12: Endelig navigasjonskart

2.2.4 Mockup

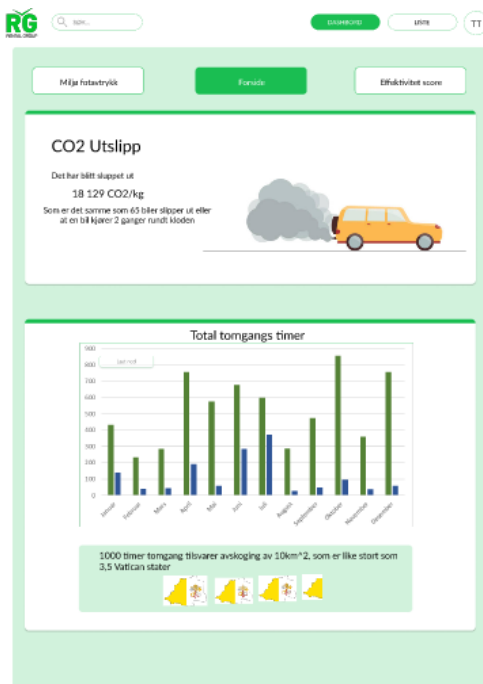
Mockup er en type low-fidelity prototype, som har fokus på å visualisere hvordan det endelige produktet kommer til å se ut (Coursera, 2023). Mens wireframe hadde bare fokus på struktur og



Figur 13: Mockup - hovedsiden

2.2.5 Prototype

High-fidelity prototype er av høyere kvalitet enn low-fidelity prototype, og har fokus på det visuelle, innholdet og funksjonaliteten til systemet (Benyon, 2019, s. 198). Den endelige prototypen skal være så likt som mulig den endelige webapplikasjonen i både utseende og funksjonalitet.



Figur 14: Prototypen - hovedsiden

oppsett til nettsiden, har mockups også fokus på selve designet til nettsiden. I motsetning til prototypen er mockupen en statisk representasjon av nettsiden (Coursera, 2023).

Etter at mockupen var laget ble den vist til interessenter, og basert på tilbakemeldinger ble det gjort endringer på designet. Wireframene ble ikke oppdatert etter at mockupen var laget og viser derfor fremdeles til første utkast av designet. Figur 13 viser hovedsiden i Mockupen, resten av mockupen er i appendiks 18.

Dette lot brukeren teste webapplikasjonen, komme med tilbakemeldinger, og tillatte gruppen å gjøre endringer basert på disse tidlig i prosessen (Benyon, 2019, ss. 195-198). Dette sparte gruppen mye tid senere i prosjektet.

Mockupen ble gjort om til prototypen ved å legge til funksjonalitet der det var hensiktsmessig for å simulere hvordan nettsiden kommer til å fungere. Figur 14 viser Dashboard fra prototypen i sprint 3, og etterpå ble andre design endringer gjort i applikasjonen. Resten av prototypen ligger i appendiks 19. Store endringer som ble gjort til designet etterpå var at Dashboardet gikk fra å bruke knapper til faner vist i figur 31.

2.2.6 Brukervennlighet

I dette underkapittelet forklarer gruppen hvordan de utførte frontend-testing. Gruppen brukte designprinsipper og gjennomførte flere brukertester gjennom utviklingen av systemet. Dette ble gjort for å forbedre brukeropplevelsen i applikasjonen.

Brukertesting

I designprosessen ble det gjennomført seks brukertester av prototypen med representanter fra Capgemini. Gjennomgangen av brukertestene er vist i appendiks 3. Brukertestene gav gruppen verdifull tilbakemelding som ble brukt for å forbedre systemets design, med kontinuerlige endringer etter hver test. Siden det ble gjennomført seks brukertester, ble det gjennomført seks forbedringsiterasjoner gjennom designprosessen. Representanter fra Capgemini delte også ressurser med gruppen for å forbedre brukeropplevelsen i applikasjonen.

Designprinsipper

Selv om gruppen har hatt flere brukertester med blant annet ansatte hos Capgemini og demoer med representanter fra RG, var det tydelig at det ikke var mulig å holde nok brukertester med de som faktisk skulle bruke systemet. Derfor har gruppen, gjennom hele prosessen fulgt Benyons designprinsipper, for å sikre at brukergrensesnittet er av god kvalitet, er enkel å forstå og bruke.

2.3 Systemarkitektur

Ifølge Robert C. Martin er målet med systemarkitektur å minke ressursene som er nødvendige til å utvikle og vedlikeholde et system (Martin, 2017, s. 27-28). Systemarkitekturen skaper derfor grunnmuren for utviklingen av hele systemet, og gir en oversikt over ulike teknologier og krav i systemet (Martin, 2017, s. 141-150).

Dette kapitlet vil presentere systemets arkitektur, de forskjellige hensynene som ble tatt når det ble utviklet ved hjelp av objektorientert analyse og design, og de forskjellige verktøyene som inngår i systemarkitekturen.

2.3.1 Prioritering av Arkitekturdesignkriterier

Før gruppen kunne definere en systemarkitektur, valgte de å analysere behovene til systemet i forhold til objektorientert analyse og design. Dette gav en helhetlig forståelse av systemets funksjonalitet og bidro til å definere klare grensesnitt mellom ulike deler av applikasjonen. Ved å følge prinsippene for objektorientert analyse har gruppen kunne bygge en løsning som er fleksibel og skalerbar, med mulighet for enkel utvidelse og tilpasning i fremtiden (Mathiassen et. al., 2018, s.179-190). Dette ble gjort ved hensyn til Capgemini, som skal overta applikasjonen ved prosjektets slutt.

Ved å utforme en klar og strukturert arkitektur, vil det være enklere for Capgemini å forstå systemet og fortsette videreutviklingen.

OOA&D Kriterier	Veldig viktig	Viktig	Mindre viktig	Irrelevant	Lett å implementere
Usable		x			
Secure	x				
Efficient				x	
Correct	x				
Reliable	x				
Maintainable			x		
Testable		x			
Flexible			x		
Comprehensible	x				
Reusable		x			
Portable		x			x
Interoperable	x				

Figur 15: Tabell for prioritering av designkriterier

Gruppens prioriteringer er illustrert i figur 15 ovenfor. Videre skal gruppen forklare hva de forskjellige prioriteringene går ut på, og hvorfor gruppen har valgt å prioritere de slik.

Veldig viktig

Secure går ut på å sikre at uautoriserte brukere ikke får tilgang til sensitiv data og fasiliteter (Mathiassen et. al., 2018, s.180). Siden systemet inneholder sensitiv kundedata, er det viktig at informasjonen ikke blir kompromittert. Derfor, er det viktig å ta i bruk sikkerhetstiltak som beskytter mot «SQL injection», «cross site scripting» og eventuelt «cross site request forgery».

Correct handler om å oppfylle krav fra oppdragsgiver (Mathiassen et. al., 2018, s.180). Gruppen har prioritert dette som veldig viktig ettersom slike krav danner grunnmuren for prosjektet.

Reliable betyr at systemet må være pålitelig med utførelsen av funksjoner (Mathiassen et. al., 2018, s.180). Reliable ble sett som veldig viktig, da det er viktig at informasjonen systemet viser frem er pålitelig, og at funksjoner gjør det de skal.

Comprehensible tilsier om et system er intuitivt og logisk, og at nye utviklere og brukere kan lære seg systemet (Mathiassen et. al., 2018, s.180). Prosjektets formål er å vise frem idéer som utviklerne til sluttproduktet kan bruke videre. Det er derfor viktig å lage et logisk system som kan gjenbrukes. Klar kode som har god struktur og dokumentasjon er viktig for å sikre at systemet er lett å vedlikeholde, i tillegg til god brukervennlighet.

Interoperable går ut på at systemet kan enkelt kobles til andre systemer (Mathiassen et. al., 2018, s.180). Systemets hovedfunksjon er å hente ut og visualisere data fra en database, og det er derfor

viktig at coupling i systemet er løst, slik at systemet kan kobles til hvilket som helst bilde av databasen, gitt at strukturen er likt.

Viktig

Usable handler om at systemet er tilpasset til den relevante arbeidsrelaterte og tekniske konteksten hos sluttbrukeren (Mathiassen et. al., 2018, s.180). Dette er viktig fordi hvis systemet ikke utfører relevante handlinger for RG vil det ikke gi verdi å bruke systemet.

Testable omhandler ressursene som er nødvendig for å sikre at applikasjonen utfører dens tildelte oppgaver (Mathiassen et. al., 2018, s.180). Det ble sett på som viktig fordi gruppen ønsker å sikre at systemet utfører oppgavene sine og har få bugs.

Reusable handler om at deler av systemet kan gjenbrukes til andre liknende systemer (Mathiassen et. al., 2018, s.180). Prinsippet er viktig dersom Capgemini ønsker å lage et eget system basert på ideene fra gruppens PoC. Derfor skal det være enkelt for dem å kunne gjenbruke deler av systemet.

Portable innebærer hva som skal til for at systemet skal flyttes til en annen plattform (Mathiassen et. al., 2018, s.180). Gruppen så dette prinsippet som viktig, og at det skulle være enkelt å implementere. Dette er fordi gruppen bestemte å ta i bruk Docker for å gjøre systemet kjørbart på flere miljøer.

Mindre viktig

Maintainable handler om hva som må til for å opprettholde og feilsøke systemet (Mathiassen et. al., 2018, s.180). Prinsippet er mindre viktig på grunn av at det ikke er noe hensikt i å vedlikeholde en PoC. Likevel er det ikke irrelevant, da det er lurt å tenke langsiktig hvis enkelte funksjoner i systemet kan bli tatt med videre i et annet prosjekt.

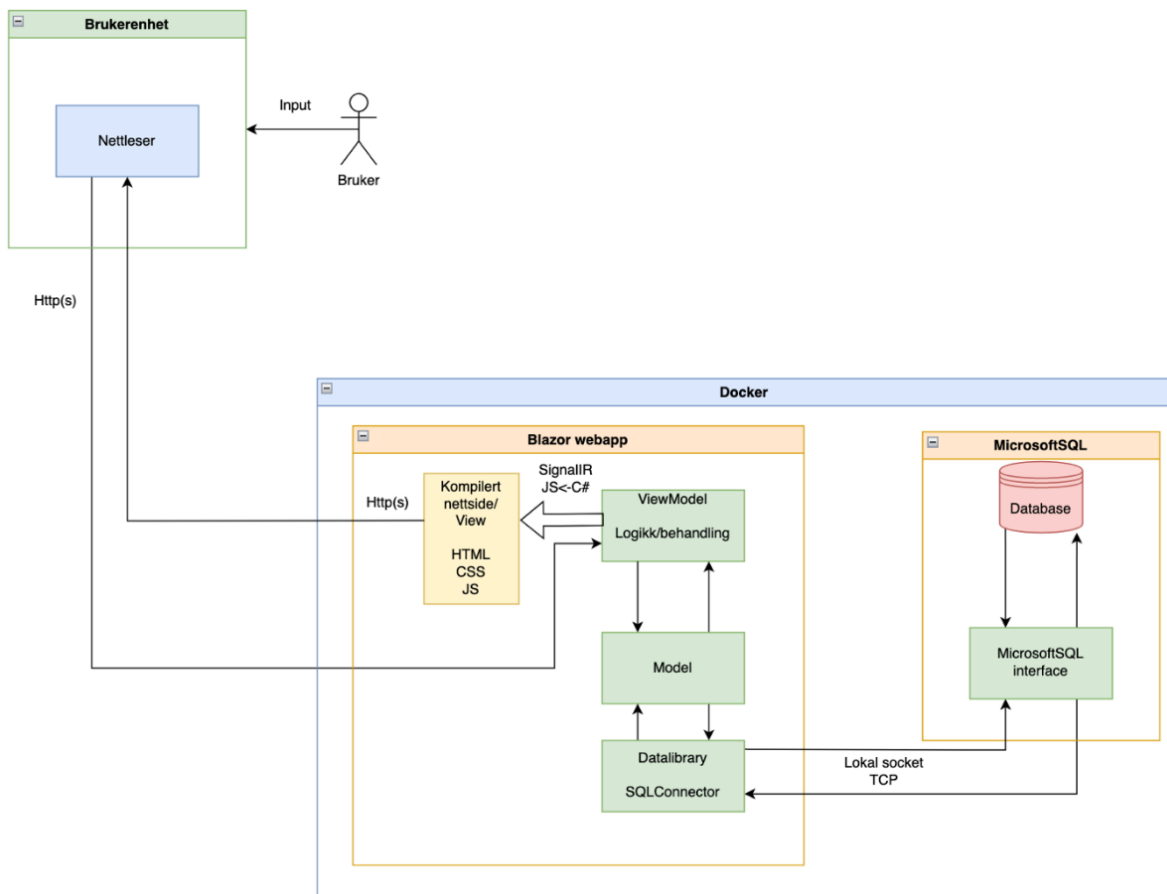
Flexible handler om kostnader for å endre eller modifisere på systemet som allerede er i produksjon (Mathiassen et. al., 2018, s.180). Dette ble nedprioritert, da systemet ikke skal ut i produksjon. Likevel, hvis deler av systemet skal bli brukt videre, bør systemet være utviklet på en forståelig måte og være enkelt å utvide eller endre.

Irrelevant

Efficient går ut på hvorvidt systemet bidrar til økonomisk gevinst (Mathiassen et. al., 2018, s.180). Systemet skal illustrere en plattform der RG kan enkelt hente informasjon for interne ansatte, og vil derfor ikke bidra direkte til økonomisk gevinst for bedriften. En kan argumentere for at det kan gi indirekte gevinst som for eksempel innsikt og data som gir dem et bedre grunnlag på fremtidige valg. Siden **Efficient** omhandler direkte gevinst som økonomi, så er dette dermed irrelevant.

2.3.2 Arkitekturmodell

Systemets overordnede arkitektur består av to separerte enheter for webapplikasjonen og databasen. Begge av disse ligger i virtualiserte «containere» som delvis isolerer dem fra hverandre. For å kommunisere mellom hverandre brukes det en lokal sokkel på maskinen, og for kommunikasjon brukes det en http eller https request, som etablerer en websocket for videre kommunikasjon. Gruppens arkitekturmodell kan en se under i figur 16.



Figur 16: Systemarkitektur

2.3.3 Docker

Gruppen valgte å ta i bruk Docker Desktop for å containerisere webapplikasjonen og databasen. Dette var gjort slik at applikasjonen skulle fungere og være rask til å sette opp uavhengig av maskinvare og programvare hos de som skulle ta over prosjektet (Docker Inc., u.d.).

Docker fungerer ved å isolere en applikasjon og dens avhengigheter inn i en «container», som er en virtuell maskin, og får tildelt systemressurser av vertsmaskinen gjennom Docker når den trenger de ved bruk av virtualisering. Virtualisering handler om å dele opp systemets maskinvare-ressurser inn til flere virtuelle enheter. I motsetning til vanlig virtualisering, deler Docker opp operativsystemet, og

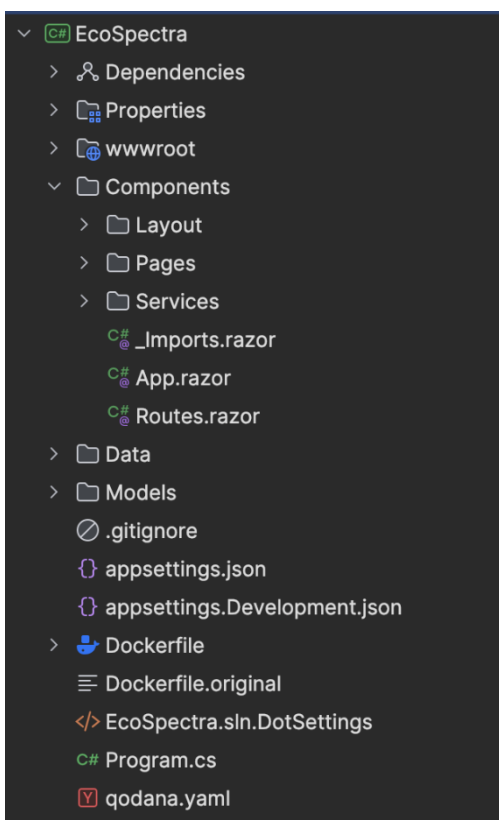
dermed gjenbruger mye av vertsmaskinens eksisterende prosesser. Containere tar derfor opp betydelig mindre plass, og kan kjøres på forskjellige plattformer uten noe krevende konfigurasjon (Docker Inc., u.d.).

Når det gjelder databasen, så ble den installert fra en basis MicrosoftSQL Server «image», som er en oppskrift på en Docker container. Webapplikasjonens container ble dannet under kompilering av systemet ved hjelp av en «DOCKERFILE», som også er en oppskrift på en container.

2.3.4 Blazor

Blazor er et rammeverk utviklet av Microsoft som gjør det mulig å kombinere backend og frontend i samme prosjekt, og skrive alt i programmeringsspråket C# fremfor JavaScript (Microsoft, u.d.). Dette er mulig på grunn av SignalR biblioteket, som er integrert i Blazor. SignalR etablerer en varig forbindelse med klienten ved å bruke en websocket. Denne forbindelsen gjør det mulig å skape en mer interaktiv opplevelse for klienten, ved at serveren kan kontinuerlig oppdatere siden uten noen input fra klienten (Microsoft, u.d.). På grunn av dette, kan applikasjonen generere JavaScript funksjoner som kaller til de opprinnelige funksjonene i serveren. Funksjonskall prosesseres da på serveren, og det oppdaterte sideinnholdet sendes til klienten etter at funksjonen er ferdig å kjøre (Microsoft, u.d.).

2.3.5 Mappedstruktur



Figur 17: Mappedstruktur

For å opprettholde en solid filstruktur, har gruppen valgt å følge standardoppsettet fra Microsofts Blazor WebApp Template. I denne malen er alle statiske elementer som bilder og CSS-filer plassert i **wwwroot**-mappen. Deretter ligger både tjenester og alle Blazor-komponenter i mappen **Components**. Disse er videre delt inn i tre undermapper. Utenfor disse finnes ulike statiske Razor-filer som blant annet legger til gjentakende statisk innhold og spesifiserer hvordan applikasjonen styrer klienten gjennom forskjellige sider.

Mappen **Layout** inneholder gjentakende dynamisk innhold, for eksempel en navigasjonsbar. **Pages** inneholder de forskjellige sidene av applikasjonen, mens **Services** inneholder tjenestene. I **Data**-mappen finnes databiblioteket, i dette tilfellet Entity Framework, og i **Models**-mappen ligger modellene for systemet.

En visuell fremstilling av mappestrukturen kan en se i figur 17.

For å strukturere de forskjellige visningene, har gruppen valgt å ta i bruk et design oppsett som heter Model-View-Viewmodel (MVVM). MVVM handler om å dele opp programlogikk og grensesnitt for å gjøre systemet enklere å utvikle, oppdatere og lese. Dette gjorde gruppen ved å skrive programlogikk i en «@code»-blokk helt nederst i hver View-fil illustrert nedenfor i figur 18 (Microsoft, u.d.).

```
84     else
85     {
86         <LoadingInfoBox></LoadingInfoBox>
87     }
88
89     }
90
91     @code
92     {
93         //variables
94         private CO2MachineForProj? clickedItem;
95         private string? customerNumber;
96         private string? ProjName;
97         private string? ProjRef;
98         private double TotalEmissions;
99         private int machines;
100        private CO2EmissionVisualized? Dataset;
101        private string? error;
102
103        //methods
104        protected override async Task OnInitializedAsync()
105        {
106            try
107            {
108                Dataset = await Co2EmissionVisualized.GetMachinesCO2ForProjAsync();
109            }
110            catch (Exception e)
111            {
```

Figur 18 - Eksempel på en "@code" blokk

2. Teknologi og Verktøy

Kapittel tre diskuterer en mer detaljert gjennomgang av teknologivalg, rammeverk og verktøy som er benyttet i prosjektet. Dette inkluderer en gjennomgang av Microsoft sitt DevOps-miljø, bruk av programmeringsspråk, bibliotek og ikke minst metoder for testing.

3.1. Administreringsverktøy

Her viser gruppen til verktøy som ble brukt for å administrere og overvåke prosjektet, samt de kommunikasjonskanalene som ble tatt i bruk under prosjektets utvikling. Azure DevOps fra Microsoft ble anvendt for å sentralisere både prosjektstyring og lagre kode. Denne plattformen gav gruppen et bredt spekter av integrerte tjenester og verktøy for å administrere et utviklingsprosjekt (Microsoft, 2024). Gruppen tok i bruk stort sett Repos, Boards og programtillegget Planning Poker.

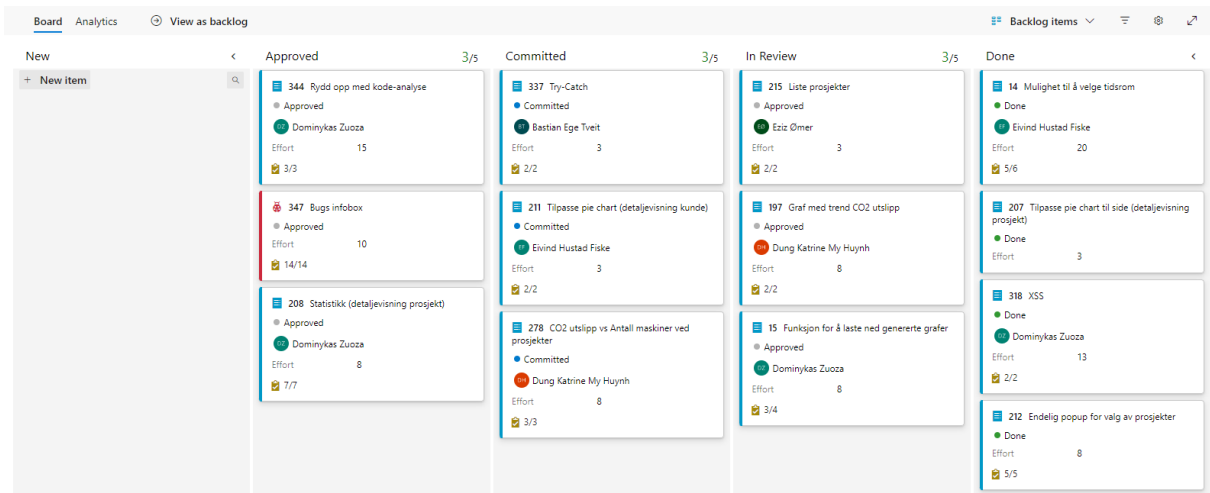
For å opprettholde åpen kommunikasjon innad prosjektgruppen, interessenter og veiledere, var det nødvendig å etablere gode kommunikasjonskanaler. Dialog med eksterne parter som Capgemini og RG ble hovedsakelig håndtert gjennom Microsoft Teams og e-post, ettersom disse plattformene allerede var aktivt brukt i deres arbeidstid. Discord ble brukt for interne dialoger med gruppen på grunn av alle medlemmenes tidligere erfaringer med denne plattformen. I tillegg var veileder fra Universitet i Agder (UiA) aktiv på Discord og e-post, noe som bidro til en enkel og fleksibel kommunikasjon.

For å forberede seg på mer alvorlige situasjoner, som uventet fravær eller eventuelle manglende oppmøte fra interessenter, ble det opprettet en detaljert kontaktliste. Dette ble en liste som inneholdt navn, telefonnumre og e-postadresser til alle involverte parter i prosjektet. Dette gav en forsikring om at alle kunne nås raskt under kritiske omstendigheter.

Disse verktøyene bidro til å skape god kommunikasjon mellom de forskjellige partene i prosjektet, samt gi oversikt over hvordan gruppen kan nå de forskjellige interessentene.

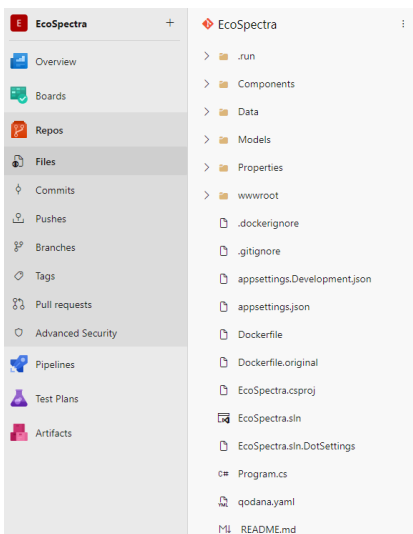
3.1.1. Azure DevOps-miljø

Det ble tidligere nevnt om bruk av Microsoft sitt Azure DevOps-miljø for å administrere prosjektet, hvor gruppen tok i bruk Boards, Repos og Planning Poker. Boards ble brukt for å opprettholde en agil tilnærming og høy kvalitet gjennom prosjektet. Den tilbyr verktøy for å planlegge, spore framgang, og diskutere arbeid i deres livssykluser (Microsoft, 2024). Her ligger de typiske Scrum-elementene som tidligere ble beskrevet og diskutert i kapittel 2.1.



Figur 19: Sprint Board

Figur 19 illustrer et eksempel på gruppens Sprint Board. Her ble PBI-er lagt inn fra backloggen, som ble formulert basert på brukerhistorier, tilbakemeldinger og ønsker fra interessentene. Disse ble vurdert med effort gjennom bruk av planning poker, noe som utdypes i kapittel 4.2. Hver PBI ble vanligvis tildelt en utvikler, og hvis en oppgave antas å bli for stor, ble den delt opp i mindre deloppgaver for å fremme iterativ framgang.



Figur 20: Azure Repository

Repos fungerer som et lagringssted for kodebasen til prosjektet. Figur 20 til venstre illustrer hvordan gruppens repository for prosjektets kodebase ser ut. Dette er utført på Microsoft sine skytjenester og er en del av Azure DevOps (Microsoft, 2024). Dette betyr at kodebasen ikke er bundet til en lokal fysisk enhet, og tilbyr større mulighet til samarbeid på samme kodebase. Hver utvikler har en lokal kopi av kodebasen på sin egen maskin, og er fri til å utføre endringer relatert til nye features, PBI-er eller bugs før endringene sendes opp til den skybaserte kodebasen.

Git-versjonskontroll

Git-versjonskontroll sporer og administrerer endringer i kildekoden, slik at et utviklerteam effektivt kan håndtere og samarbeide om endringer i systemet (Atlassian, u.d.). Ved å opprettholde en langsiktig historikk over endringer for hver fil, gir Git muligheten til å gjenopprette tidligere versjoner av kode hvis uventede problemer skulle oppstå (Atlassian, u.d.). Git muliggjør også forgrening og sammenfletting av kodebasen. I tillegg til å kunne samarbeide sømløst mellom gruppemedlemmer for hver funksjon som skal implementeres (Atlassian, u.d.)

3.2 Backend og Frontend Teknologier

Dette kapitlet omhandler teknologien brukt for utviklingen av backend og frontend i applikasjonen. Frontend viser til hva brukerne kan se og tillater brukerne å samhandle med applikasjonen, mens backend styrer applikasjonens funksjonalitet (Amazon, n.d.). Gruppen redegjør de ulike programmeringsspråkene, rammeverkene og bibliotekene som ble brukt for utviklingen av systemet. Til slutt, vil gruppen vise til teknologier og verktøy som ble tatt i bruk for databasen og sikring av kodekvalitet.

3.2.1 Programmeringsspråk og Rammeverk

C# et objektorientert programmeringsspråk utviklet av Microsoft og brukes i .NET rammeverket (Microsoft, n.d.). Gruppen tok i bruk programmeringsspråket ettersom at alle har tidligere erfaring med C#.

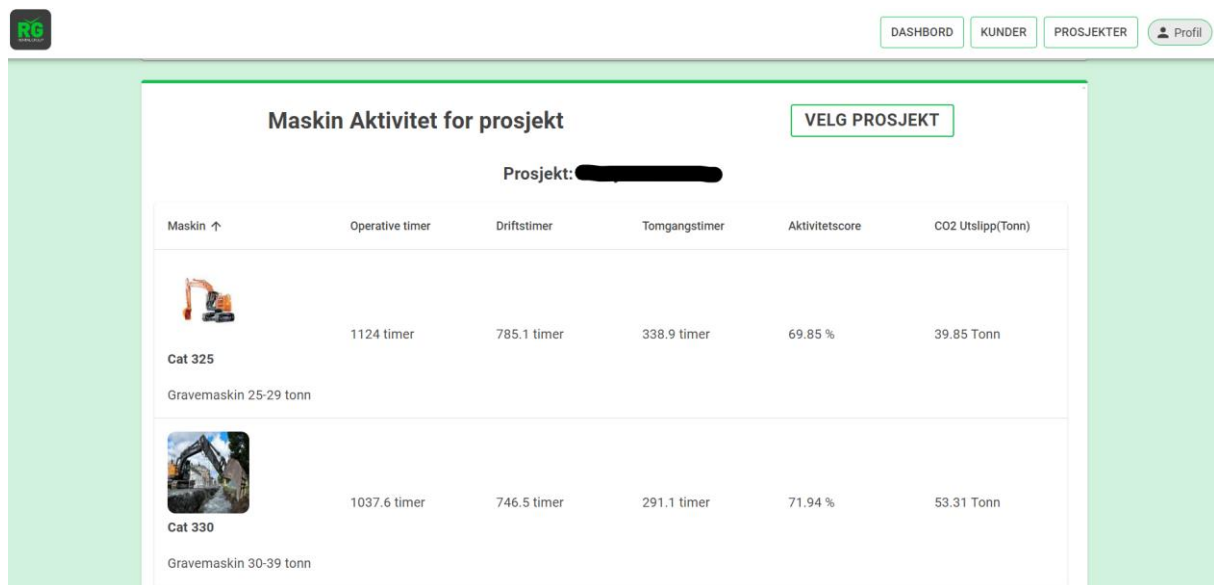
.NET er et programmeringsrammeverk utviklet av Microsoft, og tilbyr flere verktøy og ressurser for å kompilere applikasjoner (Microsoft, n.d.). ASP.NET er en underrammeverk til .NET og har en modulær struktur som tilbyr fleksibilitet og robuste funksjoner for moderne webapplikasjoner (Microsoft, n.d.). Kombinasjonen av fleksibilitet, sikkerhet og ytelse fra .NET og ASP.NET tilbyr et sterkt rammeverk for å utvikle en webapplikasjon.

I tillegg brukte gruppen Blazor for å utvikle brukergrensesnittet for applikasjonen. Som nevnt i kapittel 2.3.4 ga dette gruppen muligheten til å utvikle frontend ved bruk av C#, HTML og CSS (Microsoft, u.d.). Gruppen valgte dette med tanke på kjennskap i C# fra tidligere prosjekter, samt for å unngå unødvendig tidsbruk på å undersøke nye rammeverk som React eller Next.js. Blazor gav muligheten til å kunne skrive både tjenester og frontend på samme programmeringsspråk (Syncfusion, n.d.).

3.2.2 Bibliotek

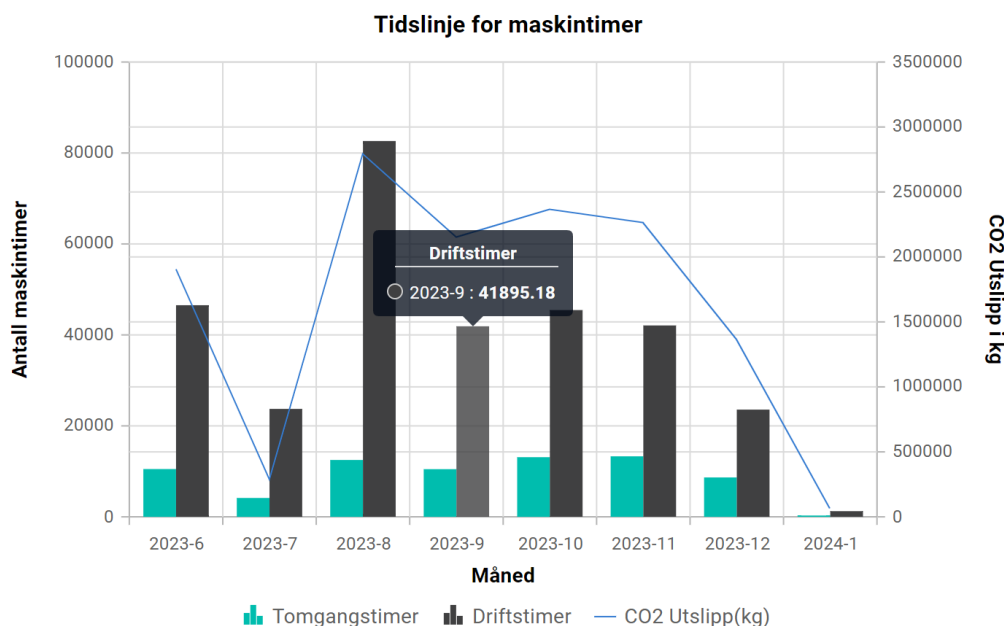
Systemet tar i bruk flere forskjellige biblioteker for utviklingen av funksjonalitet og brukergrensesnitt. Kodebibliotek tilbyr en omfattende løsning for organisering og bruk av kodesnutter slik at man slipper å skrive kode fra bunnen av (Altinn, n.d.).

Et av bibliotekene gruppen valgte å benytte i dette prosjektet var MudBlazor. Dette biblioteket bidro med ferdigutviklede komponenter som kunne gjenbrukes i systemet. Gruppens tekniske rådgiver fra Capgemini fremhevet at MudBlazor kunne være et spennende bibliotek å undersøke, og derfor valgte gruppen å ta det i bruk. Gruppen brukte MudBlazor for flere elementer i applikasjonen, dette inkluderer informasjonsbokser, knapper, tabeller og bilder. Et eksempel på dette er vist i figur 21. I tillegg gjorde MudBlazor det lettere å implementere blant annet sorterings- og filtreringsfunksjoner i tabellene.



Figur 21: Infoboks for Maskin Aktivitet - tabell for maskiner for prosjekt

Gruppen trengte et annet bibliotek for å utvikle grafene i applikasjonen, noe som biblioteket Syncfusion var godt egnet for. Dette gav gruppen et variert utvalg av diagramtyper som for eksempel linje-, kolonne- og sirkeldiagram. I tillegg til dette implementerte gruppen en scatterplot for visning av antall maskiner og prosjekter. Syncfusion har også muligheten til å blande diagramtyper, derfor kunne gruppen ha et linje- og kolonnediagram i samme graf, vist nedenfor i figur 22.



Figur 22: Miks-graf som viser tidslinje for maskintimer (driftstimer og tomgangstimer)

Dette gav gruppen muligheten til å vise frem komplekse diagrammer som kunne vise sammenhengen mellom faktorer som tomgangstimer, driftstimer og CO2-utslipp.

I tillegg tok gruppen i bruk noen eksterne biblioteker for å implementere ekstra funksjonalitet. En av disse var HTMLAgilityPack biblioteket. Dette ble brukt for å hente ut bilder fra en google søk, slik at gruppen ikke trengte å lagre alle bildene av de forskjellige bedrift-logoene og maskinene lokalt. HTMLAgilityPack gjør det mulig å sende forespørsel til eksterne nettsider (i dette tilfellet google bilder), og hente inn forskjellige elementer, for eksempel bilder eller tekst (Html Agility Pack, u.d.).

3.2.3. Database

Gruppen mottok en datadump fra Capgemini, som ble kjørt lokalt på en Docker container. I dette underkapittelet presenteres teknologiene brukt for datainnhenting fra databasen.

Gruppen brukte Microsoft SQL Server (MSSQL) som et databaseadministrasjonssystem (Tutorialspoint, n.d.). MSSQL kan enkelt sammenkobles med andre Microsoft-teknologier som ASP.NET, noe som gav gruppen et passende sett med verktøy for å utvikle systemet (Microsoft, n.d.).

Entity Framework (EF) er en moderne objektrelasjonskartlegger (ORM) som tillater utviklere å hente data fra databaser (Microsoft, n.d.). Ved å først etablere en «ConnectionString», som er adressen til databasen, kan EF koble seg til databasen og å hente ut data. EF knytter deretter resultatene fra spørringene til klasser som representerer de forskjellige tabellene i databasen.

Selv om gruppen hadde begrenset erfaring med EF, valgte de å ta det i bruk, siden den klarer å selv kartlegge strukturen i databasen, noe som er spesielt nyttig, ettersom gruppen ikke var kjent med

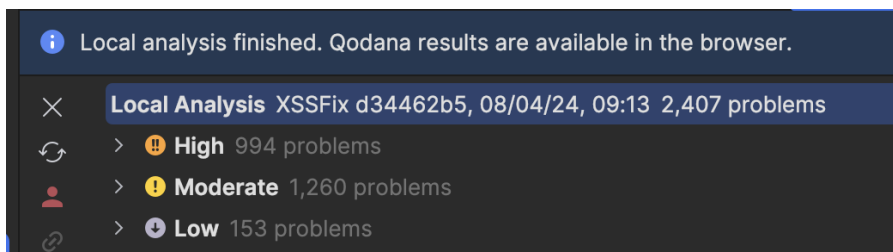
strukturen fra før. I tillegg hadde flere representanter fra Capgemini erfaring med MSSQL og hadde brukt EF i flere prosjekter. Fordelen med å bruke EF er at den effektiviserer dataauthenting. Gruppen valgte derfor å benytte EF på grunn av databasens omfattende tabeller som data skulle hentes fra.

En av de nye tingene gruppen lærte fra en teknisk rådgiver i Capgemini var å ta i bruk Common Table Expression (CTE) spørringer. Disse spørringene var bedre egnet til henting av komplekse datasett, da en kunne sette de opp stegvis og filtrere bort unødvendig data underveis (Tutorialspoint, n.d.). Gruppens bruk av CTE spørringer økte effektiviteten for datainnhenting fra databasen.

3.2.4. Verktøy for kodekvalitet

For å sikre kvaliteten til koden i systemet, har gruppen tatt i bruk analyseverktøyet JetBrains Qodana. Dette verktøyet bidrar med automatiske kodeanalyser som avdekker svakheter i koden før det er godkjent til sammenfletting med main grenen.

Qodana er et analyseverktøy for koden og bidrar til at utviklingsprosessen følger kodestandarder og sørger for at koden er forståelig, vedlikeholdbar og sikker (JetBrains, n.d.). Qodana gir raske tilbakemeldinger på kode som kan forbedres, illustrert nedenfor i figur 23, og gir anbefalinger på hvordan det kan forbedres. Analyseverktøyet sparer tid og initiativ krevd i code reviews (JetBrains, n.d.).



Figur 23: Visning av antall problemer i koden med Qodana før code cleanup

3. Prosjektgjennomføring

Dette kapitlet gir en gjennomgang av hvordan gruppen gjennomførte prosjektet, med fokus på de administrative aspektene av de ulike Scrum-aktivitetene, delt opp i sprintene som vist i kapittel 2.1. Her skal det forklares hvordan prosjektet ble gjennomført i form av product backlog, sprint backlog og planning, Daily Scrum, sprint reviews- og retrospect. Videre går kapitlet i dybden på gjennomføringen av enhver sprint.

4.1. Scrum i en Hybrid Waterfall-Agil metodikk

Scrum er et agilt rammeverk for gjennomføring av prosjekter som bidrar til struktur og fleksibilitet. Metodikken blir ofte brukt innenfor IT-bransjen og omfatter en rekke artefakter og begreper, hvor hoved-ideen er å dele opp store oppgaver i mindre deler og jobbe mer iterativt (Atlassian, u.d.). En kan argumentere for at dette fremmer effektivitet og frie rammer.

En må også skjenne Waterfall-metodikken for å se den helhetlige Hybrid metoden. Waterfall er en tradisjonell prosjektstyringsmetode som strukturerer prosjektene i fem faser: kravspesifikasjon, design, implementering, verifisering og vedlikehold. Slik sekvensiell gjennomføring gir en klar oversikt over prosjektets framgang og milepæler på hva som skal bli utført. Ulempen her er mangelen på fleksibilitet. For eksempel, hvis en fase ikke blir gjennomført slik den ble planlagt, kan dette medføre til forsinkelser eller stanse arbeidet (Atlassian, n.d.).

Dermed besluttet gruppen seg til en Hybrid Waterfall-Agil metodikk, som tar til seg alle Scrum-elementer, samtidig som den setter opp prosjektet i en Waterfall på starten av prosjektet (Zaleski, u.d.).

Før en diskuterer Scrum i en hybrid metodikk, må man få bakgrunnsinformasjon på hoved artefaktene:

- **Product Backlog** – er en samlet liste over features, forbedringer, bug fixes, tasks eller arbeid som må bli fullført for å utvikle produktet (Harris, u.d.). Disse oppgavene kalles Product Backlog Items (PBI), som gruppen henter fra, for å bygge opp den pågående sprinten i en Sprint Backlog.
- **Product Increment** – refererer til en konkret og brukbar versjon av produktet som produseres ved slutten av hver iterasjon (Harris, u.d.). Å bygge et produkt inkrementelt innebærer at man arbeider gradvis, hvor hver fase bygger på og tilfører nye elementer til produktet basert på tidligere utvikling.

- **Burndown Chart** – er ikke sett på som en offisiell Scrum artefakt, men som blir brukt for å visualisere og spore fremgang samt gjenstående arbeid (Harris, u.d.).

I tillegg har gruppen inngått i rollefordelinger i henhold til rammeverket Scrum, som har definert tre roller i et team:

- **Produkteier** – fungerer som bindeleddet mellom Scrum-teamet og kunden. Personen i denne rollen må kunne forstå kundens behov og effektivt formidle disse til teamet (West, u.d.). Vegard Sætre fra Capgemini har fått tildelt denne rollen og vil delta i sprint reviews for å gi tilbakemeldinger og få oppdatering på prosjektstatus.
- **Scrum Master** – er nøkkelpersonen som sikrer sammenheng og struktur i prosjektet. Dominykas Zuoza fra gruppen har denne rollen, mens Eziz Ømer fungerer som vara Scrum Master ved Dominykas sitt fravær. Rollen innebærer å sørge for at rammeverket følges korrekt og at teamet arbeider i tråd med de forhåndsavtalte planene (West, u.d.).
- **Utviklere** – er ansvarlige for å demonstrere fremdriften i hver sprint. De skal være selvdrevne, kunne organisere seg og ta viktige avgjørelser selvstendig. I dette prosjektet betraktes alle som fullstack-utviklere, selv om noen kan ha større fokus på enten back- eller frontend.

Gjennom arbeidet har gruppen i stor grad hatt et flatt hierarki, hvor alle gruppe-medlemmer hadde de samme ansvarsene. Likevel hadde gruppen noen roller for å sørge for at arbeidet var strukturert og oversiktlig. Gruppen begynte med å velge en gruppeleder og en vara. Disse ble likt med Scrum master rollen i Scrum rammeverket nevnt over. I tillegg ble en sekretær utnevnt, Dung Katrine My Huynh, som hadde ansvaret for å sende all e-post og drive all kommunikasjon med både bedriften og UiA, samt dokumentere arbeidstimer i begynnelsen av prosjektet. For å se antall arbeidstimer gjennomført av hvert gruppe-medlem i løpet av sprintene, se figur 28 i kapittel 4.2.4.

Gruppen valgte å ikke rullere rollene, ettersom at de som fikk rollene hadde tidligere erfaringer med dette. I tillegg var det enklere hvis én person skulle holde oversikt over dokumentasjon, for eksempel Daily Scrum og backlog.

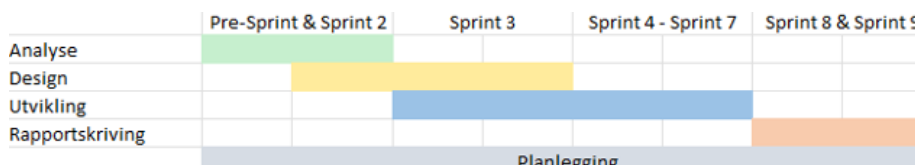
Gruppen har gjennomført følgende *Scrum*-aktiviteter:

- **Daily Standup** – er korte, daglige møter som ikke varer mer enn 10 minutter, hvor deltakerne diskuterer fremdriften og eventuelle utfordringer som har oppstått (Rehkoph, u.d.). Møtet fokuserer på tre faste spørsmål:
 - Hva har jeg jobbet med?
 - Hva skal jeg jobbe videre med?

- Har noen problemer oppstått?
- **Sprint planning** – er planleggingsfasen før starten av hver sprint, med mål om å forbedre den kommende sprinten, sammenlignet med den forrige. Her velges oppgaver fra produktets backlog eller så legges det til nye oppgaver som skal fullføres (West, u.d.).
- **Sprint review** – er en sentral aktivitet ved slutten av hver sprint, hvor utviklingsteamet og interessenter møtes for å gjennomgå det utførte arbeidet i sprinten (Radigan, u.d.). I dette prosjektet deltar produkteieren regelmessig for å få en oppdatering. Selv om kunden er en stor interessent i prosjektet, så har de minimalt med tid og er derfor ikke med på sprint reviews. Framgangen presenteres for produkteieren, som deretter gir tilbakemeldinger og peker på potensielle forbedringsområder.
- **Sprint retrospective** – er en aktivitet hvor gruppen reflekterer over hva som har gått bra, hva som gikk mindre bra og hva en tenker å ta med videre fra forrige sprint. Denne aktiviteten finner sted rett etter sprint reviews, og her la gruppen frem sine utfordringer og identifiserer forbedringsmuligheter for den neste sprinten (Atlassian, u.d.).

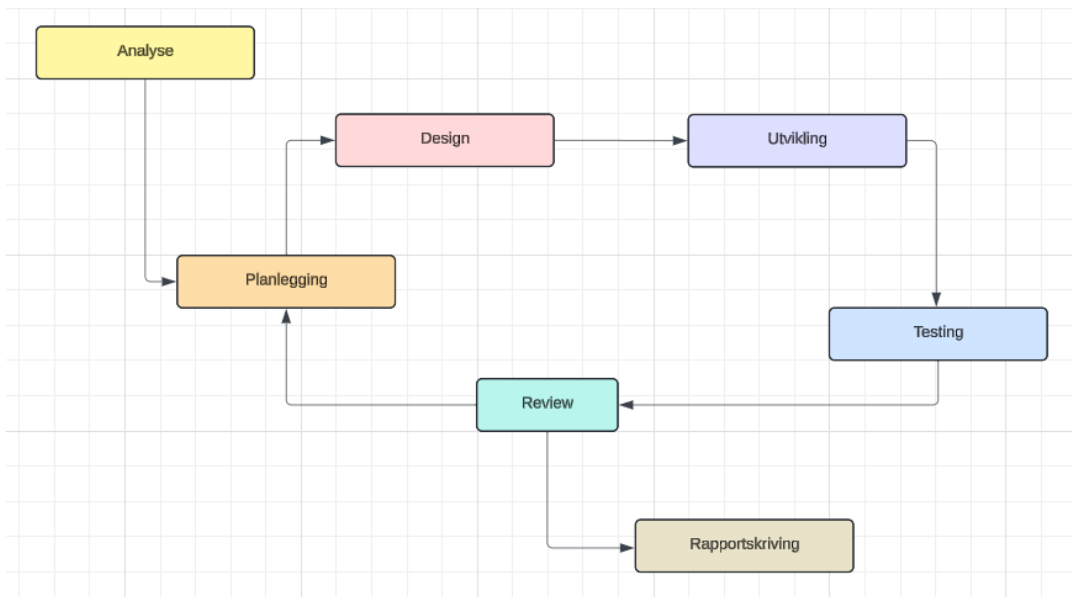
4.1.1. Scrum og Waterfall misforståelser

Det var en del forvirringer i begynnelsen av prosjektet angående hvilken metodikk som ble anvendt. Veiledere fra UiA antok i begynnelsen at gruppen brukte Waterfall-metodikk på grunn av hvordan sprintene var strukturert. Dette var delvis korrekt, da sprintene ble ubevisst planlagt på en måte som lignet Waterfalls sekvensielle tilnærming. Figur 24 viser den opprinnelige planen av sprintene.



Figur 24: Opprinnelig plan av sprintoversikt

Til tross for den sekvensielle strukturen som lignet waterfall, implementerte gruppen elementer av Scrum gjennom hele prosjektets livsløp. Pre-Sprint hadde fokus på kravanalyse og planlegging, som er startfasen i en Waterfall-struktur. Fra Sprint 2, som i all hovedsak skulle fokusere på design, begynte gruppen allerede å integrere en mer fleksibel og iterativ tilnærming, ved å arbeide med analyse, design og utvikling parallelt i sprinten. Under sprintene ble det gjennomført manuelle tester, og gjort code reviews før det ble «pushet» opp til main grenen. Den iterative prosjektstrukturen blir visualisert nedenfor i figur 25.



Figur 25: Gruppens Waterfall-Agile prosjektstruktur

Selv om hver sprint hadde et hovedfokus, opprettholdt gruppen en kontinuerlig iterativ tilnærming som er typisk for Scrum, hvor analyse og design ble revidert og forbedret gjennom hver sprint. Dette var basert på tilbakemeldinger som ble gitt av interessenter gjennom blant annet sprint reviews og møter med kunde. En slik tilnærming var i tråd med Scrum-prinsippet om kontinuerlig forbedring, selv om den overordnede planen kunne virke sekvensiell.

Ved å kombinere den strukturerte planleggingen av Waterfall og en fleksibel modell med Scrum sin smidighet og tilpasningsevne, landet gruppen til slutt på en hybrid Waterfall-Agil tilnærming (også kjent som WAgile) (Zaleski, u.d.).

4.2 Tidsestimering

Tidsestimering er det en gjør i planleggingsfasen av en sprint der man knytter en tidsverdi til en PBI. Dette er gjort slik at gruppen kan planlegge hva de skal få gjort innen en tidsperiode. Ved å ta en grundig tidsestimering av alle oppgaver, kan en sikre at gruppen har nok å gjøre, men ikke for mye. Samtidig bidro dette til å avdekke eventuelle utfordringer på enkelte oppgaver i en tidlig fase (Radigan, u.d.).

Gruppen har valgt å tidsestimere alle PBI-er som tilhører Design og Utviklingsfasen, samt deler av analysefasen. Siden analysefasen var det første gruppen skulle gjennom, opplevde de utfordringer med å estimere oppgaver ettersom at de ikke var definert på forhånd og det var mye å sette seg inn i. I tillegg til dette, hadde gruppen lite erfaring med tidsestimering og skulle få opplæring fra Capgemini.

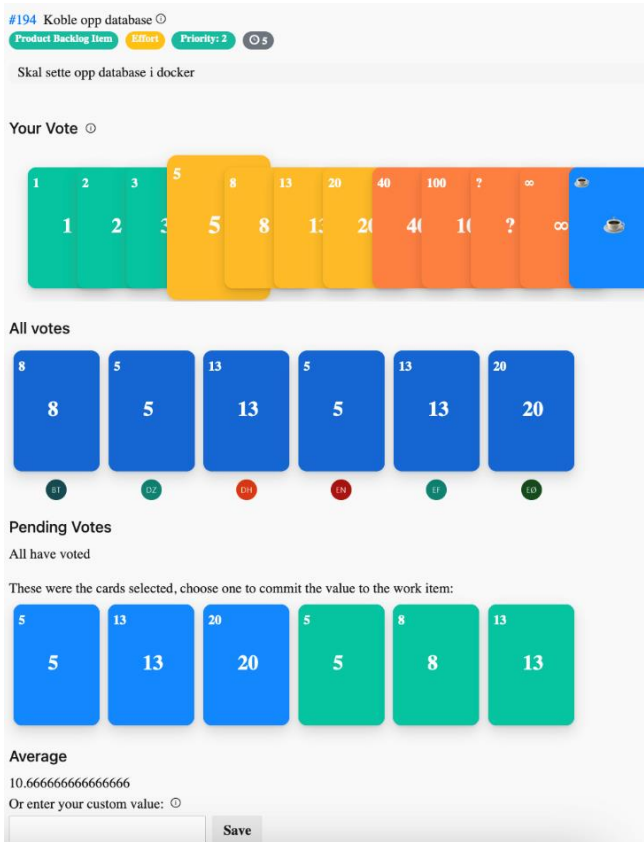
4.2.1 Tidsestimering med «Effort»

Gruppen bestemte seg tidlig, ved hjelp av en prosjektleder hos Capgemini, i å tidsestimere oppgaver med «Effort» framfor timer. Effort er et tall som representerer innsatsen som kreves for å fullføre en oppgave. Dette tallet er derfor en subjektiv enhet, og er noe gruppen selv setter en verdi på utfra erfaring. Dette gjør det lettere å ta i betraktning alle de forskjellige nyansene i en oppgave, som blant annet tid brukt til opplæring, usikkerhet, og varierende kompetanse. Ifølge prosjektlederen, er det vanlig at estimeringen ved bruk av denne metoden er litt unøyaktig i starten, men etter noen estimeringsøkter skal gruppemedlemmene tilpasse seg og kjenne bedre hva en effort tilsvarer.

Gruppen synes denne tilnærmingen fungerte mye bedre enn estimering i timer, da det ofte kan være utfordrende å tildele et konkret antall timer til en oppgave. I tillegg er effort et mer løst begrep, som i gruppens erfaring fører til at det blir mindre forskjell mellom tidsestimatene fra uerfarne gruppemedlemmer, og gruppemedlemmer som er godt kjent med oppgaven som skal estimeres.

4.2.2 Planning poker

Selve tidsestimeringen var gjort ved å bruke planning poker metoden, hvor alle medlemmer i estimeringsøkten stemte på hvor mye effort en oppgave skulle ta. Deretter diskuterte hvorfor estimeringen skulle være så høy/lav, og gjentok dette til alle var enige om hvor mye effort de



Figur 26 - Planning poker

forskjellige oppgavene måtte ha (Visual Paradigm, u.d.). Et eksempel på en slik planning poker sesjon kan en se i figur 26. Til dette brukte gruppen en innebygd plugin i deres utviklingsmiljø Azure DevOps.

I motsetning til estimering med effort, har gruppen hatt lang erfaring med å bruke planning poker til å estimere oppgaver. Å bytte over til effort i estimeringen førte som forutsett til noen unøyaktigheter i starten, men systemet ble langt mer nøyaktig etter noen runder.

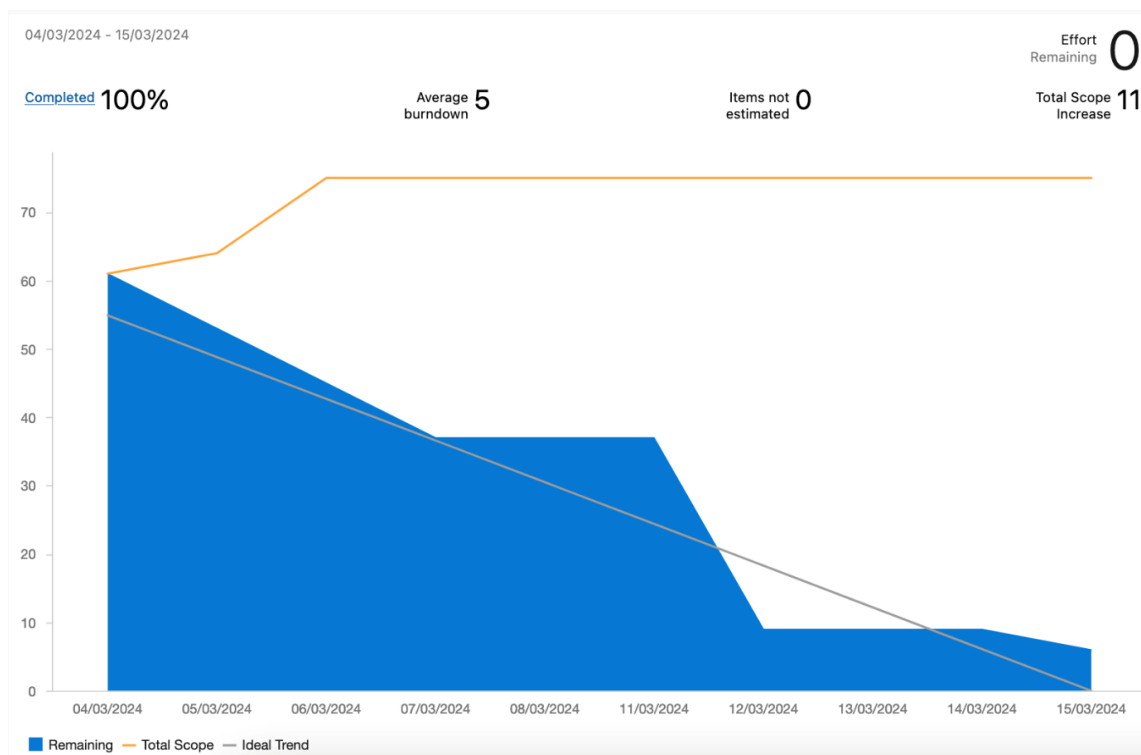
Noe nytt for gruppen var at de gikk over til en større skala i effort som kunne tildeles en oppgave. Før dette prosjektet, har gruppen vært innstilt på at alle estimeringer som var mer enn 10 skulle deles opp i mindre oppgaver. Men, etter et styringsmøte med prosjekteier fikk gruppen innsikt i hvordan tidsestimering fungerte hos Capgemini. Prosjekteieren sa at de gjorde det motsatte, og at hvis en oppgave tok kortere tid enn én dag, så var det ingen vits i å legge dette inn som en PBI.

Fordelen med denne tilnærmingen er at en slipper å dele opp og estimere flere mindre oppgaver, og dermed sparer tid som kunne blitt brukt til å løse dem.

En åpenbar ulempe med denne tilnærmingen er at det øker sjansen for at gruppen ikke blir ferdig med alle PBI-ene innen sprinten. Dette er fordi tiden det tar for å gjøre ferdig en PBI kan gå ut over flere dager, og dermed sprinter. En løsning på dette som gruppen har funnet, er å begrense omfanget av «Tasks» til maks en dags arbeid, og heller dele taskene til en PBI på flere personer. I de få tilfellene en PBI ikke blir helt ferdig til slutten av sprinten, har gruppen bare flyttet den over til neste sprint. Dette har fungert bra for gruppen, og med denne tilnærmingen klarte de å estimere sprintene på en nøyaktig måte, uten å bruke for mye tid på dette.

4.2.3 Burndown-chart

Gjennom sprintene, genererte gruppen flere «Burndown-charts» som illustrerte hvilken dag oppgaver ble gjennomført basert på deres effort, og hvor mye arbeid som gjensto. Ut ifra dette, kan en vurdere om gruppen har holdt et godt tempo gjennom hele sprinten, og om gruppen over- eller underestimerte oppgavene. Dette hjelper å tilpasse arbeidsomfanget til videre arbeid, både under sprinten ved å vise om gruppen er foran eller bak tidsskjemaet, og etter sprinten, ved å gi et overblikk over hvor mye effort gruppen klarte å håndtere på sprinten (Keup, 2022). Et eksempel på en burndown-chart fra sprint 5 kan en finne nedenfor i figur 27.



Figur 27: Burndown-chart fra Sprint 5

Ideelt skal den blå linjen følge den samme kurven som den grå, det vil si at gruppen har jobbet i likt tempo gjennom hele arbeidet, og at det er ingen arbeid som gjenstår fra denne sprinten. I figur 27, kan man se at gruppen har stort sett holdt et konstant tempo gjennom sprinten, og at det er veldig få oppgaver som gjenstår.

4.2.4 Dokumenterte timer

Gjennom prosjektgjennomføringen har gruppen hatt både planlagte felles gruppemøter og selvstendig arbeid. Selvstendig arbeid foregikk etter planlagte gruppemøter og ble tatt opp i Daily Standup dagen etter. Gruppemedlemmene var selv ansvarlige over å føre antall timer inn i timelista. Hvis manglende timer ble oppnådd, ble dette tatt opp med veileder fra universitetet. Oversikten over timeforbruket i prosjektet er vist nedenfor i figur 28. Det er verdt å nevne at enkelte uker med kun rapportskrivning er ikke inkludert i figuren.

Medlem	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Total	Oppmøte
Lengde av sprint	2 uker	2 uker	2 uker	2 uker	2 uker	1 uke	2 uker	2 uker	15 uker	-
Oppsatte gruppetimer	47,25	48,75	46,50	44,00	51,50	20,00	44,00	50,00	352,00	100%
Minimumskrav for gruppetimer	37,80	39,00	36,80	35,20	41,20	16,00	35,20	40,00	281,60	80%
Bastian	47,25	48,75	45,00	43,50	34,00	16,00	37,75	50,50	322,75	91,69%
Dominykas	47,25	48,75	46,50	49,75	51,75	17,50	42,50	49,50	353,50	100,42%
Katrine	47,25	48,75	46,50	53,00	54,50	23,00	54,50	55,50	383,00	108,09%
Eirik	47,25	48,75	52,50	44,00	52,50	22,00	46,00	49,50	362,50	102,98%
Eivind	37,00	38,00	23,50	42,00	43,00	16,00	43,00	47,00	290,50	82,53%
Eziz	47,25	48,75	46,50	37,00	45,25	18,00	40,00	44,00	326,75	92,83%

Figur 28: Timeliste per gruppemedlem for prosjektet

4.3. Gjennomføring av Scrum

Gruppen gjennomførte Scrum ved å opprettholde prinsippene product backlog, sprint planning og review, i tillegg til Weekly Scrum. Disse aktivitetene var med på å vedlikeholde og styre arbeidsflyten i løpet av prosjektet. Hele prosjektet ble dokumentert i form av en dagbok der den ble skrevet på en daglig basis av gruppens Scrum Master.

4.3.1. Product Backlog

I hver sprint planning-sesjon ble product backloggen utarbeidet av gruppen. Dette inkluderte å legge til features, PBI-er og å gjøre rede for bugs basert på brukerhistorier, samt tilbakemeldinger og ønsker fra interessenter. Product backloggen ble kontinuerlig oppdatert, og det var Scrum Masterens ansvar å legge til eller fjerne oppgaver etter behov. Dette sikret at backloggen alltid reflekterte de nyeste prioriteringene og endringer, i tillegg støttet det en smidig utviklingsprosess.

4.3.2. Sprint Planning og Sprint Backlog

Som nevnt i kapittel 4.1. foregår sprint planning første dag på starten av hver sprint, for å planlegge hvordan gruppen skal gå fram de neste to ukene. Både RG og produkteieren kom med innspill og ønsker til PBI-er. Hovedsakelig foregikk dialogen om PBI-er mellom gruppen, men til tider sammen med RG i form av møter.

Kapittelet nevnte også sprint backloggen og formålet med artefaktet. Gruppen anvender det slikt at Scrum Master velger oppgaver som skal inkluderes i den nåværende iterasjonen etter diskusjon innad gruppen. Deretter blir oppgaver tildelt til utviklerne basert på deres ønsker og interesser, for å tilrettelegge at alle var engasjert gjennom utviklingen.

Order	ID	Priority	Title	State	Assigned To	Remaining Work
1	264	2	Lage art til lastning av innhold	Done	Dominykas Zuo...	
2	252	2	Importer grafpakke	Done	Bastian Ege Tveit	
3	196	2	Piechart total CO2	Done	Dung Katrine ...	
4	214	2	Liste kunder	Done	Eziz Ømer	
5	201	2	Tommganstimer graf insikt	Done	Dung Katrine ...	
6	209	2	Kundedetaljer	Done	Dominykas Zuo...	
7	210	2	statistikk (detaljevisning kunde)	Done	Dominykas Zuo...	
8	251	2	Hovedside Style	Done	Eirik Skogland ...	

Figur 29: Sprint backlog fra Sprint 4

For å gi et konkret eksempel på hvordan en sprint backlog kan se ut, vises sprint backloggen fra sprint 4 i figur 29. Dette eksemplet illustrerer hvordan oppgaver ble organisert for den kommende sprinten, samt gir en visuell representasjon av planleggingsprosessen og oppgavefordelingen.

4.3.3. Weekly Scrum

Daily- og Weekly Scrum er en prosess hvor medlemmene i et team planlegger hva de skal gjøre fremover. Dette innebærer å sette opp en liste over de tingene fra sprint backloggen som skal være ferdige denne dagen/uken, samt å skrive en logg over hva som har skjedd.

Etter et par uker med prøving og feiling, konkluderte gruppen med at det ville være mer hensiktsmessig å omdanne Daily Scrum til Weekly Scrum. Dette tiltaket resulterte i en mer effektiv bruk av gruppens arbeidstid og reduserte mengden av diskusjoner om problemer som ikke berørte hele gruppen.

I motsetning, gjennomførte gruppen Daily Standup møter daglig, med fokus på de tre standardspørsmålene som nevnt tidligere i kapittel 4.1.

4.3.4. Sprint Review og Sprint Retrospect

Gruppen hadde sprint reviews ved slutten av hver sprint, hvor alle interessenter ideelt sett skulle delta. Det hendte til tider at produkteier ikke hadde anledning til å møte opp. For at produkteieren ikke skulle gå glipp av diskusjonen, ble det gjort tiltak hvor det ble skrevet et detaljert referat av hver sprint review. Dette referatet ble sendt til produkteier for å sikre at han var oppdatert på prosjektets fremgang og eventuelle beslutninger.

I tillegg ble det sendt flere oppdateringsdemoer til produkteieren, som et resultat av hans ønske om jevnlig oppdateringer. Dette sikret at produkteieren ble holdt informert på prosjektets status og fremgang, med tanke på at produkteier ikke kunne delta fysisk i alle møter.

Samtidig ble sprint retrospect samkjørt med sprint reviews med produkteier og veileder fra universitetet. Under et review delte alle gruppe-medlemmer sine positive og negative erfaringer fra sprinten i form av en presentasjon. Målet med disse møtene var å redegjøre tiltak for å enten forbedre eller vedlikeholde de aspektene ved gruppens arbeid som var enten negativt eller positivt. I tillegg fikk gruppen verdifulle tilbakemeldinger fra veiledere om hvordan kvaliteten på produktet kunne forbedres.

Gruppen så sprint retrospect som spesielt viktig, ettersom at dette gav gruppen muligheten til å diskutere forbedringer som kunne gjøres i sprinten, samt hva gruppen tenkte å ta med videre til de neste sprintene. Dette resulterte i økt kvalitet i arbeidet mellom sprintene, siden utfordringer ble tatt opp i plenum og håndtert i fellesskap. For mer informasjon om utfordringer, refleksjon og hva gruppen ville ta med videre, se appendiks 5.

4.4. Pre-Sprint 1 & Sprint 2 – Analyse, Planlegging og Designstart

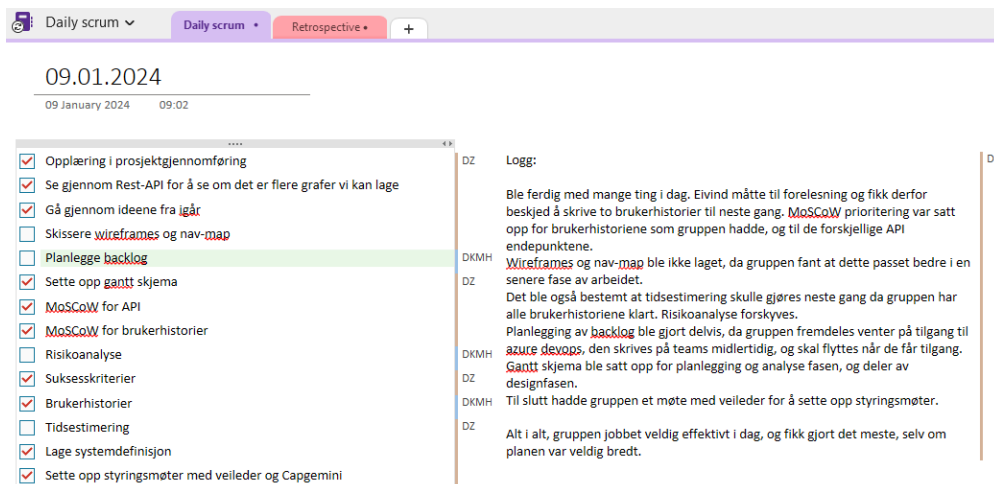
Dette delkapittelet legger frem forarbeidet som gav grunnlaget for prosjektet. Her ble det utført en rekke analyser og aktiviteter for å sette prosjektet i gang. Blant annet å definere suksesskriterier, lage systemdefinisjon og prioritering av REST-API endepunkter som skulle brukes i utgangspunktet. Prioriteringen er dokumentert i appendiks 2. Videre ble blant annet DevOps miljøet satt opp, og en Gantt-skjema ble utarbeidet. Disse aktivitetene bidro til å etablere en god struktur og klare mål for prosjektet.

4.4.1. Pre-Sprint

Pre-sprint ble benyttet til å samle og analysere store mengder med informasjon, blant annet å danne en forståelse av hva Capgemini og kunden representerer, samt klargjøre prosjektets oppgaver. Det ble også gjennomført diskusjoner med produkteier fra Capgemini om hans forventninger til gruppen og hvilke ambisjoner gruppen selv hadde satt.

Gruppen ble introdusert til et REST-API som ble utviklet av Capgemini for å utvikle kundeportalen IDA. Opprinnelig var planen at disse APIene skulle brukes til å skape en personlig portal for RG, som inkluderte kreative anvendelser. For å organisere arbeidet ble det utarbeidet en MoSCoW-prioriteringsliste for APIene og en risikomatrix for prosjektet.

Til tross for at gruppe-medlemmene tidligere hadde arbeidet sammen i andre scenarier, var dette teoretisk sett første gang de samarbeidet som en hel gruppe i ett prosjekt. Store ambisjoner fra enkelte medlemmer resulterte til et høyt antall oppgaver i pre-sprinten, i stedet for en felles diskusjon fra gruppen for å definere oppgavene.



Figur 30: Daily Scrum fra 9. januar 2024

Figur 30 viser hvordan den første dagen ble organisert, og viser at gruppen planla en stor mengde med oppgaver som skulle løses i løpet av dagen. I tillegg illustrerer den at ikke alle oppgaver ble gjennomført, noe som reflekterte gruppens ambisjonsnivå og utfordringer med å balansere realistiske mål.

4.4.2. Sprint 2

Sprint 2 markerte overgangen fra prosjektets analyse- og planleggingsfase til designfasen. I løpet av sprintens andre halvdel jobbet gruppen med forskjellige utkast til wireframes og konkluderte med en endelig versjon. Videre utviklet gruppen en mockup som var såpass detaljert at den fungerte som en prototype, med integrerte funksjoner. Det ble brukt mye tid på design, ettersom det var kontinuerlige diskusjoner med produkteier og kunde, som bidro med innspill til ønskede funksjoner og identifiserte forbedringsområder. I tillegg etablerte gruppen kontakt med flere frontend-utviklere fra Capgemini, som tilførte profesjonelle perspektiver og muligheter til design. Det var da gruppen ble informert om at kartvisning i applikasjonen var for omfattende i dette prosjektet. Capgemini var interessert i å senere implementere heatmap, men mente at det er en annen bacheloroppgave i seg selv. Derfor fjernet gruppen kartvisningsfunksjonen fra utviklingsprosessen.

4.5. Sprint 3 & Sprint 4 – Designslutt & Utviklingsstart

Sprint 3 markerte avslutningen på den planlagte designfasen og overgangen til utviklingsfasen, som fortsatte gjennom Sprint 4. I denne perioden slet gruppen fortsatt med tidsestimering, og det var ganger der enkelte medlemmer avviklet fra sprint backloggen og jobbet på oppgaver som fortsatt var relevante, men ikke planlagt for den aktuelle sprinten. Dette gjorde at perioden heller ble en kalibreringsperiode for gruppen.

4.5.1. Sprint 3

I denne sprinten delte gruppen seg i tre bolker: en bolk arbeidet med oppgaver relatert til IS-305, en annen la til endringer i prototypen i henhold til tilbakemeldinger og den tredje bolken satte opp kodebasen og installerte nødvendige avhengigheter. Blant annet ble gruppen introdusert til MudBlazor for gjenbrukbare komponenter og Syncfusion for grafer. Gruppen brukte mye tid på oppgavefordeling og effort-estimering for hver oppgave, noe som resulterte til overdreven tidsbruk på planlegging. Etter å ha diskutert med konsulenter fra Capgemini om tips på tidsestimering, endret gruppen strategi med å heller tildele større oppgaver og la utviklerne selv dele dem inn i håndterbare deloppgaver, etter deres egne forståelser. Dette forbedret arbeidsflyten til utviklerne og økte effektivitet rundt estimering i fremtidige sprinter.

Ved slutten av denne sprinten hadde gruppen et styringsmøte med veileder fra universitetet og produkteier fra Capgemini. Under dette møtet gikk gruppen gjennom prosjektgjennomføringen, hvor misforståelser om Waterfall og Agile ble avklart. Det ble også fremhevet behovet for å tilpasse tidsestimering for å lære nye konsepter, samt vektlegging av å opprettholde god kommunikasjon med produkteier. I slutten av møtet gikk partene gjennom forbedringer til presentasjonsteknikk. For utdypende informasjon se appendiks 6.

4.5.2. Sprint 4

Under Sprint 4 gjennomførte gruppen en revurdering av risikomatriksen og oppdaterte prototypen basert på tilbakemeldinger fra interessenter. Dette var viktig både for å få levert noe i hver sprint, samt for å holde gruppen oppdatert på potensielle risikofaktorer og ikke minst legge til nye funksjonsønsker fra RG. Selv om gruppen jobbet individuelt med utviklingsoppgaver, var de fysisk på kontoret for å enkelt kunne spørre hverandre om hjelp dersom noe skulle ha oppstått, eller for å ta opp noe i plenum.

4.6. Sprint 5 til Sprint 8 – Utvikling

Sprint 5 til 8 var hovedsakelig om utvikling, hvor gruppen hadde en fremgang i prosjektet etter å ha fått en bedre kompetanse innenfor flere områder. I sprint 6 som var rett før påskeferien ble fokuset satt på rapportskrivning i faget IS-305. Sprint 7 og 8 gikk ut på siste innsjutt av utviklingsdelen, hvor fokuset var på å finpusse koden og systemet.

4.6.1. Sprint 5

I Sprint 5 gjorde gruppen flere tekniske oppgaver etter å ha lært seg mer om SQL-databaser, grafpakken og generell kodeskriving. De fikk muligheten til å lære seg bruk av CTE for SQL-spørringer

fra en teknisk rådgiver hos Capgemini, noe som gjorde det enklere å kunne håndtere store spøringer. Mange av oppgavene som var planlagt for sprinten ble allerede fullført i løpet av den første uken. Dette kunne tyde på at omfanget var for lite, men gruppen mente heller at økt kunnskap fra tidligere sprints førte til en effektivisering i arbeidet. I tillegg ble det holdt en demo for interessentene hvor gruppen fikk tilbakemeldinger om hvilke elementer som kunne fjernes, ønskede funksjoner eller hvilke konsepter som trengte mer utdypning.

4.6.2. Sprint 6

Sprint 6 var en forkortet periode som kun varte en uke på grunn av påskeferie. I denne perioden ble det full fokus på rapportskrivning for faget IS-305 og videre fullførte gruppen første utkast av rapporten. De forberedte også på en presentasjon i faget, hvor de delte sine funn og mottok tilbakemeldinger med andre grupper. Programmeringsarbeid ble derfor nedprioritert i denne sprinten, noe som var håndterbart ettersom gruppen allerede lå godt an i henhold til *product backlog*.

4.6.3. Sprint 7

Sprint 7 tok for seg siste innsjutt av programmeringen hvor fokuset var sikkerhetsanalyse, feilhåndtering samt filtrering etter tidsrom. Det ble gjennomført penn-testing og unit-testing lokalt, i tillegg til håndtering av feil og sårbarheter i koden. Noe gruppen innså i sprint 7 var at fokus på testing, kodekvalitet og strengere code reviews tidligere i utviklingsprosessen kunne ha spart gruppen tid på bugfixes i tillegg til å opprettholde en god kodestruktur.

Ved slutten av sprinten hadde gruppen et styringsmøte med veileder fra universitetet og produkteier. Gruppen viste fram flere visualiseringer av forbruk og utslipp i form av grafer og tabeller. Grafene ble videreutviklet med å ha valg av tidsrom for flere grafer slik at RG kan se utslipp og forbruk i spesifikke perioder. Gruppen fokuserte mye på å sette opp bug tasks og fikse dem. Til slutt, reflekterte gruppen over sikkerhetstiltak og testing gjennomført i applikasjonen med veileder og produkteier. For mer informasjon om styremøte se referat i appendiks 7. Resten av sprinten gikk ut på å ferdigstille koden, løse de siste feilene og holde en demo for ansatte hos Capgemini med det ferdige produktet.

4.6.4. Sprint 8

Sprint 8 utgjør rapportskrivningen og små tekniske endringer for dette prosjektet. I starten av sprinten fikset gruppen bugs og la til mer dokumentasjon i koden, samt utarbeidet en liste over kjente feil i systemet og forberedte systemet for en handover. Videre, begynte gruppen med rapportskrivning for prosjektet. Gruppen holdt regelmessig kontakt med veileder, hvor gruppen fikk tilbakemeldinger om rapportskrivning.

I tillegg var det et styringsmøte med veileder. Produkteier kunne dessverre ikke stille opp grunnet sykdom og ble derfor oppdatert gjennom Teams og epost, hvor gruppen stilte noen spørsmål om overlevering av prosjektet. Under møtet fikk gruppen flere råd fra veileder om rapporten, dette gikk blant annet ut på endringer i innhold og struktur. For mer informasjon om styringsmøtet se appendiks 8.

4.7 Kvalitetssikring

Etter å ha definert kvalitetskriteriene i underkapittelet 2.1.5 for dette prosjektet, er det viktig å sikre at kvaliteten har blitt ivaretatt. Gruppen implementerte flere tiltak og aktiviteter for å sikre ønsket kvalitet.

4.7.1 Produktet tilfredsstillende både produkteier og brukerne/kunden

Under hele prosjektutviklingen forholdet gruppen seg til produkteieren, og har tatt imot innspill fra RG for funksjonalitet ettersom at de er brukergruppen. Ved å være oppmerksom på suksesskriterier, brukerhistorier og systemdefinisjonen under sprintplanleggingen, har gruppen implementert funksjoner som dekker produkteierens krav og ønsker. Det er viktig å nevne at gruppen har både utelatt og lagt til nye funksjoner etter møter med produkteier og RG. Resultatet fra dette er et system som dekker beskrivelsen av systemdefinisjonen, hvor alle nødvendige og ønskede funksjoner fra interessentene er implementert.

Gruppen holdt flere demoer med RG gjennom semesteret. Dette gav gruppen bredere innblikk i hva sluttbrukernes behov og ønsker var. Til tross for at gruppen ikke fikk testet sluttbrukerne i RG, har flere ansatte hos Capgemini fungert som brukere. Som nevnt i 2.2.6, ble det gjennomført seks brukertester under designprosessen. I tillegg, ble det gjennomført sju ende-til-ende-tester under hele utviklingsprosessen. Ende-til-ende-testene gav gruppen verdifull tilbakemelding som ble brukt for å forbedre systemets frontend, med kontinuerlige endringer etter hver test.

Det ble gjennomført totalt tretten tester med flere brukere i ulike aldersgrupper, bakgrunn og kjønn, tilsvarende ble det gjennomført tretten forbedringsiterasjoner gjennom prosjektets levetid. Ved å teste en mangfoldig brukergruppe, fikk gruppen en bredere forståelse av hvordan forskjellige brukergrupper kan samhandle med systemet.

4.7.2 God kodekvalitet i systemet

Gjennom prosjektet iverksatte gruppen flere tiltak for å bidra til at kodekvalitet skulle bli så god som mulig:

Sette felles kodestandarder og tiltak for kodekvalitet

Kodestandarder er retningslinjer og regler som utviklere følger når de skriver kode (Codacy Quality, 2023). Disse reglene for kode er nødvendige for å opprettholde en strukturert kodebase og lette samarbeidet mellom utviklere (Codacy Quality, 2023). Formålet ved å sette kodestandarder for utviklingen av systemet er å gjøre løsningen effektiv, forståelig, lett å vedlikeholde og skalerbar (University of St. Andrews, n.d.).

I begynnelsen av utviklingsprosessen definerte gruppen felles kodestandarder:

- Kommentere funksjoner i koden.
- En klasse per fil.
- Gruppere filer relatert til hverandre for å få oversikt over filene.
- Fokus på å implementere løs coupling og høy cohesion.
- Ha beskrivende navn på funksjoner og variabler.

I tillegg til disse, arbeidet gruppen med å opprettholde konsistent formatering og struktur i koden for å bedre lesbarhet (Knuth, n.d.). Gruppen la også vekt på å skrive klare og konsise kommentarer for funksjonene i koden. Det var også et mål å unngå lange kodelinjer; kodelinjene i systemet ble holdt under 120 tegn. Gruppen jobbet med å holde funksjonene korte og dedikerte hver funksjon til en spesifikk oppgave, med noen unntak for komplekse CTE spørringer til databasen. CTE-spørringene krevde mer tekst siden de henter ut spesifikk data fra flere tabeller i databasen, noe som ble gjort for å øke spørringseffektiviteten i systemet. Ved å skrive korte funksjoner som gjør én spesifikk oppgave kunne gruppen gjenbruke deler av koden. For å håndtere eventuelle feil, implementerte gruppen «exceptions» for å forhindre systemkrasj. Dette gjorde koden mer pålitelig og feilsikker. I tillegg til å ha felles kodestandarder under utviklingsprosessen, iverksatte gruppen aktiviteter for å sikre god kodekvalitet:

- **Code Reviews** – gruppen gjennomførte code reviews for hverandre i gruppen. Ved å ha flere personer som ser gjennom koden, var det lettere å oppdage feil eller finne bedre måter å utføre oppgaver på.
- **Pull Request** – Ved å gjennomgå og godkjenne hverandres Pull Requests, sørger gruppen for at flere ser på koden og sikrer at koden er forståelig og korrekt.
- **Refaktorering av kode ved hjelp av analyseverktøy** – hjelper med å se varslinger i koden. I tillegg hjelper analyseverktøyet gruppen med å finne feil og bugs i koden, oppdage sikkerhetsproblemer, bidra til å optimalisere koden og passer på at koden følger kodestandarder.

- **Bruk av veiledere** – gruppen fikk tilgang til hjelp og rådgivning fra erfarne programmere i Capgemini. Veilederne viste hvordan gruppen kunne effektivisere koden i prosjektet, dette gjelder spesielt SQL-spørringer.

Disse tiltakene bidro til kode som er mer effektiv, forståelig og oversiktlig med god dokumentasjon.

Kodetesting

Testing er avgjørende for å sikre at hver funksjon utfører sin forventede oppgave og at kravene til systemet er oppfylt (Microsoft, 2023). Under prosjektgjennomføringen har gruppen benyttet seg av manuelle tester.

Manuell testing er en type programvaretesting hvor gruppen testet applikasjonen manuelt uten hjelp av testverktøy. Gruppemedlemmene samhandlet med systemet på samme måte som en sluttbruker ville, for å identifisere feil, bugs og problemer som påvirker brukeropplevelsen negativt (Katalon, n.d.).

Manuell testing tilbyr fleksibilitet til å tilpasse seg endringer ved testtilfeller uten behov for omfattende omskriving av koden (Testscenario, n.d.). Disse testene krever ikke komplekse verktøy eller miljø, noe som reduserer ressursforbruk (Testscenario, n.d.). Manuelle tester fremmer bedre kommunikasjon og forståelse mellom medlemmene, og gir dem en bedre forståelse av funksjonene i systemet (Testscenario, n.d.).

4. Endelig produkt

Dette kapitlet forklarer funksjonaliteten til applikasjonen, og gir en kort oppsummering på veien videre for prosjektet.

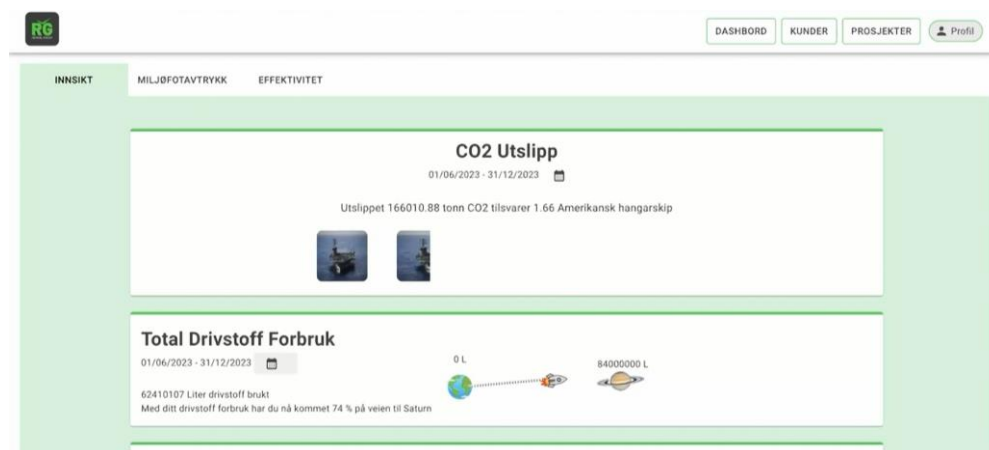
5.1 Funksjonalitet

Webapplikasjonen som er blitt utviklet har som hensikt å gi ledelsen i RG et innsyn i driften av deres maskiner og utslippsdata, samtidig som den leverer relevant kunde og prosjekt informasjon. Denne delen vil fokusere på de ulike funksjonalitetene til webapplikasjonen.

Applikasjonen er strukturert rundt tre hovedsider: "Dashbord", "Kundedetalj" og "Prosjektdetalj". Dashbord er delt opp i tre underdeler som har fokus på ulik data. Innsikt siden viser generell data med visualisering, miljøfotavtrykk er fokusert på å vise data relatert til miljøfotavtrykket til forskjellige kunder, og effektivitet har som mål å sammenligne effektiviteten til ulike prosjekter og maskiner. Mer forståelse rundt hver hovedside blir dypere forklart i de neste avsnittene.

5.1.1 Innsikt-side

Figur 31 viser innsikt-siden, den har som oppgave å gi brukeren en rask oversikt over det totale forbruket til RG sine kunder, inkludert CO₂-utslipp, drivstofforbruk og forbrukstimer på maskinene. For å gi brukeren en mer detaljert innsikt, inneholder flere moduler en kalenderfunksjon som lar brukeren justere tidsrammen for datainnhenting.

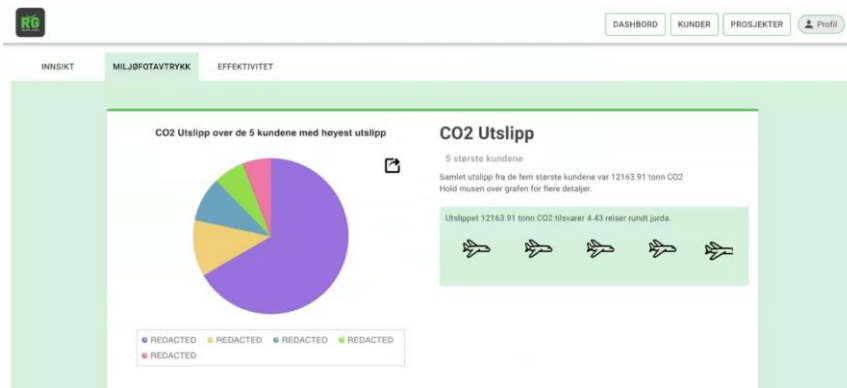


Figur 31: Innsikt-side

5.1.2 Miljøfotavtrykk-side

På "Miljøfotavtrykk"-siden presenteres CO₂-utslippsdata fra de fem største kundene i form av et kakediagram. Når brukeren beveger muspekeren over de forskjellige delene av kakediagrammet, vises informasjon om utslippet til den aktuelle kunden i et infofelt ved siden av diagrammet. Ved å

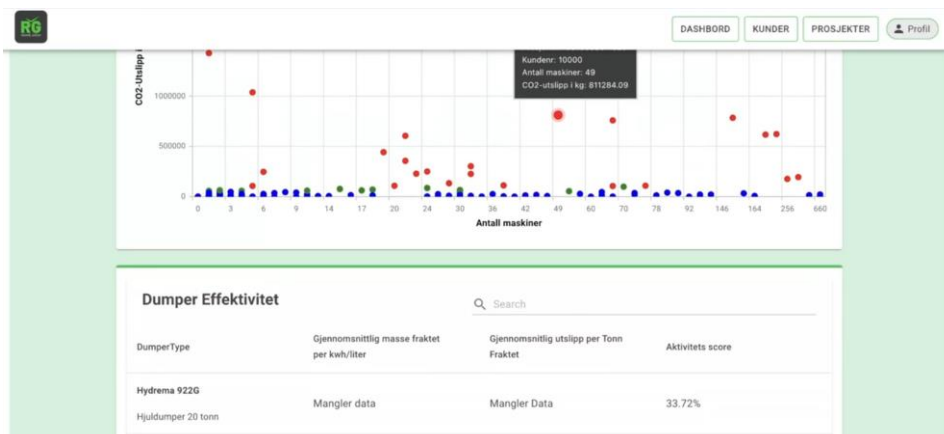
klikke på et stykke av kakediagrammet, blir brukeren tatt til kundesiden. Videre ned på siden vises et stolpediagram som viser trenden for CO₂-utslipp og driftstimer for elektriske og vanlige maskiner på månedsbasis. En del av miljøfotavtrykk-siden kan en se i figur 32.



Figur 32: Miljøfotavtrykk-side

5.1.3 Effektivitet-side

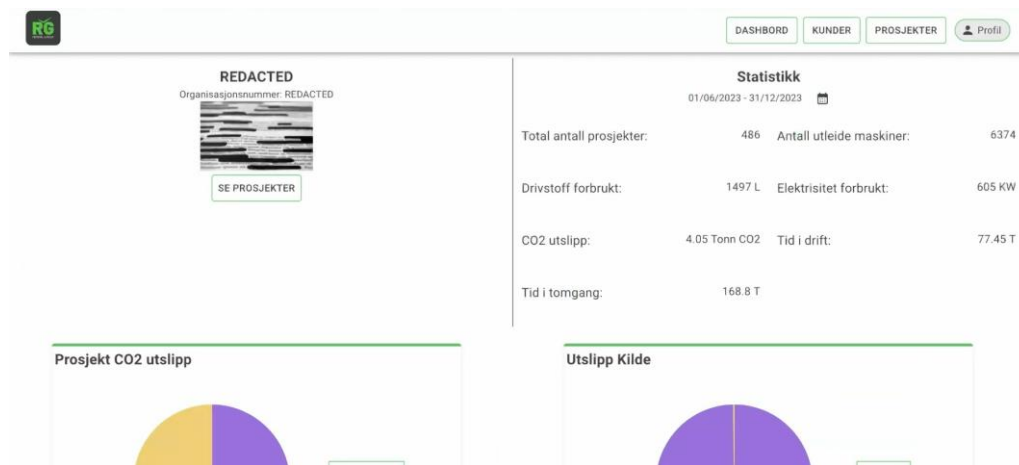
På "Effektivitet"-siden finner brukeren et scatterplot-diagram som viser utslippet til alle prosjektene, sammen med antall maskiner per prosjekt. Når brukeren beveger muspekeren over et punkt i diagrammet, vises mer informasjon om prosjektet. Ved å klikke på punktet, tas brukeren til prosjektsiden hvor det vises ytterligere informasjon om det prosjektet. Under scatterplotten vises en liste over forskjellige dumpere i RGs sortiment, samt oversikt over effektivitet, aktivitet, masse fraktet og gjennomsnittlig utslipp per tonn fraktet. Nederst på siden kan brukeren velge et prosjekt fra en prosjektliste for å få mer data om maskin aktiviteten på det prosjektet. Når et prosjekt er valgt, vises en liste over maskiner brukt i prosjektet, inkludert informasjon som operative timer, driftstimer, aktivitetsscore og utslipp per maskin. Deler av effektivitet siden vises i figur 33.



Figur 33: Effektivitet-side

5.1.4 Kundedetaljer-side

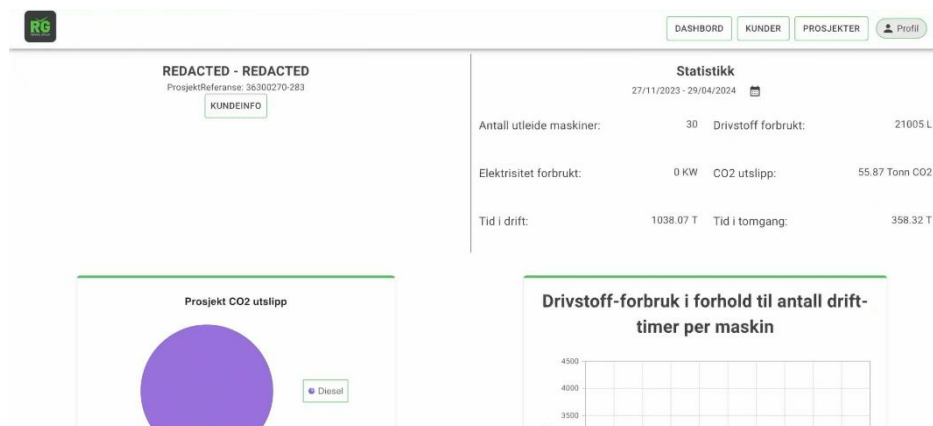
"Kundedetaljer"-siden, illustrert i figur 34 viser informasjon og data fra en spesifikk kunde. Siden har en knapp som viser fram en liste over de forskjellige prosjektene til kunden, et statistikkfelt som viser forskjellig kundedata, to kakediagrammer der den ene viser CO2-utslipp fra de største prosjektene og den andre som viser fordelingen av drivstoff og elektrisitetsforbruk. På bunnen av siden finner brukeren et scatterplot-diagram som viser utslipp og antall maskiner fra de forskjellige prosjektene til kunden. Hvis brukeren trykker på et av punktene i scatterplot diagrammet eller prosjekt kakediagrammet, så blir brukeren tatt videre til prosjektets detaljeside.



Figur 34: Kundedetaljer-side

5.1.5 Prosjektdetaljer-side

I figur 51 kan en se "Prosjektdetaljer"-siden, som viser informasjon og statistikk fra et utvalgt prosjekt. Siden inneholder et statistikkfelt med diverse informasjon om prosjektet, et kakediagram som viser utslippsfordelingen av drivstoff og elektrisitetsbruk hos maskinene, i tillegg til et scatterplot-diagram som viser de forskjellige maskinene som har jobbet på prosjektet og deres CO2 utslipp, tomgangstimer og modellnavn.



Figur 35: Prosjektdetaljer-side

5.2 Veien videre

Produktet har blitt overlevert til Capgemini sammen med en README-fil, som forklarer hvordan man setter opp systemet. I tillegg ble det lagt til en liste over kjente problemer. Gruppen har lagt stor vekt på god dokumentasjon av koden for å sikre at den er lett forståelig for Capgemini. Dette vil gjøre det enklere for dem å vedlikeholde og potensielt videreutvikle produktet etter eget behov.

5. Refleksjon

I dette kapitlet skal gruppen reflektere over prosjektet i sin helhet. Kapitlet diskuterer om gruppen oppnådde de forskjellige kravene som ble satt, hva de vil ta med seg videre, de forskjellige utfordringene og begrensningene som oppstod underveis, i tillegg til hva de hadde gjort annerledes med det de har lært.

6.1 Sosialt

I starten av prosjektarbeidet jobbet gruppen veldig hardt og effektivt i lang strekk, med lite fokus på det sosiale. Et problem som oppstod av dette, var at flere gruppemedlemmer begynte å bli slitne av å jobbe konsentrert over en lengre periode, noe som i et par tilfeller førte til dårlig stemning under gruppearbeidet. Som tiltak på dette, organiserte gruppen flere økter med teambuilding, noe som hadde en klart synlig effekt, og fikk alle til å være i bedre humør. Teambuilding bidro til mer åpenhet innad gruppen, hvor de følte seg mer trygge i å stille spørsmål og ta opp utfordringer med prosjektet. Gruppen synes at det er åpenbart at de burde ha begynt med slike aktiviteter helt fra starten av prosjektet, ikke bare for å få til mer effektivt arbeid, men for å ha det enda mer kjekt i dette lærerike semesteret.

6.2 Prosjektgjennomføring

Dette delkapitlet skal ta for seg gruppens utfordringer og hvordan de ble reflektert under prosjektet. Her reflekteres det over fordeling av oppgaver, kommunikasjon innad gruppen, tidsestimering av oppgaver og kommunikasjonsutfordringer med sluttbrukere.

6.2.1 Fordeling av oppgaver

Siden prosjektets begynnelse har gruppen hatt stor nytte av å sette opp flere planer i tilfelle noen av oppgavene ikke lot seg gjennomføres, om det var på grunn av forsinkelser eller andre uforutsette hendelser. Ved hjelp av risikoanalyser og detaljert planlegging har gruppen alltid hatt oppgaver å jobbe med, selv under perioder der gruppen måtte vente på input fra både universitetet og oppdragsgiver, eller ved varierende fremgang i prosjektet.

Det ble brukt mye tid i planlegging, noe som bidro til en effektiv prosjektgjennomføring. Til tross for dette ble det brukt en større andel av tid og ressurser enn det gruppen hadde ønsket. For å bruke mindre tid på planlegging og distribuere tiden mer effektivt, gikk gruppen over til Weekly Scrum i stedet for Daily Scrum, etter anbefaling fra produkteier. Dette tiltaket ga gruppen mer tid til å jobbe

med oppgaver relatert til applikasjonen. Selv om tiden som ble brukt på planlegging ble redusert, var det fremdeles tilstrekkelig for å håndtere uforutsette problemer.

6.2.2 Tidsestimering

Tidsestimering var en kontinuerlig utfordring igjennom prosjektet. Ikke alle gruppe-medlemmene var godt kjent med hverandre fra før, noe som gjorde det vanskelig å estimere oppgaver. Når gruppen var uenige i hvor mye effort en PBI skulle ha, ble gruppen oftest enige etter diskusjon å gi PBI-ene høyere effort. Etter hvert som gruppen ble mer kjent med hverandres styrker og svakheter, og fikk mer kompetanse i teknologiene som ble brukt, begynte gruppen å samkjøre estimeringene sine med samme effort.

Selv om gruppen forbedret seg på tidsestimering av oppgavene, kom gruppen aldri helt i mål. En gjentakende tendens som nevnt i underkapittel 4.2.3 er at gruppen ofte undervurderte sin egen kapasitet, noe som førte til at PBI-er ble tildelt høyere effort enn nødvendig. I ettertid innså gruppen at de burde i flere tilfeller ha brukt ekstra tid på grundigere code reviews og feilsøking. I sprintene hvor en kan se drastisk nedgang i estimert efforts, valgte gruppen å legge til flere PBI-er for å sikre at alle medlemmer hadde oppgaver å jobbe med.

Noe som gruppen vil ta med videre fra tidsestimering er å ha mer tro på gruppen sin kompetanse og ikke overestimere PBI-er. I framtiden vil gruppen også bruke mer tid på å kvalitetssikre oppgavene som er gjort ved feilsøking og bruke mer tid på code reviews istedenfor å hoppe direkte til neste oppgave.

6.2.3 Kommunikasjon med endelig bruker

Gjennom prosjektet har flere problemer oppstått som følge av at endelig bruker (RG) har vært vanskelig å få tak i. Selv om gruppen fikk beskjed at de skulle forholde seg til produkteieren hos Capgemini, innså de at for å lage et system som sluttbrukeren skal få nytte av, så var det hensiktsmessig å også ha kontakt med RG. I løpet av prosjektet fikk gruppen til noen møter med sluttbruker, men siden de ikke fikk så mange, så måtte gruppen stole på forståelsen de forskjellige kontaktpersonene fra Capgemini hadde for brukerens behov.

I etterkant har gruppen innsett at de burde ha forberedt seg bedre på de få møtene de hadde med endelig bruker. For eksempel, å ha gitt de muligheten å utforske systemet selv med et fjerntilgangsprogram. Dette hadde vært et nyttig verktøy for å samle inn tilbakemelding på funksjonalitet.

6.3. Utvikling

Gruppen har hatt flere utfordringer under utviklingsfasen i prosjektet, blant annet manglende og negative tall i databasen, dårlig optimalisering av koden i spørringer og kodekvalitet som ikke var opp til standard. I dette delkapittelet skal gruppen forklare mer om problemene som oppstod, og hvordan gruppen valgte å løse disse problemene.

6.3.1. Database og Datakvalitet

Gruppen mottok en databasedump fra Capgemini som de skulle forholde seg til. Det viste seg at dataen i databasedumpen ikke hadde nok informasjon, dette førte til at det ikke var mulig å visualisere enkelte ideer som gruppen kom opp med. Selv om gruppen kunne ha valgt å legge inn simuleringsdata (mock data) for å omgå denne begrensingen, ble det besluttet at gruppen ønsket å vente på en eventuell oppdatert versjon av databasedumpen. I retrospekt var dette en god beslutning, ettersom gruppen senere fikk beskjed fra Capgemini om at de var interesserte i å implementere en ny database i systemet etter overtakelse, og foretrakk at systemet hadde endepunkter til riktige datasett.

Verifikasjon av innhentet data viste seg å være en stor utfordring, grunnet begrenset kjennskap til databasen og den spesifikke bransjen. Dette gjorde det utfordrende å oppdage om tallene var riktig eller ikke. Gruppen brukte derfor mye tid på å bli kjent med hvordan dataene var organisert i databasen. Samtidig ble mer komplekse spørringer delt opp i mindre segmenter. Hvert segment ble videre testet for å sjekke at dataen faktisk stemte overens med forventningene. Denne tilnærmingen var nødvendig å sikre nøyaktighet og troverdighet i dataene som ble brukt.

Gruppen burde ha brukt mer tid på å verifisere hva som er riktig data i forhold til både utslipp, antall maskiner per prosjekt, og drifts/tomgangs- timer tidligere i prosjektet. Gruppen fikk ofte ut forskjellige tall fra databasen på grunn av forskjellig spørrings-type, oppsett, struktur og datarensing. På grunn av dette måtte gruppen i flere tilfeller få noen ned fra bedriften og bruke mye tid på å feilsøke på hvorfor tall ikke stemte i forhold til hverandre sent i prosjektet. Hvis gruppen derimot hadde brukt noen dager tidlig i utviklingsfasen på å kartlegge riktige tall, hadde de spart mye tid utover utviklingsprosessen.

6.3.2. Optimalisering av kode

Applikasjonen krevde avanserte spørringer til databasen for å visualisere dataen på ønsket måte. En utfordring som oppstod på grunn av de avanserte spørringene var at de tok for lang tid, og koblingen til databasen ble frakoblet som et resultat av dette. En rådgiver fra Capgemini introduserte gruppen til CTE spørringer som hjalp gruppen med å få de fleste spørringene effektive nok til å hente ned data

vellykket, likevel er det fremdeles en sjanse for at applikasjonen vil vise en feilmelding istedenfor data på grunn av at store mengder data fører til at spørringene tar for lang tid.

Kvalitetssikring av kodekvalitet ble mer utfordrende enn forventet. I starten av prosjektet ble det satt kodestandarder som skulle følges, men etter hvert kom gruppen frem til at kodestandardene som hadde blitt satt ikke var tilstrekkelig, og gruppemedlemmene hadde vært for snille på code reviews. Dette førte til at når gruppen nærmet seg slutten av utviklingsprosessen, var ikke kodekvaliteten opp til standarden som gruppen ønsket. Disse utfordringene ble løst ved å sette flere tiltak med tanke på utfordringene, blant annet å revurdere kodestandardene, bli strengere på code reviews og å ta i bruk automatiserte verktøy for kodeanalyse. Dette førte til at den endelige kodekvaliteten på prosjektet var akseptabel.

6.4. Har gruppen oppnådd kravene?

Det ble diskutert i et tidligere kapittel om hva Capgemini ønsket seg; et Proof of Concept som visualiserer miljødata fra anleggsmaskinene til RG. Webapplikasjonen er designet for å tilføre verdi til RG ved å fremheve hvilke beslutninger som kan tas for å redusere utslipp, samt identifisere unødvendige utslippskilder. Selv om det er usikkert om absolutt alle funksjonene i applikasjonen vil bli brukt, er det flere funksjoner som har vist seg å være relevante for RG. Disse inkluderer blant annet effektivitetssiden som viser til dumper effektivitet og maskinaktivitet for prosjekter, som de i tillegg har uttrykt interesse for å benytte eller videreutvikle. Ved at gruppen har lagt tall-verdier på utslipp og forbruk i perspektiv med bilder og grafer for å hjelpe RG med å kartlegge og forstå deres miljøfotavtrykk. Med dette som utgangspunkt, mener gruppen at systemet som har blitt utviklet vil hjelpe RG i å ta mer bærekraftige beslutninger.

Når det gjelder gruppen og deres målsettinger for prosjektet, var det viktigste at applikasjonen kunne tilby kreative løsninger for visualisering av miljødata, samtidig som at alle i gruppen fikk verdifull erfaring innen utvikling og prosjekthåndtering. Gruppen mener at de har nådd målene som de selv har lagt opp, i tillegg til å oppfylle de akademiske kravene. Hvert medlem har håndtert et bredt spekter av oppgaver, fra enkle til komplekse og fra små til store. Dette har gitt et lærerikt semester der alle har utviklet nye ferdigheter og blitt mer trygge på programvareutvikling. Videre har gruppen utforsket nye rammeverk og teknologier som ikke bare har vært relevante for applikasjonen, men som også vil være nyttig for fremtidig karriere innen teknologibransjen.

Referanser

- Altinn. (u.d.). *Kodebibliotek*. Hentet fra Altinn: <https://altinn.github.io/docs/seres/kodebiblioteket/>
- Amazon. (u.d.). *What's the Difference Between Frontend and Backend in Application Development?* Hentet fra Amazon: <https://aws.amazon.com/compare/the-difference-between-frontend-and-backend/>
- Atlassian. (2023, August 30). *Understanding the MoSCoW prioritization | How to implement it into your project*. Hentet fra Atlassian: <https://community.atlassian.com/t5/App-Central/Understanding-the-MoSCoW-prioritization-How-to-implement-it-into/ba-p/2463999>
- Atlassian. (u.d.). *How to conduct a successful sprint retrospective*. Hentet fra Atlassian: <https://www.atlassian.com/team-playbook/plays/retrospective>
- Atlassian. (u.d.). *Waterfall Methodology: A Comprehensive Guide*. Hentet fra Atlassian: <https://www.atlassian.com/agile/project-management/waterfall-methodology>
- Atlassian. (u.d.). *What is scrum and how to get started*. Hentet fra Atlassian: <https://www.atlassian.com/agile/scrum>
- Atlassian. (u.d.). *What is version control?* Hentet fra Atlassian: <https://www.atlassian.com/git/tutorials/what-is-version-control>
- Aven, T. (2023, Januar 25). *risikoanalyse*. Hentet fra Store Norske Leksikon: <https://snl.no/risikoanalyse>
- Benyon, D. (2019). *Designing user experience*. Pearson Education Limited.
- Brush, K. (2023, Mars). *MoSCoW method*. Hentet fra Techtarget: <https://www.techtarget.com/searchsoftwarequality/definition/MoSCoW-method>
- Capgemini. (u.d.). *About Us*. Hentet fra Capgemini: <https://www.capgemini.com/about-us/>
- Codacy Quality. (2023, November 14). *Code Standards: What Are They and Why Are They Important?* Hentet fra Codacy.com: <https://blog.codacy.com/coding-standards>
- Coursera. (2023, November 29). *What Is a Mockup?* Hentet fra Coursera: <https://www.coursera.org/articles/what-is-mockup>
- Derrick, B. (2020, mai 5). *The Advantages of Good Design Principles*. Hentet fra medium: <https://medium.com/@dbills777/2250-a681433d1d4b>
- Docker Inc. (u.d.). *Use containers to Build, Share and Run your applications*. Hentet fra Docker: <https://www.docker.com/resources/what-container/>
- Figma. (u.d.). *What is Figma?* Hentet fra Figma: <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>
- Frich, J. (2011, Oktober 1). *Kvalitet, kvalitetsforbedring og pasientsikring*. Hentet fra Universitetet i Oslo: <https://www.med.uio.no/studier/ressurser/fagsider/klok/info-fagplanutvalg/kvalitet-og-pasientsikkerhet.html>

- GeeksForGeeks. (2023, April 18). *Frontend vs Backend*. Hentet fra GeeksForGeeks: <https://www.geeksforgeeks.org/frontend-vs-backend/>
- GeeksForGeeks. (2023, Desember 7). *Manual Testing vs Automated Testing*. Hentet fra GeeksForGeeks: <https://www.geeksforgeeks.org/software-engineering-differences-between-manual-and-automation-testing/>
- GfG. (2023, Desember 5). *User Stories in Agile Software Development*. Hentet fra geeksforgeeks: <https://www.geeksforgeeks.org/user-stories-in-agile-software-development/>
- Gillis, A. S. (2023, Mars). *Proof of Concept*. Hentet fra TechTarget: <https://www.techtarget.com/searchcio/definition/proof-of-concept-POC>
- Gundersen, D. (2018, Mai 28). *Kvalitet*. Hentet fra Store norske leksikon: <https://snl.no/kvalitet>
- Halbo, L. (2020, Januar 6). *Kvalitetssikring*. Hentet fra Store norske leksikon: <https://snl.no/kvalitetssikring>
- Harris, C. (u.d.). *Agile scrum artifacts*. Hentet fra Atlassian: <https://www.atlassian.com/agile/scrum/artifacts>
- Html Agility Pack. (u.d.). *Html Agility Pack (HAP)*. Hentet fra Html Agility Pack: <https://html-agility-pack.net/>
- JetBrains. (u.d.). *Code quality done right*. Hentet fra JetBrains: <https://www.jetbrains.com/qodana/>
- Katalon. (u.d.). *What is Manual Testing? A Comprehensive Guide (With Examples)*. Hentet fra Katalon: <https://katalon.com/resources-center/blog/manual-testing>
- Keup, M. (2022, April 21). *Burndown Chart: What Is It & How to Use One for Agile*. Hentet fra Projectmanager: <https://www.projectmanager.com/blog/burndown-chart-what-is-it>
- Knuth, D. (u.d.). *Coding Standards*. Hentet fra Florida State University: <https://www.cs.fsu.edu/~lacher/courses/DOCS/codestandards.html>
- Lowdermilk, T. (2013, April). *User-Centered Design*. Hentet fra Google books: <https://books.google.no/books?id=XiX5bNjW0kC&lpg=PR2&dq=what%20is%20user%20centered%20design&lr&hl=no&pg=PP1#v=onepage&q&f=false>
- Martin, R. C. (2017). *Clean Architecture - A Craftsman's Guide to Software Structure and Design*. Boston: Pearson Education, Inc.
- Mathiassen, L., Munk-Madsen, A., Nielsen, P. A., & Stage, J. (2018). *Object Oriented Analysis & Design 2. edition*. Hadsund: Metodica ApS.
- Microsoft. (u.d.). *.NET Framework documentation*. Hentet fra Microsoft: <https://learn.microsoft.com/en-us/dotnet/framework/>
- Microsoft. (2023, Desember 12). *Testing in .NET*. Hentet fra Microsoft: <https://learn.microsoft.com/en-us/dotnet/core/testing/>
- Microsoft. (2023, November 14). *The C# Type System*. Hentet fra Microsoft: <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/types/>

Microsoft. (2024, Januar 10). *Introduction to .NET*. Hentet fra Microsoft:
https://learn.microsoft.com/nb-no/dotnet/core/introduction?WT.mc_id=dotnet-35129-website

Microsoft. (2024, Mars 25). *Overview of Services*. Hentet fra Microsoft Build:
<https://learn.microsoft.com/en-us/azure/devops/user-guide/services?toc=%2Fazure%2Fdevops%2Fget-started%2Ftoc.json&view=azure-devops>

Microsoft. (2024, Mai 4). *Unit testing C# with NUnit and .NET Core*. Hentet fra Microsoft:
<https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-nunit>

Microsoft. (u.d.). *ASP.NET Core*. Hentet fra Microsoft: <https://dotnet.microsoft.com/en-us/apps/aspnet>

Microsoft. (u.d.). *ASP.NET Core Blazor*. Hentet fra Microsoft Learn: <https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-8.0>

Microsoft. (u.d.). *Build beautiful web apps with Blazor*. Hentet fra Microsoft:
<https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor>

Microsoft. (u.d.). *C#*. Hentet fra Microsoft: <https://dotnet.microsoft.com/en-us/languages/csharp>

Microsoft. (u.d.). *Entity Framework documentation hub*. Hentet fra Microsoft:
<https://learn.microsoft.com/en-us/ef/>

Microsoft. (u.d.). *Introduction to SignalR*. Hentet fra Microsoft Learn:
<https://learn.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr>

Microsoft. (u.d.). *Model-View-ViewModel (MVVM)*. Hentet fra Microsoft Learn:
<https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>

Microsoft. (u.d.). *New: Enhanced managability from Azure Arc*. Hentet fra Microsoft:
<https://www.microsoft.com/en-us/sql-server>

Mjølund, Ø. (2024, April 20). personlig kommunikasjon. Kristiansand.

NUnit. (u.d.). *Downloads*. Hentet fra NUnit: <https://nunit.org/download/>

NUnit. (u.d.). *What is NUnit?* Hentet fra NUnit: <https://nunit.org/>

Radigan, D. (u.d.). *Agile sprint reviews*. Hentet fra Atlassian:
<https://www.atlassian.com/agile/scrum/sprint-reviews>

Radigan, D. (u.d.). *Story points and estimation*. Hentet fra Atlassian:
<https://www.atlassian.com/agile/project-management/estimation>

Rehkoph, M. (u.d.). *Scrum sprints*. Hentet fra Atlassian:
<https://www.atlassian.com/agile/scrum/sprints>

Rental Group. (2023, Mars 14). *RENTAL GROUP INNGÅR AVTALER MED STORE PRODUSENTER AV NULLUTSLIPPSMASKINER*. Hentet fra Rental Group: <https://rentalgroup.no/rental-group-inngar-avtaler-med-store-produsenter-av-nullutslippsmaskiner/>

Rental Group. (u.d.). *Om Rental Group* . Hentet fra Rental Group: <https://rentalgroup.no/om-oss/om-rental-group/>

Rental Group. (u.d.). *Vi lanserer IDA!* Hentet fra Rental Group: <https://rentalgroup.no/ida/>

Syncfusion. (u.d.). *What are the benefits of Blazor over Angular, React, or other JavaScript frameworks?* Hentet fra Syncfusion: <https://www.syncfusion.com/faq/1-what-are-the-benefits-of-blazor-over-angular-react-or-other-javascript-frameworks>

Testscenario. (u.d.). *Advantages & Disadvantages of Manual Testing*. Hentet fra Testscenario: <https://www.testscenario.com/advantages-and-disadvantages-of-manual-testing/>

Tutorialspoint. (u.d.). *MS SQL Tutorial*. Hentet fra Tutorialspoint: https://www.tutorialspoint.com/ms_sql_server/index.htm

Tutorialspoint. (u.d.). *SQL - Common Table Expression (CTE)*. Hentet fra Tutorialspoint: <https://www.tutorialspoint.com/sql/sql-common-table-expression.htm>

Universitetet i Agder. (u.d.). *IS-304 Bacheloroppgave i informasjonssystemer*. Hentet fra UiA: <https://www.uia.no/studier/emner/2023/host/is-304.html>

University of St. Andrews. (u.d.). *Code style and standards guides*. Hentet fra University of St Andrews: <https://www.st-andrews.ac.uk/digital-standards/code-standards/>

Utdanningsdirektoratet. (u.d.). *Intervjuteknikk for intervju i tilsyn etter barnehagelova og opplæringslova*. Hentet fra Udir: <https://www.udir.no/regelverk-og-tilsyn/intervjuteknikk--for-intervju-i-tilsyn-etter-barnehagelova-og-opplaringslova/5.-ulike-typar-intervju/>

Visual Paradigm. (u.d.). *What is Planning Poker in Agile?* Hentet fra Visual Paradigm: <https://www.visual-paradigm.com/scrum/what-is-agile-planning-poker/>

W3schools. (u.d.). *C# Introduction*. Hentet fra W3schools: https://www.w3schools.com/cs/cs_intro.php

West, D. (u.d.). *Agile scrum roles and responsibilities*. Hentet fra Atlassian: <https://www.atlassian.com/agile/scrum/roles>

West, D. (u.d.). *Sprint planning*. Hentet fra Atlassian: <https://www.atlassian.com/agile/scrum/sprint-planning>

Woke, G. (2023, Mars 24). *The difference between libraries and frameworks*. Hentet fra Redgate: <https://www.red-gate.com/simple-talk/development/other-development/the-difference-between-libraries-and-frameworks/>

Zaleski, M. (u.d.). *Traversing Hybrid Project Management: The Bridge Between Agile and Waterfall*. Hentet fra Project Managers: <https://www.toptal.com/project-managers/agile/hybrid-project-management-a-middle-ground-between-agile-and-waterfall>

Zhao, R. (2024). *Research on Teaching Web Front-End Development*. Hentet fra clausiuspress.com: https://www.clausiuspress.com/assets/default/article/2024/03/26/article_1711508977.pdf

Appendiks

Appendiks 1: Risikomatrixe

Risikoanalyse før tiltak					
Nr.	Usikkerhet	Type	Sannsynlighet	Negative konsekvenser	Status
1	Får ikke kontakt med bedrift	Risiko	Lav	Høy	Betydelig risiko
2	Prosjektet tilfredsstillende ikke behovene til produkteier	Risiko	Lav	Svært lav	Ingen betydelig risiko
3	Lav kodekvalitet	Risiko	Høy	Høy	Høy risiko
4	Et uengasjert gruppemedlem bidrar lite	Risiko	Lav	Lav	Har lite risiko
5	Flere uengasjerte gruppemedlemmer bidrar lite	Risiko	Svært lav	Høy	Har lite risiko
6	Data fra databasen lekker ut	Risiko	Svært lav	Svært høy	Betydelig risiko
7	Klarer ikke å implementere noen funksjoner	Risiko	-	Lav	-
8	Klarer ikke å implementere alle funksjoner	Risiko	-	Svært lav	-
9	Dårlig tidsestimering	Risiko	Lav	Middels	Betydelig risiko
10	Prosjektet blir kansellert midt i semesteret	Risiko	Svært lav	Høy	Har lite risiko
11	Dårlig kommunikasjon	Risiko	Lav	Høy	Betydelig risiko
12	Dårlig optimalisering av kode	Risiko	Middels	Middels	Betydelig risiko
13	Systemet er for ressurs-intensiv	Risiko	Middels	Middels	Betydelig risiko

Risikoanalyse etter tiltak					
Nr.	Usikkerhet	Type	Sannsynlighet	Negative konsekvenser	Status

1	Får ikke kontakt med bedrift	Risiko	Lav	Høy	Betydelig risiko
2	Prosjektet tilfredsstillende ikke behovene til produkteier	Risiko	Lav	Svært lav	Ingen betydelig risiko
3	Lav kodekvalitet	Risiko	Lav	Høy	Betydelig risiko
4	Et uengasjert gruppemedlem bidrar lite	Risiko	Lav	Lav	Har lite risiko
5	Flere uengasjerte gruppemedlemmer bidrar lite	Risiko	Svært lav	Høy	Har lite risiko
6	Data fra databasen lekker ut	Risiko	Svært lav	Svært høy	Betydelig risiko
7	Klarer ikke å implementere noen funksjoner	Risiko	-	Lav	-
8	Klarer ikke å implementere alle funksjoner	Risiko	-	Svært lav	-
9	Dårlig tidsestimering	Risiko	Lav	Middels	Betydelig risiko
10	Prosjektet blir kansellert midt i semesteret	Risiko	Svært lav	Høy	Har lite risiko
11	Dårlig kommunikasjon	Risiko	Lav	Høy	Betydelig risiko
12	Dårlig optimalisering av kode	Risiko	Middels	Middels	Betydelig risiko
13	Systemet er for ressurs-intensiv	Risiko	Middels	Middels	Betydelig risiko

Risikomatrix før tiltak						
Sannsynlighet	Svært høy					
	Høy			12	3	
	Middels			12 13		
	Lav	2	4	9	1 11	
	Svært lav				5 10	6
		Svært lav	Lav	Middels	Høy	Svært høy
Konsekvenser						
Risikomatrix etter tiltak						
Sannsynlighet	Svært høy					
	Høy					
	Middels			12 13		
	Lav	2	4	9	3 11 1	
	Svært lav				5 10	6
		Svært lav	Lav	Middels	Høy	Svært høy
Konsekvenser						
Tiltak						
Grønn	Ingen betydelig risiko					
Lys Grønn	Har lite risiko(vurder tiltak)					
Gul	Betydelig risiko (Risikoreduserende tiltak iverksettes hvis praktisk mulig)					
Lys Rød	Høy risiko(start å iverksette tiltak)					
Rød	Uakseptabelt (Risikoreduserende tiltak er påkrevd)					

Appendiks 2: MoSCoW RG API-er

APIer	Must Have	Should Have	Could Have	Won't have
Ambient Air Temperature Time Series			x	
Battery Charge Cycles Count Time Series				x
Battery Charger Power Line State Time Series				x
Battery Charger State Time Series				x
Battery Current Time Series				x
Battery Discharged Energy Last 24 Hours Time Series				x
Battery Discharged Energy Time Series	x			
Battery Potential Time Series				x
Battery Remaining Charge Time Time Series				x
Battery Remaining Run Time Time Series				x
Battery State of Charge Time Series				x
Battery State of Health Time Series			x	
Battery Temperature Time Series				x
CO2 Emissions Time Series				x
Def Remaining Time Series			x	
Distance Time Series		x		
Engine Coolant Temperature Time Series				x
Engine Speed Time Series				x
Engine Status Time Series				x
Fault Code Time Series			x	
Fault Status Time Series			x	
Fuel Remaining Time Series				x
Fuel Used Last 24 Hours Time Series				x
Fuel Used Time Series	x			
Idle Hours Last 24 Hours Time Series				x
Idle Hours Time Series	x			
Load Count Time Series	x			
Location Adress Time Series	x			
Locations Time Series	x			
Moving Hours Time Series	x			
Operating Hours Last 24 Hours Time Series				x
Operating Hours Time Series	x			
Operating status time series	x			
Payload totals time series	x			
Speed Time Series		x		

MoSCoW	Brukerhistorie	Funksjon	Akseptkriterier	Argumentasjon
Must have #1	Som en miljødirektør i Rental Group vil jeg se gjennom en sortert liste over de maskinene som lager størst utslipp for å vite den beste måten å redusere vårt utslipp på er.	Se gjennom og sortere maskiner etter utslipp.	Liste med sorterings-funksjon som viser maskiner.	Det er svært viktig for å kunne vite det beste steget er for å redusere utslipp, er det viktig å vite de meste forurensede maskinene.
Must have #2	Ansatt har vanskeligheter med å putte dataen fra grafene i perspektiv, han blir litt lengre ned, hvor det står "x antall utslipp tilsvarer å varme opp y antall hus".	Sammenlikne data med gjenstander - putte data i perspektiv.	Sammenlikne vist data med noe bruker kan relatere til.	Å få dataen sammenlignet med objekter brukeren kan relatere til gir brukeren en dypere forståelse av tallene. Dette er essensielt etter som at tall har ingen verdi hvis det ikke blir lagt i perspektiv.
Must have #3	For å få innsikt i hvor mange maskiner er på, men ikke i arbeid, ønsker jeg som miljødirektør å få statistikk på maskinenes tomgang.	Velge maskin(er) og få en fremvisning av statistikken i grafer.	Visning av maskiner, og sorterings-muligheter etter utslipp.	Det er viktig for å kunne se statistikken på tomgang av maskin(er). Dette er essensielt for å kunne minske tomgangstimer ved RGs kundebedriftene.
Must have #4	Som selger/drifter ønsker jeg å få utslippsmengde og detaljer om en maskin.	Finne ønsket data om maskiner og videreformidle det til kundene.	Få liste over utvalgte maskiner og dens data.	For å gi kunder med up to date informasjon om de aktuelle maskinene er det essensielt å vise dem innblikk i maskinenes data.
Must have #5	Som ansatt vil jeg se hvor stor utslippsmengde i et visst tidsrom.	Velge tidsrom og få dataen framvist i en graf.	Visning av data ifra satt dato/tid.	For å kartlegge utslipp i spesifikke perioder, er det essensielt å kunne velge og se hvilke tider maskinene ble brukt.
Must have #6	Som miljødirektør ønsker jeg å se hvor effektive maskinene våre er i forhold til utslipp.	Viser en oversikt over hvor mange lass maskinene fullfører delt på utslipp.	Viser frem data i en tabell.	Dette er essensielt for å se hvor effektive maskinene opererer for å kunne gi anbefalinger på hvilke maskiner som er mest effektive, samt slipper ut mindre utslipp.
Should have #7	Som sjef for innkjøp vil jeg se en liste over dumpere med tanke på utslipp, hvor mye last de kan flytte og ta en beslutning på hva vi burde kjøpe inn.	Sammenlikne dumpere, basert på masse fraktet per liter/kwh drivstoff/ elektrisitet.	Sammenlikne ulike dumpere (sammenligne maskintype, maskinmodell).	Dette kan være nyttig for å vite hvilken dumper en bør kjøpe inn flere av så det kan være lurt å finne ut hvilken er mest effektiv og hvilken er mest miljøvennlig.
Should have #8	For å være et sted foran konkurrentene ønsker jeg som markedskoordinatør at applikasjonen viser frem statistikk som viser hvor mye arbeid som blir gjennomført per liter drivstoff.	En del i applikasjon som viser effektivitet per liter drivstoff.	Visning av statistikk per liter drivstoff.	For å være den beste bedriften i markedet må man holde seg oppdatert på konkurrenter og passe på at man er et steg foran. Dette kan gjøres ved å få statistikk på effektivitet per utslipp, og deretter sammenligne det med effektiviteten til konkurrenter.

Should have #9	For å kunne se hvor mye av den tillatte totalvekten er tatt i bruk hver kjøretur ønsker jeg som drifter å sammenligne vekten av utkjørte last og maskinens tillatte totalvekt, gjerne i tonn, prosent og per modell gitt av en maskin.	Vise frem ønsket data.	Visning av hvor effektivt maskinen(e) er brukt.	For å kunne undersøke hvilke maskiner er effektive og blir brukt på den mest effektive måten, kan det være gunstig å se på lasteffektiviteten på maskinen.
Should have #10	For å kunne bruke statistikk fra webapplikasjonen ønsker jeg som manager å kunne laste ned de forskjellige grafene som finnes i systemet.	Nedlasting av grafer.	Knapp for å laste ned graf.	Det kan være lurt å vise fram dataen utenfor webapplikasjonen.
Could have #11	Som miljødirektør ønsker jeg å ha oversikt over den maksimale kapasiteten til de forskjellige maskinene våre.	Vise en liste over maskinene RG eier og maks tillatt kapasitet per maskin mål i tonn.	Viser en liste med maskinene sortert fra høyest til minst tillatt totalvekt.	Dette øker effektivitet og produktivitet ved å tildele maskiner som kan transportere tilsvarende vekt. Dette kan være lurt å ha med siden det kan minke situasjoner hvor store maskiner frakter lite vekt. Dette fører til mindre CO2-utslipp.
Won't have #12	Som styremedlem ønsker jeg å se CO2 utslipp fra de ulike maskinene i et utvalgt byggeplass.	Vise CO2-utslipp ved valg av byggeplass.	Visning av kart for byggeplasser og fremvise CO2-utslipp og maskiner.	RG ville heller fokusere på selve kundene, prosjektene og maskinene, derfor trengs det ikke kartvisning for byggeplasser.

Appendiks 5: Prosjektdagbok

<https://drive.google.com/file/d/1iPsR6aj3J0oL-wWUM4G5BTjE4JKW75t/view?usp=sharing>

Appendiks 6: Møtereferat Styringsmøte 1

Til stede:

Produkteier: Vegard Sætre

Veileder: Rania Fahim Hassan Ibrahim Elgazzar

Scrum master: Dominykas Zuoza

Student: Eirik Skogland Nedland

Student: Eziz Ømer

Student: Bastian Ege Tveit

Student: Dung Katrine My Huynh

Hva som ble godt igjennom på møtet:

Legge til i Risikoanalyse

- Feiltidsestimering på å lære nye ting

Waterfall/Agile

- Waterfall med agile-elementer
- Et prosjekt er aldri 100% agil eller waterfall. Så selv om vi planla som waterfall, så ble det mye fram og tilbake mtp tilbakemeldinger, møter med RG/Capgemini, deretter gå tilbake på analyse, design og prototype.

Mer kontakt med Vegard

- Gi beskjed om vi tar hjemmekontor i god tid, slik at han kan avbestille kontoret
- Mer oppdateringer
- Skriv mail med oppdateringer de ukene vi ikke har sprint review med han
- Oppsummering sendt til Vegard fredager uten sprint review
- Kan plage han så mye vi vil

Forbedring av presentasjonsteknikk

- Har alt samlet på en presentasjon, ikke hoppe mellom ulike presentasjoner
- Utklipp for backlogg
- Forklare mer om utfordringer

Tidsestimering

- Timeliste per person 25 timer per person – 150 timer i uka totalt

Mvh. Gruppe 1

Appendiks 7: Møtereferat Styringsmøte 2

Dato: 11/04/2024

Personer til stede:

- Rania Elgazzar (UiA veilder)
- Vegard Sætre (Capgemini veilder)

- Eziz Ømer
- Eirik Skogland Nedland
- Dominykas Zuoza
- Dung Katrine My Huynh
- Bastian Ege Tveit
- Eivind Hustad Fiske

Hva som har blitt presentert/sagt av gruppen:

Styringsmøte 2

- Risikoanalyse endringer
- Kodekvalitet, ressurs krevende,

- Hvordan vi setter opp Bug Tasks pga. dårlig utfyllende oppgaver
- Avvikling fra design til faktisk produkt
- Tidsrom funksjon

- Egen refleksjon på mangel av unit testing
- Kodenivå

- Alle skriver kode på sin måte, utfordringer med å standardisere hvordan kode skal se ut når det er pushet ut.
 - Sikkerhetsårsaker

- Fant noen sikkerhetshull og lo inn tiltak for å fikse de. Dette gjelder spesielt XSS og SQLi
 - Om vi har blitt ferdig med alle funksjoner på MoSCoW

- Ble ferdig med Must Haves og Should Haves

Tilbakemelding fra veileder(e) og diskusjon:

- Godt at vi jobber med reelle data.
- Har tiltakene på usikkerhetene i risikoanalysen bidratt til hjelp?
- Siden dette egentlig er en PoC, så kunne vi ha mikset og trikset på data og statistikk slik at vi kunne ha presentert hvordan applikasjon så ut i den idelle verden.
- Hvorfor ble vi ikke ferdig med å implementere alle funksjoner?

- Etter å ha hatt diskusjon med oppdragsgiver, ble det besluttet at tiltenkte funksjoner ikke var nødvendig. De så rett og slett ikke bruksområdet for funksjonene som vi senere ikke tok med. Dette er da Could Haves og Won't Haves

Appendiks 8: Møtereferat Styringsmøte 3

Dato: 02/05/2024

Personer til stede:

- Rania Elgazzar (UiA veileder)
- Fraværende: Vegard Sætre (Capgemini veileder)

- Eziz Ømer
- Eirik Skogland Nedland
- Dominykas Zuoza
- Dung Katrine My Huynh
- Bastian Ege Tveit
- Eivind Hustad Fiske

Hva som har blitt presentert/sagt av gruppen:

- Townhall presentasjon
- Rapportskrivning

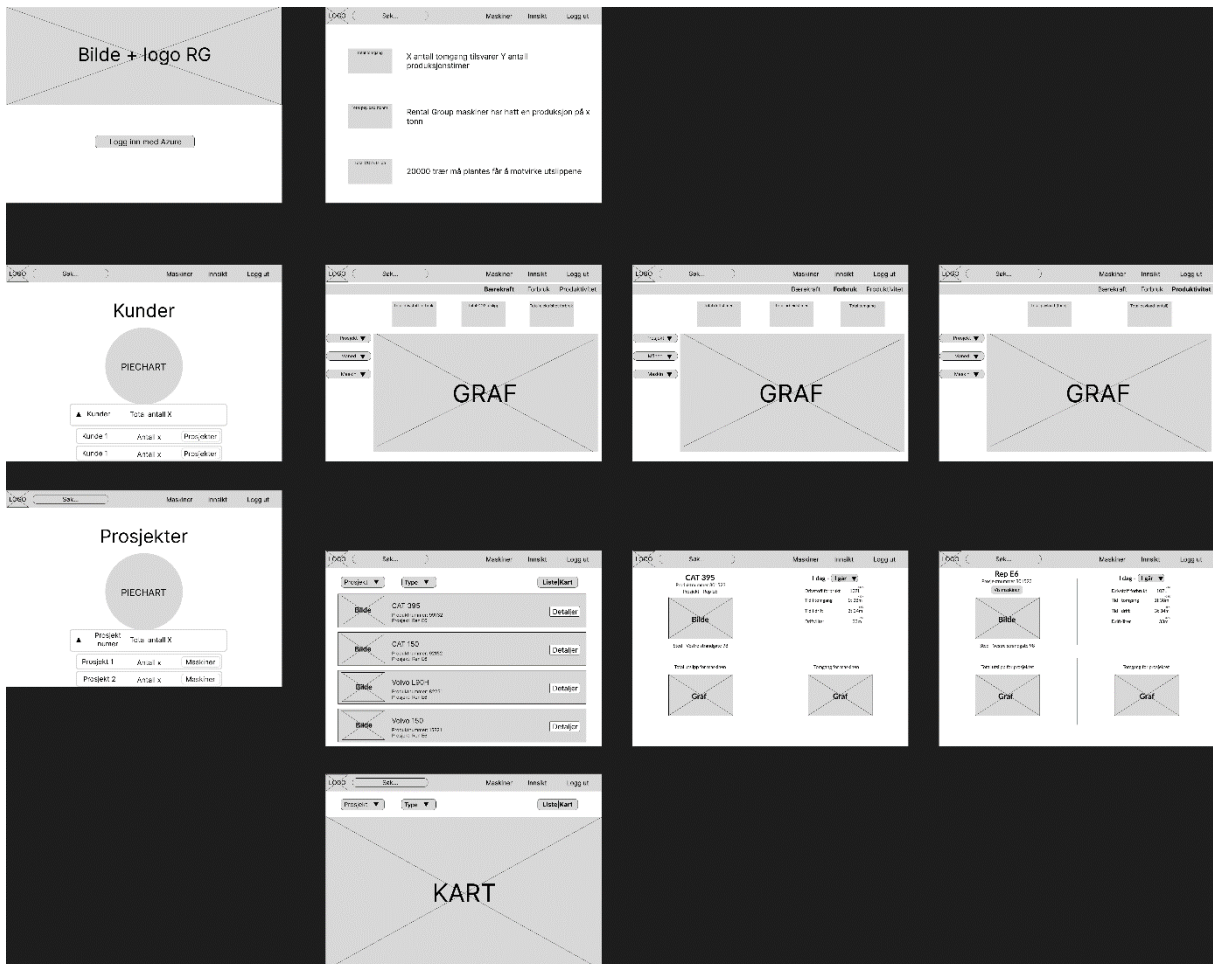
- Skal ferdigstille første utkast, deretter gå gjennom teksten sammen, videre sende det til Rania

- Rania gir tilbakemelding 14. uke (uke 19)
- Gruppen skal ferdigstille rapport i god forkant før innlevering den **16.Mai 2024 kl23:59**
- Expo (mer info på tilsendt powerpoint)

Tilbakemelding fra veileder(e) og diskusjon:

- Refleksjon og konklusjon går sammen
- Kvalitetssikring bør være fordelt på kapitler istedenfor et eget
- Planlegging & analyse: Risikomatrixe
- Design: Brukertestig
- Utvikling: Kodetesting, applikasjons-brukertesting, demo, Qodana, code reviews og kodestandarder

Appendiks 9: Endelig wireframe



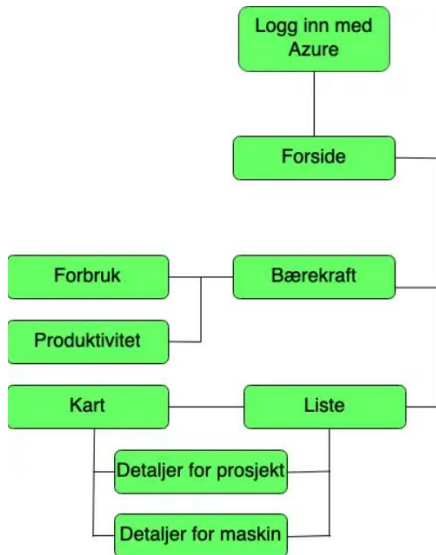
Appendiks 10: Link til Prototype

Alle med link skal kunne se prototypen

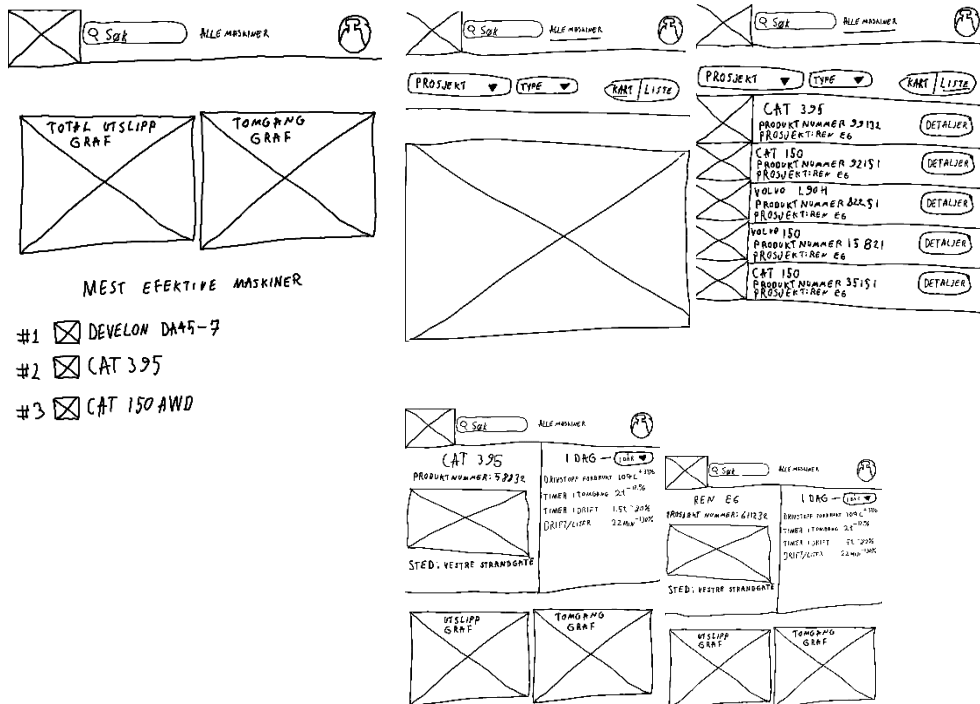
Link til prototype:

<https://www.figma.com/proto/NuJuDXrOhtwEn6Vpg1VM5s/Endelig-Wireframe?node-id=424-1820&starting-point-node-id=424%3A1820&mode=design&t=d6W8COnrErWZnQRr-1>

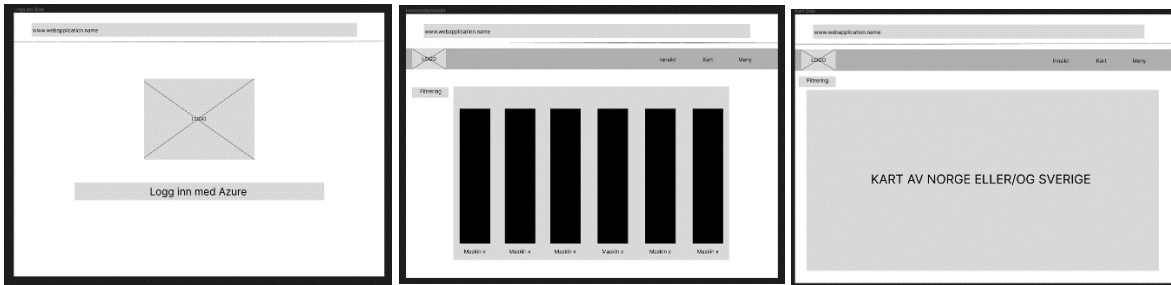
Appendiks 11: Navigasjonskart første utkast



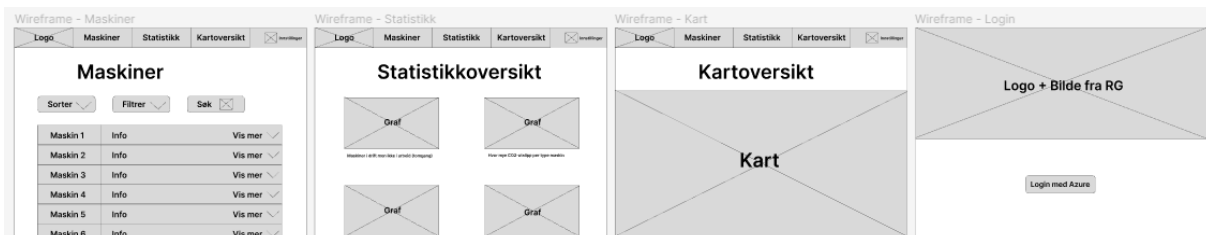
Appendiks 12: Individuell wireframe Dominykas



Appendiks 13: Individuell wireframe Eziz



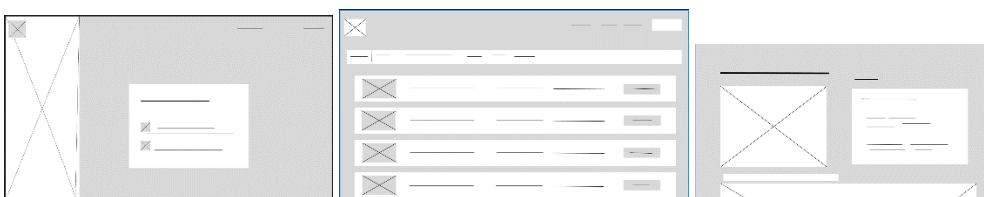
Appendiks 14: Individuell wireframe Katrine



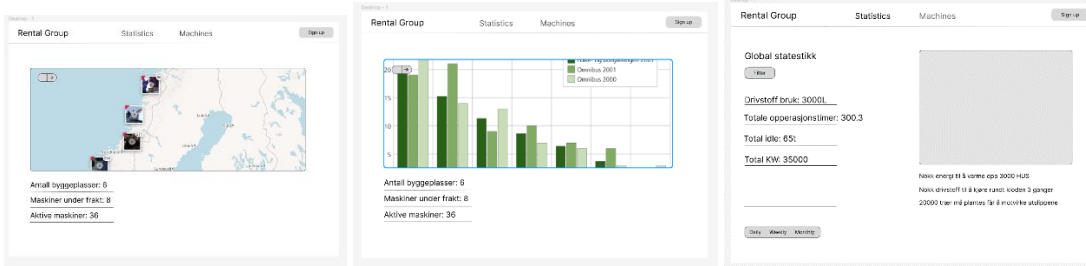
Appendiks 15: Individuell wireframe Eirik



Appendiks 16: Individuell wireframe Bastian



Appendiks 17: Individuell wireframe Eivind

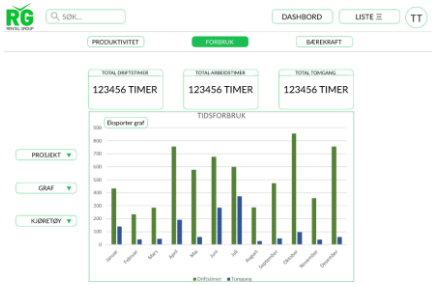
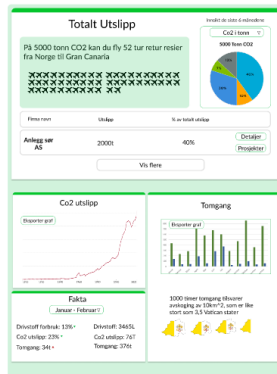


Gravemaskin 10t7v	Vi summet i Norge	Operasjonstimer
Kapasitet: 207	185.0	187
Faktor: 185.0		
Leisjon: 185.0		

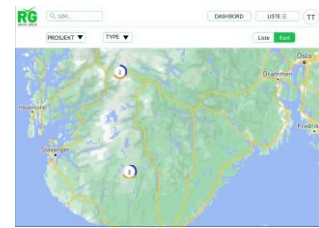
Appendiks 18: Mockup



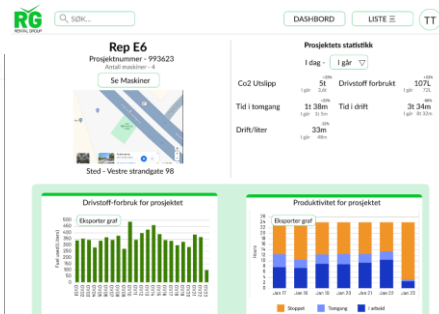
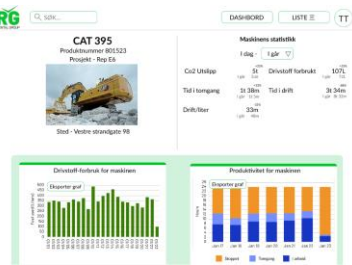
Vennligst logg in med Azure
 Azure Active Directory B2C



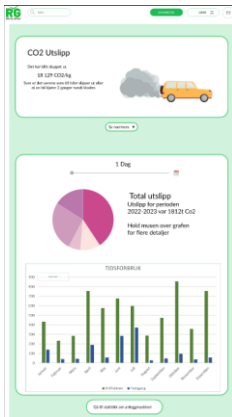
Enhet	Aktivitet	Utnyttelse	DETALJER
CAT 772G	Produksjonsnr: ABC124 Prosjekt: Tunnel-prosjekt-12	100%	DETALJER
CAT D8	Produksjonsnr: CBA987 Prosjekt: Tunnel-prosjekt-12	100%	DETALJER
AMV 3ZXC-CC	Produksjonsnr: RAC789 Prosjekt: Tunnel-prosjekt-12	100%	DETALJER
VOLVO JEC230EL	Produksjonsnr: DQZ230EL Prosjekt: Tunnel-prosjekt-12	100%	DETALJER



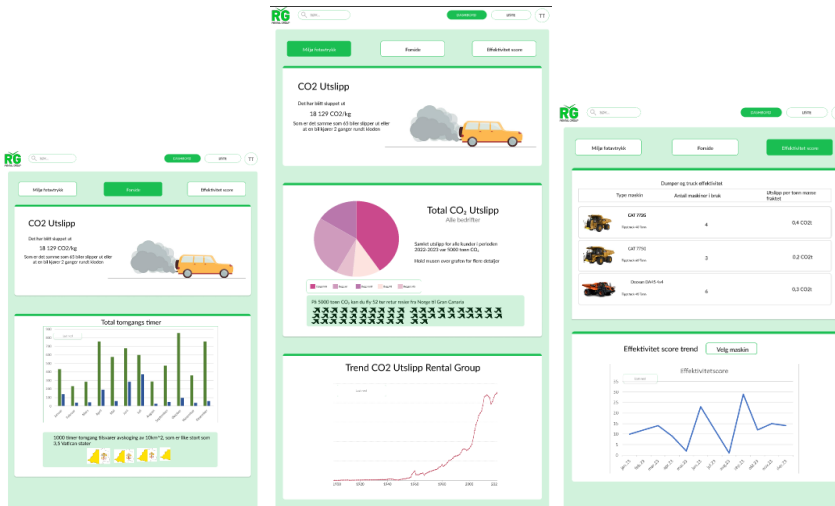
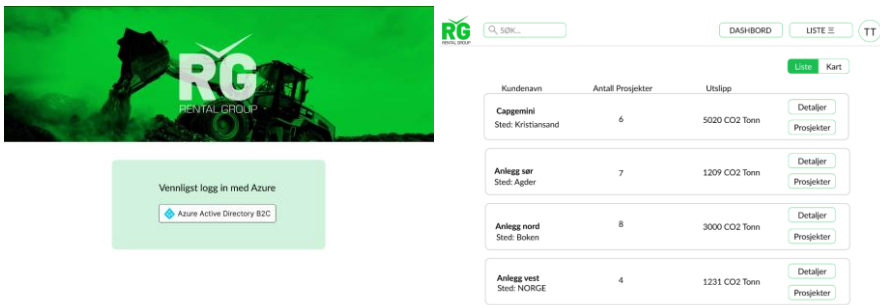
PROSJEKT	TYPE	Antall Maskiner	Utslipp	DETALJER
Tunnel-prosjekt-12	Sted: Tolmark	6	5020 CO2 Tonn	DETALJER
Rep-prosjekt-13	Sted: Agder	7	1209 CO2 Tonn	DETALJER
BOKNAFJORD-TUNNEL	Sted: Bohlen	8	3000 CO2 Tonn	DETALJER
EXP-EXPANSJON	Sted: HORDALE	4	1231 CO2 Tonn	DETALJER



Alternativ dashboard side



Appendiks 19: Prototype i Figma



RG DASHBOARD LISTE KUNDE PROSJEKTER MASKINER

PROSJEKT Kjøretøy Oppdatter sak

Prosjekt	Antall Maskiner	Utslipp
Tunnel-prosjekt-12 Sted: Telemark	6	5020 CO2 Tonn
Bro-prosjekt-13 Sted: Agder	7	1209 CO2 Tonn
BOKNAFJORD-TUNNEL Sted: Boken	8	3000 CO2 Tonn
E39-EXPANSION Sted: NORGE	4	1231 CO2 Tonn

RG DASHBOARD LISTE TT

PROSJEKT Kjøretøy Oppdatter sak

Enhet	Aktivitet	Utnyttelse
CAT 395 Produktnummer: ABC134 Prosjekt: Tunnel-prosjekt-12	Stoppet	100% 40/100
CAT D8 Produktnummer: CB4987 Prosjekt: Tunnel-prosjekt-12	Stoppet	100% 40/100
AMV 3ZXC-CC Produktnummer: Kw6789 Prosjekt: Tunnel-prosjekt-12	I arbeid	100% 40/100
VOLVO JEC230EL Produktnummer: DIG230EL Prosjekt: Tunnel-prosjekt-12	I arbeid	100% 40/100

RG DASHBOARD LISTE TT

Totalt Utslipp

På 5000 tonn CO2 kan du fly 52 tur retur reser fra Norge til Gran Canaria

Firma navn Utslipp % av totalt utslipp

Anlegg ser AS 2000t 40%

Co2 utslipp Tomgang

Fakta
Januar - Februar

Drivstoff forbruk: 13%
Co2 utslipp: 23%
Tomgang: 34%

Drivstoff forbruk: 107L
Tid i drift: 3t 34m
Drift/liter: 33m

1000 timer tomgang shower andring av 300m2, som er like stort som 3,5 Volvoen stator

RG DASHBOARD LISTE TT

Cappgemini - Rep E6

Prosjektnummer - 993623

Prosjektets statistikk

Co2 Utslipp 5t Drivstoff forbrukt 107L

Tid i tomgang 11 38m Tid i drift 3t 34m

Drift/liter 33m

Sted - Vestre strandgate 98

Drivstoff-forbruk for prosjektet

Produktivitet for prosjektet

RG DASHBOARD LISTE TT

Rep E6

Prosjektets statistikk

CAT 395 Produktnummer: ABC134 Prosjekt: Tunnel-prosjekt-12	Stoppet	100% 40/100
CAT D8 Produktnummer: CB4987 Prosjekt: Tunnel-prosjekt-12	Stoppet	100% 40/100
AMV 3ZXC-CC Produktnummer: Kw6789 Prosjekt: Tunnel-prosjekt-12	I arbeid	100% 40/100
VOLVO JEC230EL Produktnummer: DIG230EL Prosjekt: Tunnel-prosjekt-12	I arbeid	100% 40/100

RG DASHBOARD LISTE TT

Cappgemini

Kundenummer - 987654

Antall prosjekter - 4

Kundens statistikk

Co2 Utslipp 5t Drivstoff forbrukt 107L

Tid i tomgang 11 38m Tid i drift 3t 34m

Drift/liter 33m

Sted - Vestre strandgate 98

Drivstoff-forbruk for kunden

Produktivitet for kundens prosjekter

RG DASHBOARD LISTE TT

Cappgemini

Kundens statistikk

Prosjekt	Antall Maskiner	Utslipp
Tunnel-prosjekt-12 Sted: Telemark	6	5020 CO2 Tonn
Bro-prosjekt-13 Sted: Agder	7	1209 CO2 Tonn
BOKNAFJORD-TUNNEL Sted: Boken	8	3000 CO2 Tonn

Appendiks 20: Intervjuer med Cappgemini

Intervju med Konsulent 1

Om intervjuobjektet:	
Kort introduksjon av deg	Konsulent innad software engineering, C & CA
Dine arbeidsoppgaver	Utvikler
Digitalisering og påvirkning:	
Hva legger du i ordet digitalisering? (vil vite dette for å danne fundament på hva som settes som betydning av digitalisering i intervjuet)	Digitalisering, gå fra analogt til digitalt. Det er noe som blir viktigere og viktigere.
Hva gjøre Capgemini for å digitalisere andre bedrifter som et konsulentselskap? (Capgemini sin påvirkning på andre bedrifter)	<ul style="list-style-type: none"> - Lage nye systemer, endre systemer - Rådgivning for strategier, planer - Rådgivning for digitale løsninger
Hva slags systemer pleier Capgemini å lage? Er det CRM/ERP-systemer? (dette går sammen med spørsmålet over og under, for å få en diskusjon rundt påvirkning og produkter)	CRM-systemer ERP-systemer
Hvordan klarer bedriftene å tilpasse seg endring og implementering av nye løsninger?	Det kommer an på om bedriften er åpen for endring.
Capgemini tiltak:	
Når det gjelder digitalisering, hva er de mest avgjørende faktorene for vellykket digitalisering?	Endringsvillighet fra kunden og brukergruppen.
Hva er noen av de vanligste utfordringene dere møter underveis i en digitaliseringsprosess, og hvordan håndterer dere dem?	<ul style="list-style-type: none"> - Tilpasningsdyktighet - Endringsvillighet Eks: Leder kan like systemet og ønsker å ta det i bruk, men brukergruppen liker ikke å ta i bruk systemet.
Hvordan identifiserer dere potensielle digitaliseringsområder i en bedrift?	Er som ikke regel med i denne prosessen. <ul style="list-style-type: none"> - Analyse for prosesser innad kundebedriften blir gjennomført

Har dere noen eksempler fra prosjekter der deres digitaliseringsarbeid ga stor nytte til kunden?	<ul style="list-style-type: none"> - Finne kundens behov og ønsker basert på dette
Hvordan måler dere suksessen til deres digitaliseringsinitiativer?	<ul style="list-style-type: none"> - Måles med bedriftene - Måles økonomisk, tid spart og påvirkning av løsningen
Hvordan sikrer dere at løsningene er skalerbare og bærekraftige i sikt?	<ul style="list-style-type: none"> - Det menneskelige aspektet med bærekraft, er de villig til å ta i bruk løsningen. Å ta i bruk et nytt system tar tid. - Det tekniske aspektet med bærekraft, nye løsninger som har mindre forbruk.
Hvordan balanserer dere behovet for innovasjon og sikkerhet under en digitaliseringsprosess?	Innovasjon og sikkerhet går innad hverandre <ul style="list-style-type: none"> - Mer innovasjon kan lede til mer sikkerhet

Viktigste temaer:
Bærekraft

Intervju med Konsulent 2

Om intervjuobjektet:		Når det gjelder digitalisering, hva er de mest avgjørende faktorene for vellykket digitalisering?	Det som er laget, fungerer Gir verdi for kunden - Det er ikke bra å digitalisere bare for å digitalisere - Hvis penn og papir funker best hvorfor endre det <u>Unngå feature creep og tekno creep</u> Bruker kun de verktøy som er nødvendig
Kort introduksjon av deg	Programvareutvikling som konsulent		
Dine arbeidsoppgaver	Utvikle webapplikasjoner	Hva er noen av de vanligste utfordringene dere møter underveis i en digitaliseringsprosess, og hvordan håndterer dere dem?	Vanskelig for kunden å forstå hva som blir gjort Kan tro at deres ønske er et bra ønske, men vi kan tro at det er et dårlig ønske Teamet har ikke nok kompetanse for å løse dette ønsket, dermed hentes inn andre konsulenter fra huset med kompetanse innen dette
Digitalisering og påvirkning:		Hvordan identifiserer dere mulige digitaliseringsområder i en bedrift? Har dere noen eksempler fra prosjekter der deres digitaliseringsarbeid ga stor nytte til kunden?	Business Technology <u>solutions</u> Business <u>experience</u> De som ser på mulighetene for forbedring De som jobber med salg, har en business plan Ser hvordan Capgemini kan hjelpe kunden med å øke effektivitet og få økonomisk gevinst
Hva legger du i ordet digitalisering? (vil vite dette for å danne fundament på hva som settes som betydning av digitalisering i intervjuet)	Gjøre manuelt arbeid automatisk med digitale verktøyer Utføre oppgavene dine og lagre data på en digital måte	Hvordan måler dere suksessen til deres digitaliseringsinitiativer?	Har levert noe med kvalitet Ikke krasjer Godkjent med tanke på Accessibility Gir verdi Kunden bruker det daglig og det hjelper dem Såpass fornøyd at de vil videreutvikle det, Great <u>Success</u> Begynt å tjene mer penger (kunden) og trenger mer funksjonalitet (Capgemini happy)
Hva gjøre Capgemini for å digitalisere andre bedrifter som et konsultentselskap? (Capgemini sin påvirkning på andre bedrifter)	Dekker kundens behov Kundene vil ikke være irrelevant når teknologien utvikler seg Effektiviserer seg ved hjelp av digitalisering, vil skape økonomisk gevinst for dem	Hvordan sikrer dere at løsningene er skalerbare og bærekraftige i sikt?	<u>Cloud</u> , bra <u>cloud</u> kompetanse, utvikle og <u>deploye på cloud</u> er mer bærekraftig og <u>letter</u> å skalere opp og ned. Har eksperter på bærekraft og gir råd om bærekraft til kundene.
Hva slags systemer pleier Capgemini å lage? Er det CRM/ERP-systemer? (dette går sammen med spørsmålet over og under, for å få en diskusjon rundt påvirkning og produkter)	Alt, kan være ting som eksisterer, men ikke fungerer bra, oppdatering av gamle systemer, alt fra en enkel nettside til ai applikasjon, mobil applikasjon, forstå dataen de har samlet Bedrift som spør om oppdatering: folk er mot endringer, de gamle systemene begynner å bli gamle, sikkerhetsrisiko med gamle systemer, flytte ting til <u>cloud</u> Gir mening å ha disse oppdateringene, når dette gjøres fortsetter de å bruke det gamle mens det nye blir utviklet. Sakte overgang fra gamle til nye	Hvordan balanserer dere behovet for innovasjon og sikkerhet under en digitaliseringsprosess?	Sikkerhet fra dag en Det er ikke noe løsning så skal du fortsatt ha best sikkerhet fra dag en Første dag måte ha en API-nøkkel, tidlig ingen ting kan stjeles enda, men likevel sende team arbeider med <u>win</u> fil så <u>Api</u> nøkkel ikke kan leses og slettet melding etterpå. Sikkerhet er viktigst fra dag 1
Hvordan klarer bedriftene å tilpasse seg endring og implementering av nye løsninger?			
Capgeminis tiltak:			

Viktigste temaer:

- Digitalisere ting
 - Ting blir mer effektivt og enkelt
- Måle suksess med hvor mye verdi kunden får
- Må digitalisere seg for å ikke bli irrelevant
- Digitalisering vil ikke fikse alt, og kan skape andre problemer, men gjør ting lettere
 - Kult at du kan sette på en film med en gang og se på den i 4k

Eksempel prosjekt: Kunde trenger merkevare nettside, alt fra bunnen. Vet ikke helt hva de skal ha, men har ambisjon, enig med noen av våres ideer, jobber sammen med å finne løsninger, også utviklere har denne diskusjonen med kunde

Aldri hørt om dette i Capgemini, men kunden kan si at de ikke trengte dette likevel

Nøtt til å levere noe bra, men til slutt så er det kunden som skal ta dette i bruk

Intervju med Sikkerhetsspesialist		Cappgemini tiltak:	
Om intervjuobjektet:			
Kort introduksjon av deg	Cybersikkerhet rettet mot mennesker i sentrum, ikke nødvendigvis automatisering. Mer om <u>awareness</u> , <u>phishing</u> Finne sikkerhetshull, manglende <u>patching</u> Fikse koden Cyber- <u>selfassessment</u> Se sikkerhetsstatus innen teamet Security champion Gir sikkerhetskurs Være litt Tech support for dem Årlig test om status på sikkerhet Jobbe med å få sikkerhet inn i team	Når det gjelder digitalisering, hva er de mest avgjørende faktorene for vellykket digitalisering?	- Hvis man skal digitalisere, må man ha en god forståelse - Det bygges et godt fundament på dette - Finne ut av kundenes behov
Dine arbeidsoppgaver		Hva er noen av de vanligste utfordringene dere møter underveis i en digitaliseringsprosess, og hvordan håndterer dere dem?	- Vi tar over utviklingen av systemet - De gamle systemer har egne sikkerhetstiltak - Vanskelig å endre gammel kode og endre sikkerhet på systemet - Kan svare nei til å utvikle systemet grunnet økonomi, og kunden vil bare ha et produkt
Digitalisering og påvirkning:		Hvordan identifiserer dere mulige digitaliseringsområder (sikkerhet) i en bedrift? Har dere noen eksempler fra prosjekter der deres digitaliseringsarbeid ga stor nytte til kunden?	- Setter oss ned med teamet som skal jobbe med teamet - Cyber- <u>assessment</u> - Status på bedriften - Få informasjon av kunden for å få overblikk av situasjonen - Hjelpe kunden finne oversikten, rådgiving - Lage oppgaver basert på hva kunden trenger - Finne løsninger til dette - Øke sikkerhet uten at det påvirker produktlevering
Hva legger du i ordet digitalisering? (vil vite dette for å danne fundament på hva som settes som betydning av digitalisering i intervjuet)	Gå fra papir samfunn til at alt skal gjøres på pc. Systemer som jobber sammen Ting automatiserer seg mye på nettet. Hva man kan gjøre for å automatisere prosesser for mennesker. Automatiserte prosesser overvåket av mennesker. Mer kontroll av dataen.	Hvordan måler dere suksessen til deres digitaliseringsinitiativer?	- Se på tidligere tilstander - Penger, effektivitet, belastning - Hva er tilstanden før og etter løsningen - Se forbedringsområder - God endringer - Negative endringer - Reflektere
Hva gjøre Cappgemini for å digitalisere andre bedrifter som et konsultentselskap? (Cappgemini sin påvirkning på andre bedrifter)	Sitter ikke høyt nok oppe for å svare på dette.	Hvordan sikrer dere at løsningene er skalerbare og bærekraftige i sikt? (sikkerhet og bærekraft)	- Sikkerhet krever tid og ressurser - Ser ikke gevinst når det blir gjennomført - Hvis man blir hacket, hvor lang tid tar å komme tilbake - Ser ikke verdi av det, så lenge det fungerer - Prosesser og prosjekter fungerer, da tenker man ikke bærekraft, sikkerhet
Hva slags systemer pleier Cappgemini å lage? Er det CRM/ERP-systemer? (dette går sammen med spørsmålet over og under, for å få en diskusjon rundt påvirkning og produkter)	ERP er nok den som jeg kommer mest innom. ERP sitter jeg mest med.	Hvordan balanserer dere behovet for innovasjon og sikkerhet under en digitaliseringsprosess?	- Mye snakk med kunden - Hva er det vi ser for oss på dette, hva er det kunden ser for seg - Hva er risikoen for dette når det kommer til sikkerhet <u>Fokuserer på det som er mest kritisk, bruker mer ressurser og tid på dette.</u> - Diskutere de viktigste punktene med kunden
Hvordan klarer bedriftene å tilpasse seg endring og implementering av nye løsninger?	Sitter med sikkerhetsdel på implementering av løsninger.		

Viktigste temaer:

- Forståelse av sikkerhet
- Gjerne opp i systemet
- Sikkerhet er en del av et produkt
- Sikkerhet går i alt (utvikling)
- Sikkerhet i høyere stillinger

Interessekonflikt med kunden, når det kommer til sikkerhet

- Sikkerhet blir nedprioritert
- Forbedring, med at sikkerhet skal være en del av produktet
- Konflikt mellom digitalisering og sikkerhet: kunden velger om de vil ta risikoen med dette

- Fraskriver sikkerhetsansvar
- Opplever egentlig at det ikke er et problem
- Diskusjon om sikkerhetsnivå

Viktigste faktorer som holder systemet sikkert

- Sette god prosess for sikkerhet
- Jo tidligere sikkerhet er implementert, jo bedre
- Involvere kunden, med sikkerhet

Utfordring

- Tid
- Ressurs

Er det stor forskjell mellom kunder som trenger mer sikkerhet (offentlig sektor) og kunder som trenger mindre sikkerhet (mindre bedrifter)

- Andre prioritering
- Små bedrifter (blir løst av seg selv)
- Sikkerhet er mye lettere å jobbe med kunder som tenker på sikkerhet

Intervju med Forretningsutvikler 1

Om intervjuobjektet:			
Kort introduksjon av deg	Forretningsutvikler Business manager Salg eller marketunit		
Dine arbeidsoppgaver	Bygge opp de lokale kundene i Krs Jobbe med å åpne dører Komme inn i nye kundebedrifter i Krs (lokale kunder) - Også går lokalt Jobber med markedsaktiviteter, frokostseminar etc. - Relasjoner, relasjonsbygging, bruk av nettverk - Har stort nettverk for å komme i dialog med nye bedrifter – gjør det lettere å komme i kontakt i nye bedrifter		
Digitalisering og påvirkning:		Capgemini tiltak:	
Hva legger du i ordet digitalisering? (vil vite dette for å danne fundament på hva som settes som betydning av digitalisering i intervjuet)	Digitalisering - Grunnleggende faktor i hva Capgemini gjør - Ligger som et fundament - Ligger som en rød tråd gjennom bedriften, Capgemini - Innsikt i data, kunne gjøre analyser i data	Hva er noen av de vanligste utfordringene dere møter underveis i en digitaliseringsprosess, og hvordan håndterer dere dem?	Må ha med selv folket De må forstå hva som skjer, få forståelse hvordan systemet bør tas i bruk - Mange feiler pga. man får ikke med seg folket - Selv med implementering av nytt system, kan det hende av folk bruker den gamle måten å gjøre prosesser
Hva gjøre Capgemini for å digitalisere andre bedrifter som et konsultentselskap? (Capgemini sin påvirkning på andre bedrifter)	Innen flere områder - Rådgivning, forstår forretningen av sektorer - Strategisk rådgivning - Teknisk kompetanse som bistår bedriftene - Lokal, nasjonal og global kompetanse for bedrifter Eks. Bank, prosessindustri, OT, digitalisere produksjonslinje	Hvordan identifiserer dere potensielle digitaliseringsområder i en bedrift? Har dere noen eksempler fra prosjekter der deres digitaliseringsarbeid ga stor nytte til kunden?	Kan komme med spesifikke behov og så får de forslag for å dekke disse behovene. Andre tilfeller sitter man med kunden og kjenner kunden godt, og da kan komme med innspill.
Hva slags systemer pleier Capgemini å lage? Er det CRM/ERP-systemer? (dette går sammen med spørsmålet over og under, for å få en diskusjon rundt påvirkning og produkter)	Skreddersyr applikasjoner/systemer for bedrifter - Mobilapp - ERP, CRM - Kompetanse innen eksisterende systemer, Dynamics	Hvordan måler dere suksessen til deres digitaliseringsinitiativer?	Kunden får verdi av digitaliseringsløsningen.
Hvordan klarer bedriftene å tilpasse seg endring og implementering av nye løsninger?	Bygge intern kompetanse til å drifte den løsningen. - Bygger kompetanse selv internt - Capgemini tilbyr spisskompetanse for hjelp på bruk av systemet	Hvordan sikrer dere at løsningene er skalerbare og bærekraftige i sikt?	Har gode rammeverk. For å gjøre løsningene skalerbare og bærekraftige følger man disse.
		Hvordan balanserer dere behovet for innovasjon og sikkerhet under en digitaliseringsprosess?	Sikkerhet legges som et fundament av hva enn man gjør. Bygge løsningen på det fundamentet. Løse behovene, og kan få bygge på innovasjon fra dette.
		<p>Viktigste temaer:</p> <p>Må ha med deg menneskene i digitaliseringsprosessen. Du kan verdens beste løsning, men hvis ikke menneskene ikke tar det i bruk, er det ikke noen suksess der.</p> <p>Hvordan få vellykket digitalisering i bedrifter?</p> <ul style="list-style-type: none"> • Involverer brukergruppen • Gi dem forståelse for hvorfor man tar i bruk systemet 	

Intervju med Forretningsutvikler 2 og Statistikk konsulent		Hvordan klarer bedriftene å tilpasse seg endring og implementering av nye løsninger?	Passer på at prosjekter er bra koordinert med hva kunden prøver å løse og at implementeringsmåte passer bra for kunden Ser på en tidslinje for kunden, med budsjett, strategi Treffe riktig problemstillingene, se på kravene, se på stakeholderen UAT – user acceptance test
Om intervjuobjektet:			
Kort introduksjon av deg	6 år i Capgemini - Statistikk konsulent, flettingsanalytiker - Forretningsutvikling - Se hvilke <u>offerings</u> vi har, og hva vi kan tilby - <u>Petroleum</u> -ingeniør, IT-ledelse		
Dine arbeidsoppgaver	Forretningsutvikling		
Digitalisering og påvirkning:			
Hva legger du i ordet digitalisering? (vil vite dette for å danne fundament på hva som settes som betydning av digitalisering i intervjuet)	Betyr lite Digitalisering blir ganske vanlig, er ikke bare fra papir til digitalt format - <u>Bottom</u> up digitalt - Er en nødvendighet nå Eks. Prosessforbedring med digitalisering, er nødvendig nå	Hva er noen av de vanligste utfordringene dere møter underveis i en digitaliseringsprosess, og hvordan håndterer dere dem?	Digitalisering adresserer de rette problemene og dekker kundenes og brukergruppens behov. Tenke: Hva er den beste måten å løse problemet. Må ikke være det beste eller ha se så bra ut. Kostnad spiller også en rolle. - Ikke løser riktige problemer, på en bra måte - Mangel på forståelse av brukeren/prosjektet - Når man tror at man kan ting - Passer på at du ikke feiler - Hvilken effektivitet klarer du? Økonomisk, tid - Teknisk ikke god nok Eks: Offshore logistikk. Eks. <u>Equinor</u> prøvde å løse digitalt løsning trading av olje og gas. Ble avlyst etter at kundegruppen ikke ville ta i bruk systemet
Hva gjøre Capgemini for å digitalisere andre bedrifter som et konsultantselskap? (Capgemini sin påvirkning på andre bedrifter)	Ekspertise - Kapasitet til å gjøre oppgaver (kommer på sektor, lokalt/internasjonalt) Eks. olje og gass - Capgemini har kapasitet og kapabilitet til å løse slike oppgaver - Kvalitet og kvantitet på løsninger - Capgemini hjelper med å kartlegge mål for kunden, lage en plan, lage en analyse for behov og finne løsninger for dette. (design, utvikling og teknologi) - Dette kommer an på kriterier for kundebedriften - Ubestemte kunder: Kommer an på situasjon. Kommer med prosjekt. Kan være strategikonsulent -> Eks. hvordan minske kostnad innad bedriften. - Jobber agilt er bra med å jobbe med ubestemte kunder.	Hvordan identifiserer dere potensielle digitaliserings områder i en bedrift? Har dere noen eksempler fra prosjekter der deres digitaliseringsarbeid ga stor nytte til kunden?	Ulike Selskaper følger ulike kriterier. Eks: <u>Cloud FinOps</u> prosjekt for et energiselskap. Ville ha pådrag på <u>Azure</u> . Gikk gjennom det med kunden. Sparte kunden 3 millioner på dette. Leverte bare gjennom <u>stacken</u> med dem. Eks: Bedre brukertilfreds, tid spart og økonomisk Eks: To ulike bedrifter skulle samarbeide, og hadde ulike ERP-systemet. Jobbet med å effektivisere deres ERP-systemet. Kartla hvordan systemet så ut. (Jobbet med masterdata) Noen var vanskelig å adressere. Gjorde at bedriftene lærte av hverandre og førte til bedre samarbeid. De fikk bedre oversikt over inventar førte til mindre forbruk.
Hva slags systemer pleier Capgemini å lage? Er det CRM/ERP-systemer? (dette går sammen med spørsmålet over og under, for å få en diskusjon rundt påvirkning og produkter)	HR-systemer ERP-systemer Bruker: - Dynamics 365 - IFS - SAP CRM-systemer	Hvordan måler dere suksessen til deres digitaliseringsinitiativer?	- Kvantitative - Kvalitative - Klare <u>businesscase</u> - Varier på hvilket nivå som det leveres på - Implementeringsløsning og utviklingsløsning, tidsfrister, leveres i tide - Hvor tilfreds er du med produktet - Egne oppsett for ulike prosjekter/kunder - Drifter IT-systemer for kunden - Score for IT-drift - Net promoter score

Hvordan sikrer dere at løsningene er skalerbare og bærekraftige i sikt?	- Overgang til sky - Kunder ønsker å ting modulbasert (bedre for <u>pick and choose</u> senere) - Forretningsiden: Hvordan man forstår problemer, er det noe vi kan jobbe med framover. Eks: olje og gass - Standardisering er viktig Bærekraft - Spare 10 millioner tonn i årlig utslipp, gjennom kundene - Forbedring sammen med kunden, dokumentasjon - Eks: logistikk, effektiviserte prosessen Sparte 90 000-100 000 tonn i dette. - Datalagring, applikasjonen som kjører bør kjøres så økonomisk som mulig - Bruk av sky (<u>Cloud FinOps</u>) – unødvendig bruk av <u>cloud</u> , ubrukelig - Sikre energisertifikater - Top-Down perspektiv, istedenfor enkeltkomponent, ser man på helhet for å navigere ned i detaljer, dekker mer av problemstillingen
Hvordan balanserer dere behovet for innovasjon og sikkerhet under en digitaliseringsprosess?	Ikke sterkest innen sikkerhet Innovasjon skjer gjerne gjennom sammen med kunden og har et spesifikt mål - Nye forbedringer, krav - Sikkerhet: sikre kravene Innovasjon - Strategisk, innovasjonsstrategi, design - Tjenesteutvikling, fokus på <u>avslekt</u> mot marked og kunden - <u>Viable, feasible</u> - Inkrementelt nivå er med på hva som må til - Utside perspektiv til kunden

Viktigste temaer:

- Forstå hva gjør digitalisering løser problemstillingen for kunden. Hva dekker deres behov
- Digitalisering kan realisere løsning for et problem

Bedrift som ikke er åpen til endring?

Man tror man har løsningen fra kundens side, men tenker ikke på brukerne av systemet. Eks: Kunde som gikk fra et ERP-system, og systemet ble utviklet. Det ble sett på et teknisk perspektiv. Hadde ikke god kommunikasjon med brukerne, systemet ble «slakta av brukergruppen». Derfor er det viktig med god oppfølging av kunden og brukerne. Kunden har en for teknisk vinkling.

Intervju med Leder av en avdeling

Om intervjuobjektet:	
Kort introduksjon av deg	Leder av cloud and custom applications i Capgemini
Dine arbeidsoppgaver	Personalansvar, oppfølging av ansatte, passer på at ansatte får bygd kompetanse, passe på å få konsulenter ut på interessante prosjekter til bedrifter. Viser kompetanse i ansatte innad Capgemini globalt
Digitalisering og påvirkning:	
Hva legger du i ordet digitalisering? (vil vite dette for å danne fundament på hva som settes som betydning av digitalisering i intervjuet)	<ul style="list-style-type: none"> - Effektivisere prosesser - Handler om manuelle prosesser og transformerer til effektive og automatiserte prosesser
Hva gjøre Capgemini for å digitalisere andre bedrifter som et konsultentselskap? (Capgemini sin påvirkning på andre bedrifter)	<ul style="list-style-type: none"> - Leverer ende-til-ende-løsninger - Rådgivning, strategiplanlegging - Implementere det tekniske ved strategien - Utvikle systemer, QA, testing, design, dataanalyse og databehandling
Hva slags systemer pleier Capgemini å lage? Er det CRM/ERP-systemer? (dette går sammen med spørsmålet over og under, for å få en diskusjon rundt påvirkning og produkter)	<p>Kjernesystemer</p> <ul style="list-style-type: none"> - CRM, ERP - De fleste har fungerende kjernesystemer, har stort potensial med de systemene - ASIS-analyser for å se potensialet - Identifisere kundens behov og finner ut av hva de trenger, ToBe-analyse - GAP-analyse - Finne behov for kunden, vet ikke alltid hva de vil ha. Hvordan få bedre tall basert på dataen de har. Hypoteser, uten bevis. Her kommer digitalisering inn, for innsikt og utvikle potensialet. <p>Eks. nettbutikk, påvirkning av annonser på salg</p>
Hvordan klarer bedriftene å tilpasse seg endring og implementering av nye løsninger?	<p>Ofte gradvis.</p> <ul style="list-style-type: none"> - Derfor er det viktig å ivareta personene som tar i bruk systemet. - Tar prosessen stegvis, for å se påvirkning. <p>Rask implementering av noen endringer</p> <ul style="list-style-type: none"> - Bytte ERP system helt

Capgemini tiltak:	
Når det gjelder digitalisering, hva er de mest avgjørende faktorene for vellykket digitalisering?	<p>Faktorer</p> <ul style="list-style-type: none"> - Tilpasningsdyktigheten på bedriften - Hvor villig er de til å endre seg, endring i arbeidsoppgaver - Respekt for ansatte i bedriften <p>Kan hende toppleder ser at endringer må til for digitalisering. Derfor skjer endringen.</p>
Hva er noen av de vanligste utfordringene dere møter underveis i en digitaliseringsprosess, og hvordan håndterer dere dem?	<p>Sosiale utfordringer:</p> <ul style="list-style-type: none"> - Mangel på endringsvillighet. <p>Tekniske utfordringer:</p> <ul style="list-style-type: none"> - Udokumenterte gamle systemer – gjør det utfordrende for å lage et nytt system ut av det, eller forbedre systemet - Gamle formater – hvordan konvertere det gamle formatet til nytt? <p>Potensielle Løsninger:</p> <ul style="list-style-type: none"> - Mangel på endringsvilje: Både ledelsen (stakeholders and management) og ansatte må være inkludert i hele prosessen - Udokumenterte gamle systemer: Få gjennomgang av systemet av de som kan det best hvis mulig, gjør flere analyser av bedrift for å forstå deres behov - Gamle formater: Manuelt trykke inn den nye dataen, tungvint
Hvordan identifiserer dere potensielle digitaliseringsområder i en bedrift? Har dere noen eksempler fra prosjekter der deres digitaliseringsarbeid ga stor nytte til kunden?	<p>ASIS-analyse av virksomheten, og prøver å finne <u>pinpoints</u> innad bedriften.</p> <p>Eksempler:</p>
Hvordan måler dere suksessen til deres digitaliseringsinitiativer?	<ul style="list-style-type: none"> - Lager hypotese, inntjening - Blir egentlig aldri ferdig - Analyse på kundens <u>return of investment</u> - Økonomi, tidssparing, effektivisering, produktivitet, sette tall på retur-verdi
Hvordan sikrer dere at løsningene er skalerbare og bærekraftige i sikt?	<ul style="list-style-type: none"> - Kommer råd til kunden for å gjøre det mer bærekraftig - anbefaler løsninger som gir bedrifter et mer positivt miljøfotavtrykk

Hvordan balanserer dere behovet for innovasjon og sikkerhet under en digitaliseringsprosess?	<ul style="list-style-type: none"> - Innovasjon kan føre til bedre sikkerhet - Sikkerhet blir stadig viktigere <p>Eks: Copilot</p> <p>Utfordring:</p> <ul style="list-style-type: none"> - Ingen kontroll på dokumentene sine - Copilot kan lese gjennom alle dokumentene <p>Må sørge for sikkerhet innad bedriften, data før bruk</p> <p>Nikolai Tangen Podkast, 10% er migrert til <u>cloud</u>, dagens løsninger. Microsoft <u>sustainability cloud</u>.</p> <p>Security by obscurity</p>
--	--

Viktigste temaer

- Ikke undervurder mennesker endringsvilje, tilpasningsdyktighet
- Innovasjon og sikkerhet
- Skal du kjøre digitaliseringsprosesser, tenk på menneskene som skal ta i bruk systemet.

Appendiks 21: Known issues

Style-relaterte feil

- [Naybar items](#) er ikke [active](#) når du er på siden, skaper forvirring for bruker
- Søkning etter google bilder basert på kunde navn eller maskin type returnerer relevante/feil bilder av og til.
- Bør legge til informasjon om hvordan regnestykke til [aktivitetsscore](#) fungerer
- Kalender til tidsrom valg er lilla istedenfor å følge farge valget til resten av nettsiden
- Tabell søkefelt får en lilla understrek når bruker skriver noe i den istedenfor en farge som samsvarer med resten av designet
- Totalt antall maskiner i prosjekt detalj statistikk ser ut som skal bli påvirket av tidsrom, men vil alltid vise det samme uavhengig av tidsrom som er valgt
- Enkelte elementer mangler målenhet, f.eks. [piechart onhover tooltip](#) på [KundeDetaljer](#) og [ProsjektDetaljer](#)
- Maskinene i "maskin aktivitet i prosjekt" info boks har ikke med produkt id
- Forskjellig avrunding på statistikk og [pie charts](#) i [detaljeside](#)

Funksjonalitets relaterte feil

- Valg av tidsrom: Velger en start tid som er etter slutt tid vil krasje hele infoboksen
- HVO100 er ikke lagt inn i [piechart](#) logikken og vil sannsynligvis krasje infoboksen.
- Hover over to 5 customers pie chart på [dashboard](#) miljøfotavtrykk vil aldri gå tilbake til "total [emission](#)" etter at et kakestykke har vært [hovret](#) over
- Eksportering av grafer åpner opp en side i stedet for å prompte brukeren å laste ned som [pdf](#)
- Antall utleide maskiner i prosjekt [detalje](#) statistikk samsvarer ikke med totalt antall maskiner i prosjekt tabell listen
- "Maskin aktivitet i prosjekt" info boks vil krasje systemet

Feildata

- Står at kunder har prosjekter i kundeliste, men ikke når du trykker prosjekter.
- Står på [prosjektliste](#) at en prosjekt har f.eks. 16 maskiner, men i realiteten har den mye mindre eller ingen (sannsynligvis verktøy eller noe)
- På enkelte kunder mangler prosjekter i både [piechart](#) og [scatterplot](#). (eks. Kunde med el-data «ekno-32058»)

Ytelse

- For mange tråder på maskineffektivitet infoboksen.
- Mange spørringer er ineffektive.
- Noen ganger, dersom man laster inn prosjektdetaljer på [en prosjekt](#) med for mange maskiner, vil «Drivstoff-forbruk i forhold til antall drift-timer per maskin» [scatterplotten](#) time ut og gå til 404

Gruppekontrakt

Vi, gruppe nr. 1 består av følgende medlemmer:

Bastian Ege Tveit

Dominykas Zuoza

Dung Katrine My Huynh

Eivind Hustad Fiske

Eirik Skogland Nedland

Eziz Ømer

Skal jobbe sammen om følgende prosjekt: Prosjektgjennomføring

Bakgrunn og motivasjon Fordi: Vi ønsker å lære om, og mestre fagets læremål, slik de er oppgitt på Canvas!

Bestemmelser om fremmøte og samarbeid

Jeg forplikter meg overfor gruppen min, til at jeg vil:

- Følge forelesninger, og delta i gruppearbeid, samlinger og øvelser, be om/ta imot veiledning, gjøre selvstudium av litteratur og gjøre designforskning; delta aktivt i gruppearbeid med feltstudier.
- Respektere kravet om minst 80% fremmøte og også oppgi gyldig grunn hvis jeg er forhindret. Fravær: etter 30 min uten beskjed. Konsekvens: 3 advarsler, deretter utkastelse.
- Overholde frister og levere oppgaver som publiseres på Canvas, innen fristene.
- Sørge for at arbeid fordeles jevnt mellom gruppemedlemmer, men samtidig utnytte hver enkelt students spesielle ferdigheter og bakgrunn
- Være aktiv på discord. Evt. konflikter rundt samarbeid og innsats søkes løst gjennom diskusjon i gruppen. Fører ikke dette frem, kontaktes læringsassistent eller lærerne, så snart som mulig og før innleveringer.

Andre bestemmelser

Gruppeleder: Dominykas Zuoza

Vara: Eziz Ømer

Sted, dato, underskrifter: Kristiansand, 25.10.23

Dominykas Zuoza
Katrine Huynh
Bastian Ege Tveit
Eivind Fiske

Eziz Ømer
Eirik S. Nedland

Attest – Bachelorprosjekt våren 2024

Capgemini er et konsulentselskap med over 380 000 medarbeidere i over 50 land. I Norge er vi cirka 1600 ansatte og etablerte vårt kontor i Kristiansand høsten 2021. Siden 2021 har vi vokst betraktelig på Sørlandet, og med det har vi også fått et godt samarbeid med blant annet Universitetet i Agder. Det blir et stadig større fokus på bærekraft i samfunnet, og kundene til Capgemini er intet unntak. Vi ønsket å utfordre studentene på hvordan de kunne skape innsikt rundt miljøfotavtrykk. Det ble laget en relativt åpen oppgave hvor studentene skulle jobbe for en kunde av Capgemini.

Etter å ha presentert oppgaven på RefreshIT falt valget vårt på gruppe 1. Gruppen ble valgt fordi de fremsto faglig dyktige, strukturerte og viste entusiasme for både Capgemini og prosjektet. Det var også en gruppe som hadde arbeidet mye sammen før og hadde en god fordeling med hensyn til interesser og kompetanseområder. Det var også en klar rollefordeling i gruppen.

I gjennomføringsperioden har Capgemini bistått prosjektgruppen med ressurser tilknyttet veiledning. Disse har gitt opplæring og veiledning i Azure, prosjektledelse og presentasjonsteknikk. I tillegg har prosjektgruppen gjennomført tester med ansatte i Capgemini for å sikre kvalitet og kontinuerlig tilbakemelding underveis. Prosjektgruppen har utvist stor grad av selvstendighet og evne til å ta disse valgene selv, men også evnet å spørre om råd og veiledning ved behov. Vårt totalinntrykk av gruppen har bekreftet de initielle inntrykkene fra intervjuene. De har gjennom hele prosjektet opptrådt svært profesjonelt og entusiastisk. Prosjektet har vært gjennomført etter smidig metodikk hvor Azure DevOps har blitt brukt for prosjektstyring. De har benyttet Capgeminis lokaler under arbeidsmøter 2-3 ganger per uke. Representanter fra Capgemini har hatt regelmessig kontakt med prosjektgruppen og fulgt fremdriften gjennom sprint reviews og styringsmøter med veileder fra universitetet.

Capgeminis observasjon er at gruppen har utvist stort engasjement og forpliktelse i prosjektet. Gruppen har tilpasset seg de utfordringer og endringer som har oppstått underveis, samtidig som de har vært bevisste på tidsbruk og levert innenfor tidsfrister. Gruppen har hatt tydelige roller og gjennomført prosjektet over forventning. Capgemini ønsker likevel å nevne at selv om vår totalvurdering av gruppen har overgått forventningene, så gjelder ikke dette alle de involverte. Det har vært varierende interesse fra enkelte gruppede medlemmer gjennom prosjektets levetid. Når studentgruppen trenger hjelp med prosjektet eller ønsker kurs, forventer vi at alle deltar og viser interesse.

Capgemini er svært fornøyd med det prosjektgruppen har levert og den innsatsen de har lagt ned. Gruppen presenterte sluttproduktet under et fellesmøte for alle ansatte i Kristiansand, hvor de fikk svært gode tilbakemeldinger. Vi er veldig stolte av å være en del av dette bachelorprosjektet og ønsker å utvise takknemlighet overfor gruppen for godt samarbeid.

Lykke til videre!

Kristiansand, 06 mai 2024

Med vennlig hilsen

Vegard Sætre

Senior Consultant, Capgemini

Eirik

Som alle andre på gruppen har jeg vært med på planleggingsdelen av prosjektet. Jeg har vært med på å lage wireframes, mockups. I utviklingsdelen jobbet jeg med styling av hovedsiden, lagde total «fuel consumed infoboks» og tabellen som sammenligner effektiviteten til ulike dumpere. Jeg har jobbet med både backend og frontend i disse infoboksene. Jeg har hjulpet med å lage to av kakediagrammene på kundedetalj siden, hover effekten til topp fem kunder kakediagrammet på dashbord siden og kakediagram på prosjekt detalj siden. Jeg har jobbet med å verifisere at data som blir hentet ut fra databasen var riktig, og å finne feil og fikse bugs i applikasjonen. Sammen med resten av gruppen har jeg vært med på å skrive rapporten.

Eziz

I løpet av prosjektet har jeg hatt rollen som vara Scrum Master i tilfelle Dominykas var fraværende. Min rolle som utvikler har vært mest i frontend, og har også jobbet med backend. Spesifikke oppgaver inkluderer utvikling av universell styling, navigasjonsbar, kunde- og prosjektdetalje-side, og enkelte integrasjoner av kalenderfunksjonen i infobokser. I designdelen var jeg med på utvikling av endelig wireframes, mockups og prototype. Jeg var også involvert i analysearbeid og gjennomføring av intervjuer sammen med Katrine. Til slutt har jeg bidratt til å skrive rapporten sammen med gruppen.

Dominykas

Jeg har hatt rollen Scrum Master gjennom prosjektet, dokumentert arbeidet og holdt alle gruppe-medlemmer organisert ved å organisere prosjektets backlog og dele ut oppgaver. I starten har jeg, som de andre jobbet tett med analysearbeid, wireframes, mockup og prototype, samt ledet et intervju med en cybersikkerhetsspesialist hos Capgemini. Når utviklingsdelen startet, jobbet jeg med å hente og vise frem forskjellig statistikk for både kunder og prosjekter. Senere har jeg laget noen grafer som scatterplot i detaljesidene, og metric infoboksen i innsikt-siden, samt laget tjenesten som søker opp og henter bilder fra google. Når gruppen fikk opplæring i CTE spøringer, spilte jeg sammen med Katrine og Eirik en viktig rolle i å hente ut og kartlegge riktig data fra databasen. I tillegg har jeg laget alle illustrasjoner og animasjoner til prosjektet, laget et “eksporter graf” funksjon og drevet

med sikkerhetstesting og tiltak både for XSS og SQLi. Ved slutten av utviklingsfasen har jeg jobbet med å rydde opp i kode med analyseverktøyet Qodana. Til slutt har jeg bidratt til å skrive rapporten sammen med gruppen.

Katrine

Som sekretær var jeg ansvarlig for all kommunikasjon mellom gruppen, Capgemini og UiA. Oppgavene som sekretær består av å planlegge møter, sende eposter og meldinger, og informere alle partene. I starten av prosjektet, jobbet jeg sammen med de andre i analyse- og designprosessen, hvor jeg lagde wireframes, mockups og prototype, samt bidro i å gjennomføre intervjuer og brukertester. Som utvikler har jeg utviklet frontend og backend for flere infobokser i Dashbord og bidro til å utvikle kunde- og prosjekt-liste med Eziz. Jeg utviklet flere hover-, sorterings- og filtreringsfunksjoner, samt utviklet flere visninger av bilder, tabeller, kakediagram, miks-diagrammer (kolonnediagram + linjediagram) og scatterplot med Eirik, Dominykas og Bastian. Til slutt har jeg bidratt til å skrive rapporten sammen med gruppen.

Bastian

Gjennom prosjektperioden har jeg hatt en rolle som utvikler i form av frontend- og backend-utvikling. Jeg var med på wireframes, prototype og forslag når gruppen holdt på med designet av prosjektet. Jeg har vært med på å bidra til utforming av brukervennlighet, visualisering og funksjonaliteter. I tillegg har jeg jobbet med å utvikle interaktive grafer, søkefunksjon, implementering av try-catch funksjon og vært med på spørringer sammen med Katrine og Eirik. Jeg implementerte graf-pakken sammen med Dominykas og holdt på med det å gi tilbakemelding til andre gruppemedlemmer angående kode som bidro til et bra læringsmiljø. Til slutt har jeg bidratt med å skrive på rapporten sammen med alle de andre.

Eivind

Jeg arbeidet på design av webapplikasjonen, gjennom wireframes, mockup og prototypen, samt jobbet som utvikler på prosjektet. Som utvikler jobbet jeg både på frontend og backend. Jeg sto for design og utviklingen av flere kakediagrammer som viste info om CO₂ utslipp og drivstoff forbruk fra kunder og prosjekter. Jeg arbeidet også på en kalenderfunksjon som lot bruker velge tidsrom på fremvisning av data. Jeg bidro også med skriving på endelig prosjekt rapport.