

DOKUMARKED

VÅR 2021

Anvendt dатateknologi

Bacheloroppgave ved OsloMet

Institutt for informasjonsteknologi, Teknologi, Kunst og design (TKD)

Gruppe 43

Adrian Nilssen Nicolaisen

Eskil Nysether

Oskar Ruyter

Simen Nordgaard Renberg



Institutt for Informasjonsteknologi

Postadresse: Postboks 4 St. Olavs plass, 0130 Oslo

Besøksadresse: Holbergs plass, Oslo

PROSJEKT NR. 43

2020 - 2021

BACHELORPROSJEKT

HOVEDPROSJEKTETS TITTEL	DATO
Dokumarked	25.05 - 2021
PROSJEKTDELTAKERE	ANTALL SIDER / BILAG
Simen Nordgaard Renberg - s331179 Adrian Nilssen Nicolaisen - s333938 Eskil Nysether - s330457 Oskar Ruyter - s333993	108

OPPDRAKGIVER	KONTAKTPERSON
Way2Easy AS	Lars-Erik Erlandsen

SAMMENDRAG
Planlegging, utvikling og dokumentering av en nettbasert plattform for kjøp og salg av faglige dokumenter, samt et administreringspanel for bedriften.

3 STIKKORD	Dokumenthandel	Kommersiell
Webapplikasjon		

Sammendrag

Dette er en prosjektrapport for det avsluttende bachelorprosjektet for Institutt for informasjonsteknologi ved OsloMet - Storbyuniversitetet, våren 2021. Gruppen består av fire medlemmer fra anvendt data teknologi. Tidlig i dette prosjektet var tittelen Way2Easy Documents, men det er senere blitt bestemt at produktet gruppen har utviklet skal hete Dokumarked. Vi har derfor også valgt å la dette være tittelen til denne prosjektrapporten.

Gjennom denne rapporten vil vi gjennomgå prosessen og dokumentasjonen for å utvikle et nettbasert marked for Way2Easy. Dette dokumentet er først og fremst ment som en ressurs for vår oppdragsgiver for å kunne vedlikeholde og videreutvikle systemet som vi har levert. Produktdokumentasjonen er derfor en større del av sluttrapporten, og det er dermed valgt å ha gruppens prosessdokumentasjon vedlagt. Vi vil gjennom forklarende tekst, illustrasjoner, diagrammer og skjermbilder forklare systemets funksjon, og dermed gi en teknisk gjennomgang av hva som har blitt utviklet og hvilke utfordringer gruppen har arbeidet for å løse.

Dokumentet inneholder flere referanser til egne punkter som er lenket sammen. På grunn av dette så anbefales det å lese dokumentet i PDF-format på en PC for å kunne navigere dokumentet effektivt.

Til slutt vil vi gjerne takke hele Way2Easy for å ha gitt oss denne muligheten, og for å ha vært en så god samarbeidspartner i gjennom hele prosjektets løp. Vi vil også takke vår interne veileder Roza Abolghasemi, som har vært til stor hjelp under hele dokumenteringen av dette prosjektet, og har alltid vært tilgjengelig når vi har trengt hjelp.

Innholdsfortegnelse

1. Presentasjon	6
1.1 Gruppen	6
1.2 Oppdragsgiver	7
1.3 Terminologi	7
1.4 Bakgrunn for oppgaven	8
1.5 Dagens situasjon	8
1.6 Konsept	9
1.7 Effektmål	10
1.8 Resultatmål	10
2. Kravspesifikasjonen	11
2.1 Formål	12
2.2 Målgruppe	12
2.3 Systembeskrivelse	13
2.4 Ikke-funksjonelle krav	15
2.5 Endringer i kravspesifikasjonen	15
3. Planlegging og metode	17
Terminologi	17
3.1 Gantt diagram	18
3.2 Metodikk	19
3.2.1 Feature-drevet utvikling	20
3.2.2 Stegene	21
3.3 Verktøy	23
3.3.1 Kommunikasjon	24
3.3.2 Versjonshåndtering	24
3.3.3 Konfigurasjonsstyring	25
3.3.4 Utvikling	26
3.3.5 Design	27
3.3.6 Dokumentering	27
4. Produktdokumentasjon	29
4.1 Systemarkitektur	35
4.2 Funksjonalitet	46
4.3 Design, UI og UX	76

5. Diskusjon om faglige utfordringer og valg	88
5.1 Fremgangsmåte når utfordringer oppstår	88
5.2 Viktige valg som har blitt gjort under oppbygging av systemet	89
5.2.1 Endring av rammeverk	89
5.2.2 Endring av database	89
5.3 Tekniske utfordringer	89
5.3.1 Preview av dokument	90
5.3.2 Filtreringskombinasjoner	91
5.3.3 Håndtering av page reload	92
5.3.4 Research	92
6. Refleksjon	93
6.1 Læringsutbytte	93
6.2 Videreutvikling	94
6.2.1 Grunnlag	94
6.2.2 Forslag	95
6.3 Bachelorprosjektet under covid-19 pandemien	97
7. Konklusjon	97
Referanser	99
Liste over figurer	101
Vedlegg	108
1. Prosessdokumentasjon	109
2. Kravspesifikasjon	128
3. Prosjektets feature-liste	141
4. Prosjektdagbok	143

1. Presentasjon

1.1 Gruppen

Gruppen består av fire medlemmer fra Anvendt Datatekologi som har siden første semester hatt et godt samarbeid i både teori-dype og praktiske emner. Medlemmene i gruppen har i gjennom studiene hatt et studieløp med et fokus på programvareutvikling, webprogrammering og design.



Simen N. Renberg
s331179, Anvendt Datatekologi
s331179@oslomet.no
Gruppeleder



Adrian N. Nicolaisen
s333938, Anvendt Datatekologi
s333938@oslomet.no



Eskil Nysether
s330457, Anvendt Datatekologi
s330457@oslomet.no



Oskar Ruyter
s333993, Anvendt Datatekologi
s333993@oslomet.no

Intern veileder ved OsloMet

Roza Abolghasemi
rozaabol@oslomet.no

Ekstern veileder

Lars Erik Erlandsen
way2easyas@gmail.com

1.2 Oppdragsgiver

way2easy

Way2Easy AS er en privat bedrift og nyoppstartet aktør, stiftet i 2020. Bedriftens formål er å selge effektive og innovative produkter til studenter, for å bedre deres læreutbytte. Bedriften driver per dags dato ingen tjenester, og dette er bakgrunnen for hvordan gruppen kom i kontakt med bedriften. Grunnet en kjennskap til bedriftens daglige leder ble vi ble spurta om vi var interesserte i å utvikle ett av bedriftens planlagte produkter. Da denne utviklingen ville kollidert med vår bacheloroppgave ble bedriften bedt om å sende forespørsel til OsloMet. OsloMet godkjente oppgaven og vi ble fikk ansvaret om å utvikle dokumarked.no

Hensikten med fullførelsen av dette prosjektet har for bedriftens del vært å få utviklet en av deres første produkter, som kan gi bedriften en inntektskilde for å vokse deres virksomhet. For gruppens del har oppgaven vært en meget god introduksjon i å utvikle et produkt fra planleggingsfasen til slutfasen. I hele utviklingsprosessen har vi hatt et tett samarbeid med oppdragsgiver og vi ser frem til dagen dokumarked.no er ute i drift. Vi har lært mye om tidsstyring, utviklingsmetodikk og koordinering av oppgaver.

1.3 Terminologi

Uttrykk	Forklaring
Selger	Bruker som har lagt ut et dokument de vil selge på nettsiden
Kunde	Bruker som kjøper dokumenter som

	selges på nettsiden
--	---------------------

1.4 Bakgrunn for oppgaven

Oppdragsgiveren har planlagt å innen høsten 2021 ha Dokumarked på nettet, som er bedriftens første produkt. De vil gi studenter i norge en plattform på nettet hvor de kan kjøpe studiemateriale fra andre studenter som har fått gode resultater i et eller flere emner. Det er tenkt at denne nettsiden skal fungere som et marked, hvor Way2Easy regulerer hvilke produkter som er tillatt å selge på siden.

Dersom du som student har fått en god karakter i din skolegang skal du kunne legge ut dette materiale på siden, og tjene penger på det samtidig som du hjelper dine medstudenter. Rent teknisk vil dette si at nettsiden skal støtte kjøp, salg, lagring og behandling av .pdf og .xls samt .xlsx-dokumenter som studenter i norge har produsert under sitt studieløp. Gruppen mente denne oppgaven virket meget spennende da vi ville lære mer om hvordan betalingsløsninger og hvilke regler som gjelder for dette fungerer, og fordi vi har lyst til å få erfaring med bygging av større webapplikasjoner og kompliserte nettsider.

Løsningen vil, dersom alt går etter planen, i kort tid etter bacheloroppgaven er levert være tilgjengelig fra dokumarked.no, og vil dermed gi Way2Easy ett av de første produktene de har planlagt at bedriften deres skal drive.

1.5 Dagens situasjon

Way2Easy er en nyoppstartet bedrift. Deres mål er å utvikle metoder og produkter ment til å øke studenters nyttiggjørelse av faglige ressurser og studiemateriale. Da bedriften er nylig opprettet har de per dags dato kun styremedlemmer, og ingen avdeling for utvikling av dataeknologi.

Et av produktene Way2Easy har planlagt å tilby er dokumarked.no, som er oppgaven vi har blitt tildelt. Som tidligere nevnt har ikke Way2Easy en egen avdeling for utviklingen av denne dataeknologien, så gruppen vil bli satt i den rollen, med regelmessig kommunikasjon med oppdragsgiver for å levere et tilfredsstillende produkt.

Gruppen har ikke erfaring med utvikling av betalingsløsninger, og har heller ikke erfaring med å legge ut et ferdig utviklet produkt på nettet, så research og undersøkelse av de beste metodene gjenstår og gjøre før utviklingsprosessen utføres.

1.6 Konsept

Dokumarked skal være en nettside som vil gi studenter i Norge en plattform der de kan legge studierelaterte dokumenter de selv har produsert ut til salgs. Fra denne plattformen skal det være enkelt for privatpersoner å bidra med sine egne studiemateriale, i tillegg til å kjøpe studiemateriale de behøver, fra andre privatpersoner.

Siden internett har vært tilgjengelig for kommersiell bruk har en stor andel av virksomhetene tilgjengelig vært kjøp- og salgsvirksomhet (*Ellis-Chadwick, 2019*). En større andel av disse virksomhetene er bedrifter som selger varer/tjenester til privatpersoner, mens en plattform hvor privatpersoner kan selge digitale dokumenter til andre privatpersoner er et marked Way2Easy anser som ubenyttet.

Nettsiden kan anses som en slags etterligner av finn.no, eller amazon.co.uk, dog det er viktig å presisere at fysiske varer ikke vil være til salgs på siden. Brukere som har lastet opp dokumenter vil, etter de er blitt manuelt godkjent av bedriften, se de ligge ute til salgs på forsiden, eller ved søk på tittel/emnekode. Fra profilen deres, "Min Way2Easy", vil brukerne se oversikten over dokumentene de selger, har kjøpt, og deres eventuelle inntjening for perioden de har hatt produkter til salgs i. Utbetaling av penger tjent gjennom salg av dokumenter skal gjøres manuelt fra bedriften sin tredjeparts banktjeneste til brukerens private bankkonto.

Siden vil også for de ansatte i Way2Easy fungere som et bedriftsverktøy med støtte for regnskap, administrering av innhold og brukere på siden og samtidig gi et innblikk i nettsidens trafikk. Dokumarked vil dermed behøve en egen innlogging for administratorer som vil trenge tofaktor-autentisering. Fra administratorpanelet vil administratorer kunne godkjenne og avvise nye dokumenter som blir lastet opp,

deaktivere dokumenter som viser seg å bryte sidens retningslinjer, utestenge brukere, godkjenne dokumenter og se oversikten over inntjening/kommende utbetalinger. Vanlige brukere skal også kunne hjelpe til med modereringen av innhold på siden ved å kunne sende rapporteringer om andre brukere, dokumenter og/eller omtaler som potensielt bryter retningslinjene.

Når all funksjonaliteten som er beskrevet i kravspesifikasjonen er implementert skal dokumarked være et fullstendig produkt som gir brukere et marked for kjøp og salg av studiemateriale, og gir administratorer verktøyene som er nødvendige for å kunne regulere dette markedet.

1.7 Effektmål

Effektmålet av produktet er at Way2Easy vil få sin første tjeneste å drive. Per dags dato er de kun styremedlemmer med flere ideer, men så fort dokumarked.no er i drift, vil bedriften ha en inntektskilde som kan gi dem mulighet for å vokse. Det er vanskelig å sette et konkret effektmål før produktet ligger ute på nettet og er testet, men etter bedriftens undersøkelser skal det være et marked for dette produktet som vil oppnå denne ønskede effekten.

Gruppen vil få erfaring i å jobbe som utviklere og designere i et startup-miljø, med ansvaret og profesjonaliteten som er nødvendig i denne settingen. Vi vil få erfaring med å utvikle et produkt som vil være tilgjengelig for allmennheten. Ingen i bedriften er utdannet innen datateknologi, så erfaring i å kommunisere løsninger og systemer uten bruk av terminologi som kun er kjent for oss vil også komme godt med. Gruppen vil som nevnt gjennom dette prosjektet få erfaring med å utvikle reelle betalingsløsninger og få erfaringen med å sette et kommersielt prosjekt ut i drift.

1.8 Resultatmål

1.8.1 Brukerne

Når prosjektet leveres inn til Way2Easy skal dokumarked.no være klar for å få inn dokumenter og begynne kjøp- og salgsvirksomheten. På nettsiden skal brukerne ha

oversikt over hvilke dokumenter som ligger ute til salgs med hvilke karakterer, emner, emnekoder, studiesteder og priser som selgeren av dokumentet har opplyst om.

Brukerne skal kunne søke og filtrere dokumentene som er lastet opp for å effektivt og lettvint finne det dokumentet de leter etter. Brukerne skal kunne interagere med nettsiden gjennom å legge inn omtaler om dokumenter de har kjøpt, og få tilbakemelding fra systemet gjennom eposter dersom noe viktig for de forekommer i systemet.

Brukerne skal ha tilgang til dokumentene de selger og har kjøpt gjennom profilsiden deres, og de skal få et innblikk i deres salg og inntjeninger for den nåværende perioden.

1.8.2 Administratorene

Administratorløsningen på siden vil ikke bare være et verktøy for å moderere innholdet og brukerne som har tilgang til siden, men vil også inneholde mange sensitive og viktige funksjonaliteter for bedriften sin virksomhet. Et eksempel på slik funksjonalitet er en side hvor administratorer ser oversikten over utbetalinger til privatpersoner som har utført salg på siden. Sikkerhet er derfor av høyeste prioritet og gruppen vil utnytte løsninger som er ferdig utviklet for å sikre at dette gjøres på riktig måte.

Fra administratorsidene vil bedriften ha full oversikt over hvor mange brukere som er på siden, og hvor mange dokumenter som er på siden. Det er også herfra de får sett hvilke dokumenter som lastes opp til systemet før de eventuelt godkjennes eller avvises. Brukere og/eller dokumenter som bryter retningslinjene på siden skal utesettes, og dersom det er gjort feil under denne beslutningen skal dette kunne reverseres.

2. Kravspesifikasjonen

Dette kapittelet er en kortere gjennomgang av hovedtrekkene i kravspesifikasjonen som er vedlagt. Kravspesifikasjonen er det formelle dokumentet som definerer hvilken funksjonalitet som skal utvikles i prosjektet og hvilke rammebetingelser

utviklingsløpet skal følge. Dokumentet fungerer også som en kontrakt mellom gruppen og oppdragsgiver.

2.1 Formål

Kravspesifikasjonen var ett av de første dokumentene som ble opprettet i samarbeid med oppdragsgiveren, og har vært ett av de viktigste dokumentene i utviklingsprosessen. Den har fungert som et grunnlag til å definere milepæler under møtene vi har hatt med bedriften hver uke. Bedriften har fått en ukentlig oppdatering hver fredag om hva som er blitt gjort siden sist fredag, og under møtet har vi både fått og gitt innspill i hva som bør fokuseres på i neste uke. På grunn av denne flyten har det i løpet av utviklingsprosessen oppstått endringer i kravspesifikasjonen.

2.2 Målgruppe

Da systemet er utviklet med både en administrasjonsdel og en vanlig brukerdel kan målgruppen til systemet defineres som to distinkte grupper.

2.2.1 Administratorer

Administratorer er ansatte hos oppdragsgiveren som har som oppgave å administrere innholdet på nettsiden. I tillegg vil de bruke nettsidens funksjonaliteter rettet mot bedriftsoppgaver, som for eksempel regnskap og utbetalinger til selgere på nettsiden. Det er ment at administrasjonspanelet til systemet skal kunne benyttes av de ansatte i Way2Easy til å drifte og holde oversikt over nettsiden uten å måtte ha kompetansen til å forstå koden bak løsningen.

2.2.2 Brukere

Brukerne av nettsiden kan igjen deles opp i to forskjellige typer: Kunder og Selgere. Da nettsiden skal fungere som en markedspllass for studierelaterte dokumenter er det gitt at en bruker kan være både en selger og en kunde. For utarbeiding av brukerhistorier, definering av features og utvikling av løsningen hjelper det å ta for seg hva en potensiell selger kan ønske fra systemet, og deretter hvordan en kunde ville interagert med nettsiden.

2.3 Systembeskrivelse

Følgende er en kort gjennomgang av systemets funksjonelle egenskaper slik de i kravspesifikasjonen ble definert i starten av prosjektet. Kravspesifikasjonen kan leses i helhet under vedlegg. Endringer i disse funksjonelle egenskapene vil bli redegjort etter at dette er gjennomgått.

2.3.1 Registrering og innlogging

Det er nødt til å implementeres en innloggings-funksjonalitet på nettsiden, da dokumentene som lastes opp og kjøpes inn skal kunne lagres på brukeren som har gjort kjøpet, eller har lastet opp dokumentet.

.NET Core inneholder et system som heter ASP.NET Core Identity, som man kan velge å inkludere “out-of-the-box” under oppretting av et nytt prosjekt. Dette sikrer at lagring av brukere overholder standarder for datasikkerhet. Anbefalte brukerinnstillinger, som endring av passord, validering av epostadresser og sletting av bruker er også inkludert.

For å spare tid og for å sikre at det er så sikkert som mulig, velger vi å gå for denne løsningen.

2.3.2 Opplastning av filer

Systemet skal legge til rette for opplastning og lagring av studiemateriale. Det er kun .pdf og .xlsx-formater som skal være mulige å laste opp. Under opplasting av filene vil det i tillegg til tittel, beskrivelse, karakter og pris behøves at en rekke andre detaljer oppgis for å enklere kunne finne igjen dokumentet ved å søke på forsiden. Emnekode, studiested, emnetittel, kategori - som jus, matematikk, informatikk, pedagogikk etc er eksempler på hvordan søkeord vil bli behandlet.

Dersom en bruker laster opp et dokument med en karakter, må dette dokumenteres. For å minske sjansen for juks vil brukeren trenge å sende bedriften et vitnemål med karakteren som gjelder gjennom vitnemålsportalen.no.

2.3.3 Handlevogn

Brukere skal kunne legge flere forskjellige studiemateriale i en handlevogn og kjøpe alle på en gang.

2.3.4 Kjøp av filer

Systemet skal legge til rette for kjøp av dokumentene som er lagt i handlekurven. Bedriften har vært i kontakt med Vipps og har kommet frem til at det vil være mulig for brukere å betale for studiemateriale gjennom Vipps sitt API.

2.3.5 Utbetaling til selgere

Bedriften ønsker at systemet skal kunne automatisk utbetale inntjeningen til privatpersoner som har solgt sitt dokument på nettstedet. PayPal har et API som vi kan benytte oss av som vil la bedriften samle opp alle inntjeninger i en bankkonto, og en gang i måneden utbetale det som skyldes, minus et prosentvis beløp som går til bedriften.

I tillegg til denne programmatiske løsningen, så er bedriften i kontakt med DNB for å utforske om det er mulig å gi dette ansvaret til et tredjeparts regnskapsprogram.

2.3.6 Søk

Det skal være mulig for brukere å søke etter studiemateriale de leter etter. Oppslag i databasen basert på søkeord definert under opplastning av filer er første steg. Effektive algoritmer for å sikre kapasitet og hurtige oppslag vil bli vektlagt.

2.3.7 "Mine dokumenter til salgs" / "Mine kjøpte dokumenter"

Disse to kravene er slått sammen til ett, da de behøver to veldig like metoder på serveren. Det skal være mulig for brukeren å se hvilke studiemateriale de har ute på siden til salgs, og hvilke studiemateriale de har kjøpt, for å kunne laste de ned på nytt etter å ha kjøpt de. Dette vil være tilgjengelig på brukerens profil.

Det bør også være mulig for en bruker som selger studiemateriale å se sin totale inntjening ved oversikten over deres materiale til salgs.

2.3.8 Admin-side

Det samme nettstedet skal ha en administratorseksjon, med den funksjonalitet man kan forvente for å kunne moderere innholdet på siden, i tillegg til enkelte verktøy for å kjøre driften av sidens spesifikke virksomhet mulig. Eksempler på generelle

administratorverktøy kan være bl. a. mulighet til å utestenge brukere som har brutt brukervilkårene, deaktivere dokumenter som har blitt godkjent, men viser seg at burde vært avvist, og slette omtaler som bryter med vilkårene på siden. Eksempler på administreringsverktøy som blir spesifikke for sidens virksomhet er godkjenning og avvisning av dokumenter, hvor godkjente dokumenter blir tilgjengeliggjort for salg, mens avviste dokumenter slettes fra databasen. Dokumenter med karakterer må bevises gjennom vitnemålsportalen.no, en side som lar privatpersoner tilgjengeliggjøre hele/deler av vitnemålet sitt innen en tidsfrist. Vitnemålet inkluderes i opplastningen av karaktersatte dokumenter, og kontrolleres av en administrator gjennom administratorpanelet.

2.4 Ikke-funksjonelle krav

2.4.1 Sikkerhet

Alt på nettsiden skal lagres på en sikker måte, og det skal aldri være mulig for en uautorisert bruker å få tilgang til sensitiv data.

2.4.2 Design

Nettsiden skal følge bedriftens retningslinjer angående produktets identitet. Med det menes det at fargevalg, design og utseende skal følge arbeidsgiver sin visjon. For å oppnå dette, vil gruppen være i et møte med arbeidsgiver en gang i uken.

2.5 Endringer i kravspesifikasjonen

Endringer i kravspesifikasjonen er å forvente i en smidig utviklingsprosess. Der det har oppstått behov for endringer har disse blitt prioritert etter hvor nødvendig endringen er for at systemets formål oppnås, og hvor ressurskrevende endringen er med tanke på tid og budsjett.

2.5.1 Utbetaling til selgere

Allerede fra tidlig av i prosjektet var dette en funksjonalitet som ble brukt mye planleggingstid på. Vi visste at det finnes systemer som ville latt oss løse denne problemstillingen, men det var viktig for oppdragsgiver at brukerne ikke var nødt til å registrere seg på en tredjeparts plattform, som for eksempel PayPal, for å kunne ta i bruk tjenesten. Dette var viktig for oppdragsgiver siden dette ekstra steget kunne

potensielt føre til at brukerne av systemet valgte å ikke bruke det. Derfor kan denne problemstillingen defineres slik: Hvordan skal vi utbetale penger fra bedriftens konto til en brukers private bankkonto?

Som tidligere nevnt var Way2Easy, fra 4. Februar 2021 i kontakt med forskjellige aktører, som Fiken AS, Visma og til slutt i kontakt med DNB, i et forsøk om å finne en automatisk utbetalingsløsning som bedriften kunne benytte seg av. Den 20. Februar 2021 ble det opplyst om at den automatiske utbetalingsløsningen, slik bedriften så den for seg, ikke ville være mulig for noen tredjeparts aktør å levere.

På grunn av dette ble kravet endret til å lage en løsning for oppdragsgiver som gir en oversikt over hvilke kommende utbetalinger til privatpersoner som er i vente, for å kunne gjøre utbetalingerne manuelt gjennom bedriftens valgte banktjeneste.

Systemet skal kunne la oppdragsgiver sette statusen på disse utbetalingerne til "Betalt" eller "Mislykket" etter at en utbetaling forsøkes. Systemet skal legge til rette for at oppdragsgiver kan bruke denne siden for regnskapsformål, og analyse. Denne funksjonaliteten er dokumentert i punkt [4.2.3.7](#) i produktdokumentasjonen.

2.5.2 Filtrering

Det ble tidlig i utviklingen av søkefunksjonaliteten funnet ut at en mulighet for å filtrere dokumenter ytterligere var nødvendig. Det ble derfor implementert felt og knapper for å filtrere dokumenter basert på, blant annet, semester, skole, pris og omtaler. Full oversikt over hvilke filtreringsmuligheter som er tilgjengelige finnes i punkt [4.2.1.3.4](#) i produktdokumentasjonen.

2.5.3 Dokumenter for videregående

Det ble forespurt av oppdragsgiver om det ville være mulig for systemet å støtte salg av dokumenter på videregående nivå. Da arkitekturen var ferdig laget for opplastninger av dokumenter viste det seg at denne funksjonaliteten ikke ville være særlig omfattende eller tidkrevende, og ble derfor implementert.

2.5.4 Tofaktor-autentisering

For å sikre den administrative siden av produktet ytterligere, er det tatt i bruk ASP.NET Core Identity sin inkluderte rutine for å bruke tofaktor-autentisering.

Brukere som har rollen “Administrator” vil ved innlogging alltid bli bedt om å autentisere seg med en tredjeparts autentiseringstjeneste.

2.5.5 Profilbilde

Oppdragsgiver hadde et ønske om å la brukere på nettsiden få personliggjøre sin profil med et profilbilde. Implementeringen av lagringen av .jpg-, .png- og .jpeg-filer viste seg å være enkel, da vi allerede hadde implementert en måte å lagre .pdf-, .xls- og .xlsx-filer på server.

2.5.6 Innlogging med ekstern bruker

Vi ble bedt om det var mulig å inkludere innlogging med eksterne tjenester, som for eksempel facebook-profil og google-profil. Dette viste seg å være en enkel implementering da microsoft har utviklet egne pakker som vi kunne implementere i vårt system.

2.5.7 Ikke opprette en egen ”om-oss” side

Det ble bestemt at siden bedriften allerede har en bedriftsnettseite, ville vi ikke opprette en egen side som presenterte bedriften på dokumarked.no. Det ble heller lagt til en lenke til way2easy.no.

3. Planlegging og metode

Planleggingsprosessen begynte i oktober 2020 og ble avsluttet tidlig i januar 2021. I planleggingen ble det spesifisert hvilken utviklingsmetodikk og hvilke verktøy som skulle benyttes. Faglige utfordringer ble også definert og prioritert etter hvor mye research som ville være nødvendig før gruppen kunne utvikle systemet med god kvalitet i koden.

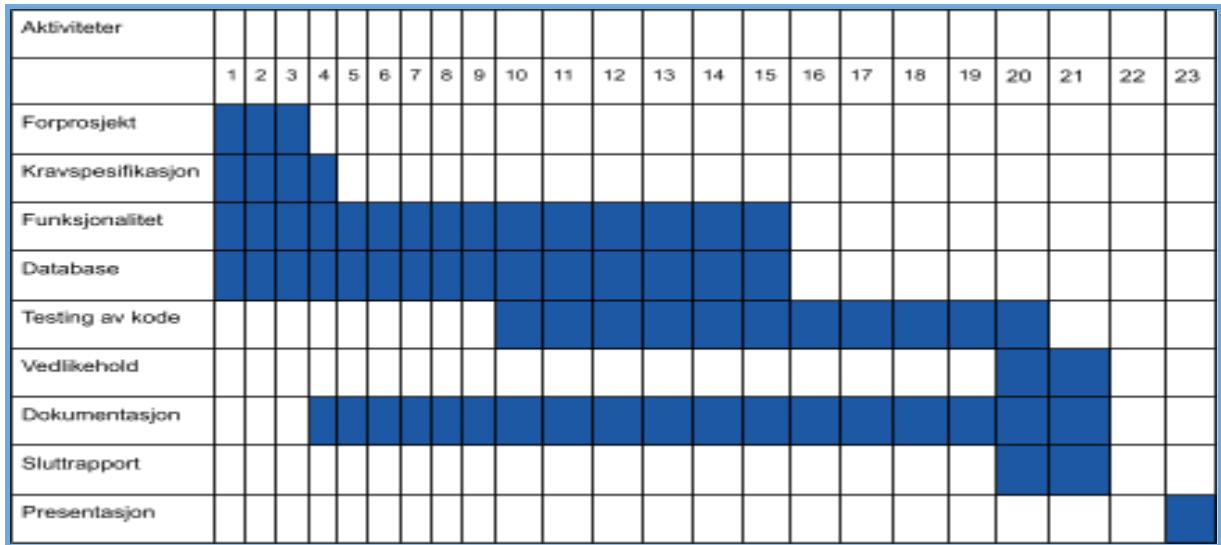
Terminologi

Begrep	Forklaring
Main branch	I Git benytter man seg av forskjellige ”grener”, hvor man oftest har en hovedgren som inneholder den kildekoden til

	produktet som er testet og godkjent til implementering. Dersom man skal utvikle ny kode basert på hovedgrenen, oppretter man en ny gren for å utvikle, for å så “merge” den nye koden, men hovedgrenen
Merge	Når man har utviklet og testet ny kode i prosjektet i sin egen gren, og skal få denne koden inn i hovedgrenen, så utføres en “merge”. Dette vil si at den gamle koden og den nye koden “smeltes sammen”.
Pull Request	En pull-request er en forespørsel om å merge en branch inn i main. Det er ofte her hvor en annen utvikler utfører en statisk kildekodeanalyse for å bestemme om koden er av høy nok kvalitet til å merges med main branch.
Wireframes	En minimalistisk representering av hvordan de mest utpregede elementene på en nettside vil se ut. Det er ikke brukt farger eller kompliserte tegninger i et wireframe. En wireframe er kun ment til å vise hvordan forskjellige elementer på en side vil ligge.
Feature	I denne rapporten er ordet feature benyttet i kontekst av feature-driven development. En feature er en del av systemets funksjonalitet.

3.1 Gantt diagram

I Planleggingsfasen ble det utviklet et gantt-diagram for å illustrere prosjektets livssyklus, og for å visualisere den planlagte fremgangen.



Figur 1: gruppens fremdriftsplan uke for uke, i form av et gantt-diagram.

Diagrammet har for mesteparten av prosessen vært viktig for gruppen, da det har gitt oss en oversikt over hvor vi står til enhver tid og om vi burde endre våre prioriteringer underveis. Det er verdt å nevne at gruppen ikke fulgt dette diagrammet til punkt og prikke. Vi har eksempelvis brukt langt lengre tid på sluttrapporten enn 2 uker.

Diagrammet har uansett vært en god ressurs for å estimere hvor mye tid resterende arbeid vil ta, fordi vi har kunnet basert oss på hvor langt før eller etter skjema gruppen ligger.

3.2 Metodikk

Gruppen var sikre på at vi ville benytte oss av en smidig metodikk da vi antok på forhånd at vi kunne påregne en del endringer i kravspesifikasjonen underveis.

Grunnene til dette er blant annet at vår oppdragsgiver sin bedrift er nyoppstartet, og hadde enda ikke fått utformet sin visjon for produktet på det punktet hvor vi takket ja til å utvikle det. Det var også klart fra starten av prosjektet at oppdragsgiver ville være en sentral del av utviklingsløpet og at de ville helst ha en del innsikt i prosjektets fremgang. For å oppnå dette ble det tidlig i prosjektet bestemt at et møte med oppdragsgiver skulle skje hver fredag for å vise frem hva vi har jobbet på, og hva vi tenker å begi oss på i " neste uke". For å oppfylle disse kravene har gruppen valgt "Feature-drevet"-utvikling.

3.2.1 Feature-drevet utvikling

Metodikken som er blitt benyttet er en litt modifisert versjon av metoden som heter “Feature-driven development”. Det er en smidig metode som kombinerer mange “best practices” fra forskjellige smidige metoder (*Palmer & Felsing, 2002*). De “beste praksisene” som gruppen har tatt fra feature-driven development er som følger:

- Utvikling etter feature

Funksjoner som er for kompliserte til å implementeres innen 2 uker brytes ned til mindre deler som defineres som egne features i systemet. Et eksempel på å bryte opp en feature av systemet til mindre features er slik:

I stedet for:

1. La en bruker *kjøpe* dokumenter

Brytes det for eksempel ned til:

1. La bruker *velge* dokumentene de vil kjøpe
2. Sett dokumentene *inn i* handlekurven
3. Kalkuler *totalpris* på et kjøp en bruker vil gjøre
4. La bruker *kjøpe* dokumentene
5. La systemet *registrere* kjøpet
6. La bruker *hente* dokumentene etter de er kjøpt

- Individuell kode- og/eller klasse-eierskap

Distinkte klasser og/eller kode i systemet eies til en hver tid av en av gruppemedlemmene, som har ansvar for å sikre at denne koden er “up-to-date” og klar for å bygges feilfritt.

- Inspeksjoner

Regelmessige inspeksjoner av kode gjøres for å sikre en god kvalitet på design og kode

- Konfigurasjonsstyring

Oversikt over prosjektets kildekode og oversikt over alle de forskjellige “delene” som til enhver tid er i spill, er oppnådd ved bruk av ClickUp, og ved bruk av GitHub - i tillegg til daglige møter og ukentlige møter med bedriften.

- Regelmessige builds

Systemet skal “alltid” være klar til å vises til oppdragsgiver, for å vise features som er utviklet, og for å få tilbakemelding på om disse features-ene opprettholder kravspesifikasjonen og kan implementeres.

- Synlighet av fremgang og resultater

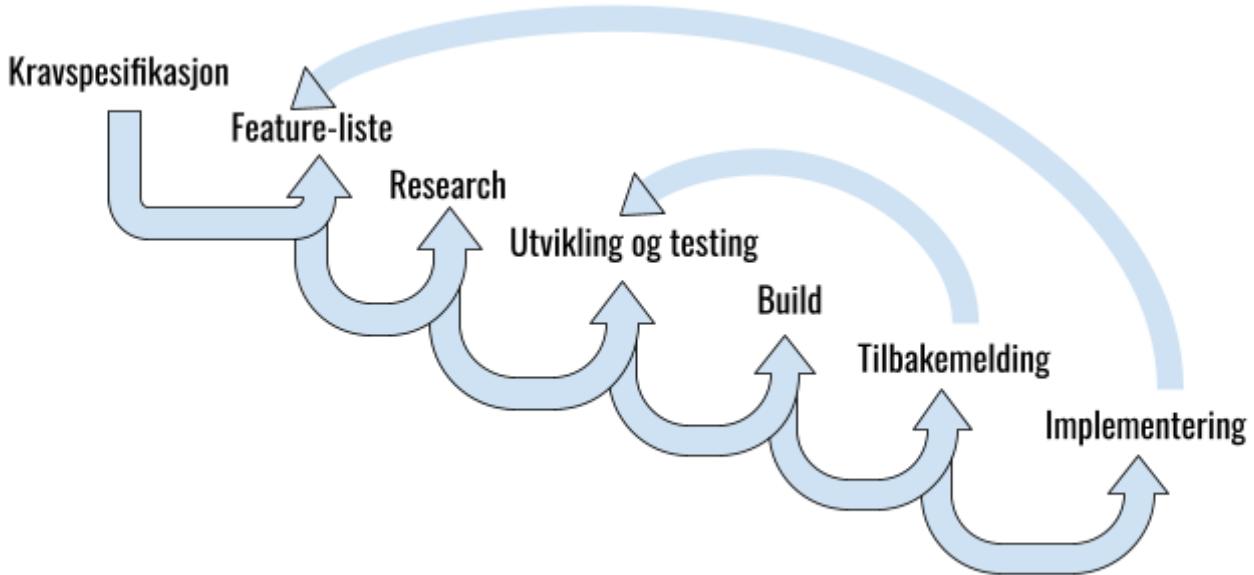
Ett av de mest utbredte “beste praksisene” i smidig utvikling er å holde en oversikt over hva som må gjøres, hva som blir gjort, og hva som er ferdiggjort. Dette er oppnådd gjennom gruppens bruk av ClickUp.

En rekke av disse “beste praksisene” utfyller oppdragsgiver sitt ønske om å få regelmessige oppdateringer og innsyn i prosjektet. Regelmessige builds og synlighet av fremgang og resultater gir et spesielt godt utgangspunkt når gruppen skal presentere fremgangen til prosjektet i møtet med oppdragsgiver hver fredag. En utdypet forklaring av hvordan denne metodikken hjalp oss holde oppdragsgiver oppdatert på deres produkt forklares i prosessdokumentasjonen, som er vedlagt.

I tillegg til å benytte seg av disse “beste praksisene” er en sentral del av feature-drevet utvikling utarbeidingen av en overordnet feature-liste. Feature-listen er enkelt og greit et dokument som definerer alle funksjonene/features som systemet skal inneholde når prosjektet er ferdig. Feature-listen vi utarbeidet for dette prosjektet ligger vedlagt

3.2.2 Stegene

Under planleggingen av utviklingsløpet til dette systemet produserte gruppen i samarbeid med arbeidsgiver denne “oppskriften” for hvordan utviklingen av hver feature ville foregå. Disse stegene er basert på arbeidsgivers ønske om å gi tilbakemelding på gruppens fremgang og kunne være med å avgjøre hva som blir implementert i systemet, og hva som må arbeides på ytterligere. I dette underkapittelet vil vi gå gjennom hver av stegene og forklare hvordan flyten i denne oppskriften fungerer i praksis.



Figur 2, utviklingsprosessen i korte trekk. Etter tilbakemelding fra arbeidsgiver blir feature enten implementert, eller videreutviklet. Dersom en feature blir implementert, går man til feature-listen for å finne neste feature som skal utvikles.

Kravspesifikasjon

Kravspesifikasjonen er det tidligste steget i vår prosess og har bestemt feature-listen som all utvikling går ut fra. Endringer i kravspesifikasjonen medfører en endring av feature-listen, som gjør at ny research, utvikling, testing og tilbakemelding er nødvendig.

Feature-liste

Oversikten over alle features som skal utvikles i systemet. Denne listen er utviklet både som et vedlagt dokument, og er overført til ClickUp for konfigurasjonsstyring.

Research

Ofte innebærer det research som må gjøres på forhånd når nye features skal utvikles. Research kan bestå av lesing av dokumentasjon eller lesing av teori og "beste praksiser" bak designmønstre.

Utvikling og testing

Den desidert største andelen av arbeidet utført i denne utviklingsprosessen. Her gjøres all utvikling av features etter de er definert og researchet. Etter at utvikleren er

fornøyd med sin kode så utføres grundige manuelle enhetstester. Dersom koden som er skrevet passerer testingen så sendes den som en pull-request i github. Statisk kildekodeanalyse utføres av en annen utvikler, før den godkjennes for en eventuell merge.

Build

Etter kildekoden for den nye, eller bearbeidede gamle featuren er merge-et med main-branch-en på github, kjøres systemet igjen og testes for å se om det fortsatt er stabilt.

Hyppige builds og hyppig testing av den nyeste build-en er også for å legge til rette for møtet hver fredag med oppdragsgiver for å få tilbakemeldinger om gruppens fremgang.

Tilbakemelding

Hver fredag siden starten av prosjektet har gruppen hatt møte med oppdragsgiver. Her viser gruppen frem det de har jobbet med denne uken, og får tilbakemelding fra oppdragsgiver om følgende features fyller kravene satt i kravspesifikasjonen. Dersom features ikke oppfyller kravene, eller bør endres for å oppfylle oppdragsgivers ønsker om å følge deres visjon, vil feature-en videreutvikles neste uke. Dersom de er fornøyd med det som er blitt utviklet, blir feature-en implementert i systemet.

Implementering

Dersom en feature er ferdig utviklet og godkjent av oppdragsgiver, blir feature-en implementert i main-branch-en til prosjektet, og blir dermed en del av det ferdige produktet.

3.3 Verktøy

Verktøy for å gjøre prosjektet enklere for gruppen er blitt benyttet i gjennom hele prosjektets løp. Vi tatt i bruk diverse verktøy for kommunikasjon, versjonshåndtering, konfigurasjonsstyring, utvikling, design og dokumentering. I dette underkapittelet vil vi gå gjennom verktøyene som gruppen har benyttet og vi vil forklare årsaken til hvorfor vi har brukt verktøyene vi har brukt.

3.3.1 Kommunikasjon

3.3.1.1 Microsoft teams

Microsoft teams er en samarbeids- og kommunikasjons-plattform, og har vært spesielt nyttig under restriksjonene satt av Oslo Kommune som følge av korona-pandemien. Teams ble benyttet på grunn av OsloMet sin integrasjon med microsoft sine produkter. Vi benyttet oss av vår profil ved oslomet for å logge inn i Teams. Verktøyet ble brukt til å arrangere videomøter, deling av filer og generell kommunikasjon innad i gruppen i gjennom hele prosjektet.

3.3.1.2 Messenger (Facebook)

Messenger er Facebooks umiddelbare kommunikasjonsplattform. Messenger ble brukt etter bedriftens ønske til å kommunisere med gruppen, både når vi hadde spørsmål som kunne besvares over tekstmeldinger og når gruppen hadde sitt ukentlige møte med oppdragsgiver.

3.3.1.3 GMail

GMail er epostleverandøren som gruppen har benyttet seg av når kontakt med vår interne veileder har vært nødvendig. Det har også vært benyttet til å sende mer formelle forespørsler til oppdragsgiver, i tillegg til å sende vedlegg som er for store til å sendes i messenger.

3.3.2 Versjonshåndtering

3.3.2.1 Git

Git er et open-source versjonshåndteringssystem utviklet for å la flere utviklere kollaborere på det samme prosjektet samtidig, uten å måtte arbeide fra den samme lokale filen. Bruken av Git har latt gruppen koordinere arbeid, og latt oss utvikle forskjellige features uavhengig fra hverandre, for å så merge det sammen til slutt.

3.3.3 Konfigurasjonsstyring

3.3.3.1 ClickUp

ClickUp er en konfigurasjonsstyringsverktøy som er blitt benyttet til å gi synlighet av fremgang og resultater. ClickUp som et produkt kan ligne litt på Trello, som er en webapplikasjon som gir brukere et brett hvor de kan sette opp virtuelle "post-it"-lapper om oppgaver og features som må utføres, blir utført, og hvilke oppgaver/features som er ferdige. ClickUp gir dette, i tillegg til langt flere konfigurasjonsstyrings-muligheter, som for eksempel å deletere oppgaver til personer i gruppen, sette frister og prioritets-nivåer på oppgaver som må utføres, og en oversikt over hvem som fullførte hvilken oppgave til hvilken tid.

3.3.3.2 Trello

Trello kan tenkes på som en såkalt "light-weight" versjon av ClickUp, som gir brukere en virtuell tavle hvor de kan henge opp lapper. Man kan skrive hva som helst på disse lappene slik man selv vil, men det er oftest brukt for å gi innsyn i systemets fremgang og resultater gjennom bruken av tre kolonner - "Må gjøres", "Blir gjort", og "Ferdig".

Trello er ikke brukt som vår konfigurasjonsstyrings-plattform, fordi ClickUp er laget for å gi en langt mer effektiv oversikt over produktet som en helhet. Trello er blitt brukt fra person-til-person slik de selv ønsker, for å hjelpe til i å gi en oversikt over enkelte "bugs" de har møtt i featurene de utvikler, og hva som må gjøres på oppgaven de har valgt fra ClickUp.

3.3.3.3 Github

Github er et skybasert vertstjeneste som hjelper brukere til å holde styr på git-repositories. Gjennom Github ble et repository opprettet, som har gitt gruppen tilgang på kildekoden samt latt medlemmer av gruppen utføre en statisk kildekodeanalyse av kode som skal "merges" til "main branch"-en.

3.3.4 Utvikling

3.3.4.1 Visual studio 2019

Visual studio 2019 er et integrert utviklingsmiljø (IDE) utviklet av Microsoft. Visual Studio Enterprise 2019 med lisens gitt av OsloMet ble brukt til utviklingen av systemet. Visual Studio er blitt valgt fordi teknologien systemet er utviklet i (ASP.NET Core MVC) er godt integrert i Visual Studio, med tilleggspakker, generert kode i tillegg til integrert og godt dokumentert systemarkitektur.

3.3.4.2 Visual studio code

Visual Studio Code kan anses som en “light-weight” versjon av Visual Studio, men er først og fremst en kraftig IDE som ikke er produsert kun for å utvikle C#-basert programvare, slik Visual Studio er. Visual studio code ble brukt mest for å løse merge conflicts i enkeltfiler uten å måtte åpne hele prosjektet, men også for noe redigering av HTML, JavaScript og CSS.

3.3.4.3 DB Browser For Sqlite

DB Browser For Sqlite er et verktøy designet til å lage og redigere SQLite-databaser. DB Browser ble brukt under utvikling mens systemet benyttet seg av SQLite i starten av utviklingen, fordi en fysisk .db-fil på disk tillatte raskere utvikling, testing og oppsett av databasen enn å sette opp en tilkobling til MSSql Server.

En annen grunn til at Sqlite ble brukt under den initiale utviklingen av databasen er fordi innad i gruppen har vi forskjellige operativsystemer (Mac OSX og Windows), og bruken av localdb gjennom MSSql tillates kun gjennom bruk av Windows-pcer.

3.3.4.4 Docker

Docker er et kraftig program som er utviklet for å kunne opprette, kjøre, og lansere applikasjoner og andre systemer i “containers”. Vår bruk av docker er litt mindre kravstør. For å tillate bruken og testingen av en lokal MSSql server hos de utviklerne i gruppen som har Mac OS X som sitt operativsystem, ble det tatt i bruk docker for å kjøre en Linux server i sin egen container, som ble koblet til gjennom en Connection String til adressen 127.0.0.1 (localhost) i kildekoden.

3.3.5 Design

3.3.5.1 Figma

Figma er et webbasert redigeringsverktøy for oppretting og redigering av wireframes, prototyper og grafikk. Tidlig i prosjektet startet vi med skissering av ulike deler av systemet ved bruk av figma. Nettsidens ferdige design er basert på disse skissene.

3.3.5.2 Adobe XD

Adobe XD er et vektorbasert verktøy for utvikling og design av brukergrensesnitt. Applikasjonen er utviklet av Adobe Inc. Adobe XD ble brukt til skissering av frontend-en til nettsiden.

3.3.5.3 Adobe Photoshop

Adobe Photoshop er et kraftig bilderedigeringsprogram utviklet av Adobe Inc, og ble brukt for redigering av digital grafikk på nettsiden.

3.3.5.4 Webaim.org

Webaim ble brukt for å regne ut kontrast-forholdet mellom bakgrunn og forgrunnsfarger. Dette ble gjort for å forsikre oss om at vi tilfredsstille WCAG 2.1-standarden for kontrast.

3.3.6 Dokumentering

3.3.6.1 Google docs

Google Docs er Google sin webapplikasjon for skriving av dokumenter, og er hvor gruppen har skrevet all dokumentasjon av- og for bacheloroppgaven.

Funksjonaliteten til programmet kan sammenlignes med Microsoft Word, dog Google Docs er en langt mindre kraftig applikasjon. Hovedgrunnen til at Google Docs er tatt i bruk over Microsoft Word er fordi Google Docs tillater kommunikasjon, kommentering av tekst og forslag for endringer i sanntid, noe som har vært spesielt nyttig under restriksjonene satt av Oslo Kommune under korona-pandemien.

3.3.6.2 Microsoft OneNote

OneNote er Microsoft sitt produkt for notatskrivning og er utviklet for å skrive mindre strukturert enn man kanskje ville i Microsoft Word. Verktøyet er blitt benyttet for å skrive korte referat om de viktigste punktene som ble tatt opp under møtene med bedriften.

3.3.6.3 Google Drive

Google Drive er google sin skylagringstjeneste som gruppen har benyttet seg av på grunn av sin integrering med Google Docs, som har tillatt gruppen å lagre alle filer og dokumentasjon relatert til bacheloroppgaven på samme sted. Google Drive har gjort at gruppen kan samarbeide i sanntid på flere dokumenter.

3.3.6.4 GoFullPage

Noen av figurene i dette dokumentet viser skjermbilder av en hel nettside, hvor normalt kun en del av denne nettsiden ville være mulig å se på en gang. GoFullPage er en utvidelse til Chromium-nettlesere som tillater brukere å ta skjermbilder av en hel nettside der innholdet på siden er stort nok til å egentlig behøve et scrolle-hjul.

4. Produktdokumentasjon

Forord

Produktdokumentasjonen inneholder den tekniske beskrivelsen av systemet.

Her går vi dypere inn på hvilke teknologier som er brukt, de forskjellige funksjonalitetene som er blitt implementert. Vi vil også gå reddegjøre programmeringsteknikker, systemarkitektur og designvalg som er blitt brukt i løpet av utviklingen av produktet. I gjennom dette dokumentet vil vi også beskrive de forskjellige tekniske utfordringene som har oppstått under systemets utvikling, og hvordan disse ble løst. Vi kommer ikke til å vise noen kildekode fra systemet, da dette er hemmeliggjort av vår kontrakt med oppdragsgiver, men noen eksempler på kildekode er inkludert for å forklare konsepter. Disse kodesnuttene er ikke representative for systemets faktiske kildekode. Vi vil likevel ta leseren gjennom systemet uten å gi spesifikk informasjon om hvordan det fungerer "bak lerretet" gjennom bruk av skjermbilder, illustrasjoner og diagrammer, samt forklarende tekst. Merk at nettsidens design har endret seg hyppig gjennom utviklingsprosessen, og det kan derfor forekomme avvik mellom design i skjermbilde og endelig design.

Way2Easy er et startup-firma hvor det kun er i dag ansatt styremedlemmer uten bakgrunn i IT-utvikling. Produktdokumentasjonen blir derfor forklart grundig i detalj for å tilrettelegge for deres kompetanse innen dette området. Det er tenkt at denne dokumentasjonen skal bli brukt av et fremtidig utviklingsteam som overtar utvikling og vedlikehold av systemet.

Innholdsfortegnelse

Terminologi	32
<hr/>	
4.1 Systemarkitektur	35
4.1.1 Generell oppbygging av systemet	36
4.1.2 Programmeringsteknikker	37
4.1.2.1 Dependency injection	37
4.1.2.2 Programvarelager	38
4.1.2.3 Lazy-loading	38
4.1.2.4 Task/Await	38
4.1.3 Database	39
4.1.3.1 Oppsett	39
4.1.4 Autogenerert kode	41
4.1.4.1 Microsoft.AspNetCore.Authentication.Facebook & Microsoft.AspNetCore.Authentication.Google	41
4.1.4.2 Automapper.Extensions.Microsoft.DependencyInjection	41
4.1.4.3 Entity Framework	42
4.1.4.3.1 Identity.EntityFrameworkCore og Identity.UI	42
4.1.4.3.2 Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore	42
4.1.4.3.3 Microsoft.EntityFrameworkCore.Proxies	43
4.1.4.3.4 Microsoft.EntityFrameworkCore.SqlServer	43
4.1.4.4 Microsoft.AspNetCore.Mvc.NewtonsoftJson	43
4.1.4.5 PDF.js	43
4.1.4.6 PDFSharp	43
4.1.4.7 Sendgrid	43
4.1.4.8 Bootstrap	44
4.1.4.9 UIKit	44
4.1.4.10 IbanNet	44
4.1.4.11 Stripe.net	44
4.1.4.12 System.Text.Encoding.CodePages	45
4.1.4.13 JQuery	45
4.1.4.14 Ion Range Slider	45
4.1.4.15 Daterangepicker	45
4.1.4.16 FontAwesome	45
4.1.4.17 Google Fonts	45
4.1.4.18 Rater.js	46
<hr/>	
4.2 Funksjonalitet	46
4.2.1 Anonyme brukere	46
4.2.1.1 Registrere bruker	46
4.2.1.2 Logge inn	48

4.2.1.3 Forsiden	49
4.2.1.4 Se detaljer om et dokument	54
4.2.1.5 Side for handlekurv	55
4.2.1.6 FAQ	55
4.2.1.7 "Om oss"	56
4.2.1.8 Brukervilkår	56
4.2.1.9 Offentlige profiler	56
4.2.1.10 Feilside	57
4.2.2 Brukere	57
4.2.2.1 Selg et dokument	57
4.2.2.2 Kjøp et dokument	59
4.2.2.3 Min Way2Easy	60
4.2.3 Admin	64
4.2.3.1 Godkjenning og avvising av dokumenter	64
4.2.3.2 Side for å kunne utestenge brukere	66
4.2.3.3 Oversikt over utestengte brukere	67
4.2.3.4 Oversikt over rapporterte dokumenter	67
4.2.3.5 Oversikt over rapporterte brukere	68
4.2.3.6 Oversikt over rapporterte omtaler	69
4.2.3.7 Oversikt over kommende utbetalinger	70
4.2.3.8 Side for å endre roller på nettsiden	73
4.2.3.9 Side for å oppdatere kommisjonsprosent	73
4.2.3.10 Side for å oppdatere prisen det koster å promotere et dokument	74
4.2.4 Andre sider	74
4.2.4.1 _layout.cshtml	74
4.2.4.1.1 Varsel om bruk av informasjonskapsler (cookies)	74
4.2.4.1.2 Header	75

4.3. Design, UI og UX	76
4.3.1 Design	77
4.3.1.1 Farger	78
4.3.2 UI	79
4.3.2.1 Ikoner	80
4.3.2.2 Skrifttype	81
4.3.3 UX	82
4.3.3.1 Mikrointeraksjoner	82
4.3.3.2 Dark mode	84
4.3.4 Universell utforming	86
4.3.4.1 Dark mode	87
4.3.4.2 Tabindex	87
4.3.4.3 Tooltips	87
4.3.4.4 Aria labels og alt-tekst	88

Terminologi

Begrep	Forklaring
HTML	HyperText markup language. Brukt for å strukturere nettsider. Kan tenkes på som skjelettet til en nettside.
CSS	Cascading style sheets. Et språk for å endre utseende til nettsiden. Alt av farger, posisjonering, animasjoner osv. er laget i css.
JavaScript (JS)	Et programmeringsspråk mest brukt for å skape interaktivitet og dynamiske nettsider. Dette kan være alt fra hva som skjer når en knapp trykkes, til innlastingen av dokumentene på forsiden. Javascript kjøres i nettleseren til klienten og ikke på server.
XML	Extensible markup language. Forskjellig fra html ved at det ikke blir brukt å strukturere nettsider, men heller dokumenter generelt. Et xml dokument definerer visse regler som gjør det mulig for både maskin og mennesker å lese innholdet.
Markup	Markup er skallkode for å strukturere et dokument. Det kan være snakk om både HTML og XML. Ved bruk av ordet markup i denne rapporten vil det som regel være snakk om HTML, med mindre noe annet spesifiseres.

DB	Database
API	Application programming interface. Et API er et grensesnitt som tillater kommunikasjon mellom forskjellige applikasjoner.
DAL	Data-Aksess-Lag. For å hindre at systemet blir “for stort” og uoversiktig er det ofte tatt i bruk lagdeling (Patton, 2006). Dette vil si at visse deler av systemet har forskjellige ansvar og forskjellige rutiner. Data-aksess-laget er ofte ett av de dypeste lagene i et system, som har ansvar for kommunisering og håndtering av databasen; DAL kan ses på som en bro mellom Controller og Databasen.
Entitet	En C#-klasse som typisk kun inneholder attributter, og ingen metoder. Denne klassen benyttes av Entity Framework Core for å definere tabellene og datafeltene i databasen gjennom kall på DbSet<T> hvor T er Entiteten.
Spørring	Når man snakker om en spørring i database kontekst så menes det en kommando for å utføre en handling mot databasen. Spørringer kan bli meget store og kompliserte, men de oppnår generelt alltid et mål innenfor CRUD, som står for Create, Read, Update, Delete. Altså å sette inn, hente,

	oppdatere, eller slette informasjon. Eksempler på spørring i pseudokode kan være ”Hent alle fornavn fra tabellen som heter Kunder”
Event	En event er noe som skjer på en nettside. Dette kan være en knapp som klikkes eller et valg i en dropdown meny. Javascript blir brukt for å fange disse handlingene og utføre tilhørende logikk.
LINQ	Language Integrated Query. Dette er et verktøy som åpner for spørrings muligheter direkte i C# språket.
MVC	Model View Controller. Et ofte brukt rammeverk for utvikling av systemer.
Modell	M-en i MVC. En modell er en klasse uten logikk, kun med datafelter (attributter).
View	V-en i MVC. Et View er som en vanlig HTML-fil, med unntaket at innholdet, og dataen i det prosesseres og opprettes på server før det blir sendt til brukerens nettleser
Controller	C-en i MVC. En controller er en fil på server som knytter klientsiden med serversiden og er ansvarlig for å passere, behandle og sende tilbake data fra ”Back-enden” til ”Front-enden”.
WebHook	En webhook er et endpoint på serveren som lytter etter varsel fra en annen

	<p>server. Det kan i praksis tenkes på som en “API i revers”. I vårt tilfelle har vi utviklet en webhook til å lytte etter varsel fra Stripe, for å utføre prosesser som må gjøres etter at en bruker har kjøpt et dokument, eller betalt for promotert plass for et dokument de eier.</p>
Dokumentpassord	<p>Alle pdf-dokumenter som er lastet opp på nettsiden skal, ved kjøp, krypteres og passordsettes med et passord kjøperen definerer. Dette er for å sikre kopibeskyttelse i henhold til åndsverkloven.</p>
BLOB	<p>Binary-Large-Object. Dette er et felt i databasen som lagrer større objekter (oftest filer) ved å ta filens binære kode og lagre dette. Herfra kan filen konstrueres igjen ved å benytte seg av dette byte-arrayet som lagres.</p>
Cookie	<p>Informasjonskapsel. Dette er en liten fil som lagres på maskinen til brukeren. Dette blir brukt hos oss for å ta vare på viktig informasjon som hvilke dokumenter brukeren har lagt til i handlekurven og om brukeren er logget inn.</p>

4.1 Systemarkitektur

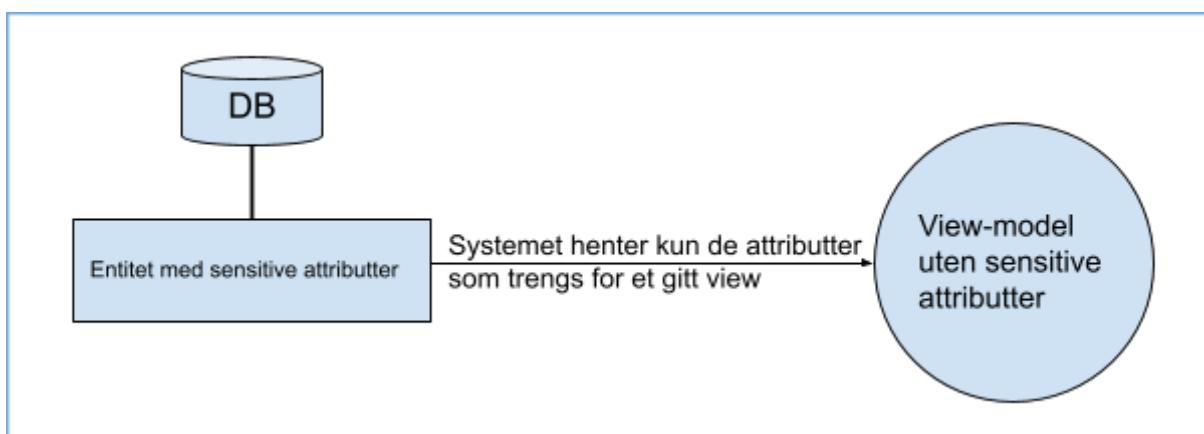
Systemets arkitektur er preget av vår bruk av Microsoft sitt ASP.NET Core MVC-rammeverk (forkortet til MVC-rammeverket) for å utvikle systemet. Den

overordnede arkitekturen følger MVC-standarden, men de andre utvidelsespakkene, rammeverkene og API-kallene som utføres av systemet gjør at vårt system likevel har en arkitektur som er utviklet av gruppen. I dette kapittelet vil vi gå igjennom den generelle oppbyggingen av systemet, som er utformet for å effektivt bruke MVC-rammeverket. Hvilke forskjellige programmeringsteknikker gruppen har benyttet seg av for å skrive kode av god kvalitet og hvordan databasen er opprettet vil også utførtes, og vi vil gjøre rede for hvilken kode som utføres av systemet som ikke er vår egen.

4.1.1 Generell oppbygging av systemet

Systemet er utviklet i MVC-rammeverk laget av Microsoft for å utvikle webapplikasjoner. Når man oppretter et prosjekt med dette rammeverket er det forskjellige alternativer man kan huke av for å få med ferdig generert funksjonalitet.

Hvert view som behøver data fra server (som for eksempel forsiden, som viser frem dokumenter som ligger ute til salgs) har en tilhørende modell. View-modellene skiller seg fra entitetene i databasen ved å inneholde kun de attributter som forekommer i hvert view. Dette oppnår effekten at et objekt til en hver tid ikke vil inneholde ellers sensitive data som medføres i entitetene. Et praktisk eksempel på dette er på forsiden, hvor hvert dokument har informasjon om studested, karakter, pris, beskrivelse m. m. som hentes fra databasen og lagres i forsidens view-modell. De sensitive attributtene i entiteten, for eksempel selve dokument-filen eksisterer ikke i view-modellen. Dette sikrer systemet mot direkte kommunikasjon med databasen.



Figur 3, en forenklet modell av hvordan forholdet mellom entitet og view-modell sikrer kommunikasjon mot DB.

4.1.2 Programmeringsteknikker

Flere programmeringsteknikker er tatt i bruk for å sikre god kvalitet i koden, samt optimalisere systemet for sitt formål. I dette prosjektet er det utviklet en kommersiell nettside som vil si at koden bør bygges for å støtte tyngre trafikk i tillegg til å sikre at varer som er betalt for alltid vil registreres. Følgende teknikker er dermed tatt i bruk for å sikre både skalerbarhet og også generell god kvalitet på kildekoden.

4.1.2.1 Dependency injection

For all kommunisering med databasen fra controller er det brukt dependency injection-mønsteret. Dette vil si at klassene unngår en direkte avhengighet til hverandre ved å “injisere” de objektene de behøver til å fungere, når de behøver de. Til tross for at vi ikke skriver kode for Android, gir Dokumentasjonen til Android Developers fortsatt et godt eksempel på hvorfor man kan trenge dependency injection:

Classes often require references to other classes. For example, a Car class might need a reference to an Engine class. These required classes are called dependencies, and in this example the Car class is dependent on having an instance of the Engine class to run. There are three ways for a class to get an object it needs:

- 1) The class constructs the dependency it needs. In the example above, Car would create and initialize its own instance of Engine.
- 2) Grab it from somewhere else. Some Android APIs, such as Context getters and getSystemService(), work this way.
- 3) Have it supplied as a parameter. The app can provide these dependencies when the class is constructed or pass them in to the functions that need each dependency. In the example above, the Car constructor would receive Engine as a parameter.

The third option is dependency injection! With this approach you take the dependencies of a class and provide them rather than having the class instance obtain them itself. (*Dependency Injection in Android*, 2020)

Dependency injection er altså blitt benyttet for å skille controlleren fra DAL ved å injisere avhengighetene i klassens konstruktør. Dette oppnår at controlleren får tilgang på metodene for å hente informasjon ut av databasen uten å måtte inneholde disse metodene selv.

4.1.2.2 Programvarelager

Programvarelager, ofte kalt software-repositories, eller bare repositories, er måten vi har konstruert vårt data-aksess-lag. Forskjellige controllere er ansvarlige for forskjellige deler av systemet. For eksempel er HomeController (/Home/) ansvarlig for anonyme brukere sin trafikk, mens UserController(/User/) er ansvarlig for innloggede brukere sin trafikk. Hver av disse controllerne har et tilhørende programvarelager i DAL som henter den informasjonen controlleren trenger fra databasen. Kommunikasjonen mellom controller og dens tilhørende programvarelager gjøres gjennom Dependency Injection, som blir forklart dypere i punkt [4.1.2.1](#).

4.1.2.3 Lazy-loading

Lazy-loading, også kalt asynkron lasting, er et designmønster benyttet for å ikke laste objekter sine attributter og metoder før det er gjort et spesifikt kall på de. Dette mønsteret er spesielt godt egnet for webutvikling, der brukere på siden ikke nødvendigvis er på en side som behøver all informasjonen et objekt inneholder til enhver tid.

4.1.2.4 Task/Await

I programvarelageret (og derfor også i alle kontrollere) er alle metoder som kommuniserer med databasen asynkrone. Dokumentasjonen til Microsoft (2016b) forklarer godene ved bruk av asynkrone metoder i Entity Framework slik:

Async programming is primarily focused on freeing up the current managed thread (thread running .NET code) to do other work while it waits for an operation that does not require any compute time from a managed thread. For example, whilst the database engine is processing a query there is nothing to be done by .NET code.

In client applications (WinForms, WPF, etc.) the current thread can be used to keep the UI responsive while the async operation is performed. In server applications (ASP.NET etc.) the thread can be used to process other incoming requests - this can reduce memory usage and/or increase throughput of the server.

På grunnlag av dette ble det bestemt å bruke async/await for å sikre systemets skalerbarhet. Nettsiden behøver også hyppige kall mot databasen, og det ble benyttet seg av asynkrone metoder i et forsøk å minske effekten av større trafikkmengde mot databasen.

4.1.3 Database

Databaseleverandøren som er blitt benyttet er MSSQL, også kalt Microsoft Sql Server. Vi har også benyttet oss av entity framework core, som er microsoft sitt rammeverk for databasehåndtering i .NET Core. LINQ er også brukt for å forenkle spørninger mot databasen.

4.1.3.1 Oppsett

For å opprette og jobbe mot en SQL database må man vanligvis skrive SQL spørninger. Disse kan være veldig lange og blir fort vanskelige å lese for mennesker. Ved bruk av entity framework trenger man kun å opprette entiteter. Ved å definere DbSet<Entitet> i databasens kontekstklasse, vil DbSet-ene bli navnet på de forskjellige tabellene, og navnet på attributtene til entitetene bli de forskjellige kolonnene i entitetenes tabell. Ved bruk av disse verktøyene forenkles det å lage, forstå og vedlikeholde spørninger, tabeller og transaksjoner. Følgende eksempel er tatt fra pensumet i Webapplikasjoner høsten 2019 (*T. Krattebøl, Personlig*

kommunikasjon, Databaser - Entity Framework - 1, 26. aug. 2019. s6) ved OsloMet viser hvor forenklet spørringene blir gjennom bruk av Entity Framework.

For å legge til et nytt bruker-objekt ved bruk av SQL i en tabell som heter Personer trengs følgende:

SQL

```
1 using System;
2
3 public class Test
4 {
5     public static void Main()
6     {
7         var sqlStreng = "Insert into Personer ([Navn],[Adresse],[Postnr], [Poststed]) values";
8
9         sqlStreng += "(" + nyPerson.Navn + "," +nyPerson.Adresse + "," +
10            + nyPerson.Postnr + "," +nyPerson.PostSted+ ")";
11
12         var kommando = new SqlCommand(sqlStreng, knytning);
13         knytning.Open();
14         kommando.ExecuteNonQuery();
15     }
16 }
17
```

Figur 4, Pseudokode som viser hvordan en spørring for å legge inn en ny person i databasen kan se ut ved bruk av SQL.

Entity Framework

```
1  using System;
2
3  public class Test
4  {
5      public static void Main()
6      {
7          db.Personer.Add(nyPerson);
8          db.SaveChanges();
9      }
10 }
11
```

Figur 5. den samme pseudokode-operasjonen som i Figur 4, gjennom bruk av Entity Framework.

Etter at entitetene (tabellene) som ville være nødvendig for applikasjonen ble planlagt, ble disse entitetenes tilhørende klasser opprettet med tilhørende attributter og deretter ble de definert i databasens kontekstklasse. En egen fil med navn DBInit.cs ble opprettet for å legge inn de grunnleggende data i databasen som ville trenges for at applikasjonen skal kjøre riktig. Et eksempel på disse er navnene på alle universiteter og høgskoler i norge, som blir benyttet når en bruker legger et universitetsdokument ut til salgs på siden. Et annet eksempel er testobjekter som ikke vil være med i databasen når systemet legges ut i produksjon, som for eksempel en ferdiglaget administratorbruker brukt til å teste administrator-delen av nettsiden.

4.1.4 Autogenerert kode

Dette systemets autogenererte kode kommer fra NuGet Package Manager og pakker/biblioteker som ikke var tilgjengelig fra NuGet Package Manager ble lastet ned fra pakkens nettside. NuGet Package Manager er utviklet av Microsoft og gir muligheten til å laste ned og produsere pakker. Pakker er kildekode som løser problemstillinger som du kan laste ned og benytte deg av i ditt prosjekt, og er ofte open-source. Vi har benyttet oss av følgende pakker og biblioteker i systemet:

4.1.4.1 Microsoft.AspNetCore.Authentication.Facebook & Microsoft.AspNetCore.Authentication.Google

Disse NuGet-pakkene er utviklet av Microsoft for å tillate brukere av nettsiden til å logge inn på nettsiden gjennom deres eksterne brukere i Facebook og/eller Google. Begge pakkene har Apache 2.0-lisens (open source).

4.1.4.2 Automapper.Extensions.Microsoft.DependencyInjection

Denne NuGet-pakken er utviklet av Jimmy Bogard for å “automappe” objekter. Kort sagt vil dette si at når en entitet skal overføre attributter til en view-modell så sparer vi tid ved å slippe å skrive kode for å overføre en og en attributt. Vi kan kalle på automapper sine metoder og oppgi objektet som skal få attributter av objektet som skal gi attributter. Pakken har MIT-lisens (open source).

4.1.4.3 Entity Framework

Entity Framework er rammeverket utviklet for databasehåndtering, utviklet av Microsoft. Pakken kan brukes som en ORM (Object Relational Mapper) som tillater utviklere å jobbe mot databaser med C#-objekter. Dette vil si at i stedet for å kalle på databasen med tradisjonelle SQL-spørrenger, vil man kunne bruke C#-kode for å utføre CRUD-operasjoner (Create, Read, Update, Delete). Alle Entity Framework-pakker bruker Apache 2.0-lisens (open source). Følgende tilleggspakker til Entity Framework Core som vi har benyttet er:

4.1.4.3.1 Identity.EntityFrameworkCore og Identity.UI

Disse to pakkene er kombinert til ett. Identity.EntityFrameworkCore er benyttet til å generere all kildekode og tilhørende logikk for å håndtere lagring og behandling av brukere på nettsiden. Identity.UI er benyttet til å generere all tilhørende markup. Tilhørende markup er endret for å opprettholde kravspesifikasjonen, men "skallet" er fortsatt ikke generert av oss.

Under oppretting av prosjektet i Visual Studio benyttet gruppen seg av Microsoft sin integrerte autentiserings-rammeverk - .NET Core Identity. Med denne pakken ble prosjektet opprettet med ferdig generert kode for registrering og innlogging for brukere, samt ekstra funksjonalitet som gjenoppretting av glemt passord, tofaktor autentisering, epostbekrefteelse og redigering av profil.

Beslutningen om å bruke ferdig-generert kode for autentisering og autorisering ble gjort fordi prosjektets krav for behandling av sensitiv persondata, inkludert (men ikke begrenset til) bankkontonummer, e-postadresse og passord. Gruppen og oppdragsgiver følte en større tillit til Microsoft sin ferdiglagde og grundig dokumenterte løsning enn den løsningen vi kunne opprettet med vår utilstrekkelige kompetanse innen datasikkerhet. Alt dette ville også gjøre at gruppen fikk bedre tid på å utvikle annen funksjonalitet.

4.1.4.3.2 Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore

Denne pakken er blitt benyttet under utvikling for å oppdage og diagnostisere feil under kommuniseringen systemet har med databasen.

4.1.4.3.3 Microsoft.EntityFrameworkCore.Proxies

Denne pakken lar oss benytte oss av lazy-loading på våre entiteter.

4.1.4.3.4 Microsoft.EntityFrameworkCore.SqlServer

Dette er entity framework sin databaseleverandør-tjeneste for Microsoft SQL Server.

4.1.4.4 Microsoft.AspNetCore.Mvc.NewtonsoftJson

Denne pakken er Microsoft sin tjeneste som tillater input og output-formatering for JSON. Pakken sin lisens er Apache 2.0 (open source).

4.1.4.5 PDF.js

PDF.js er en PDF-renderer bygget i HTML5, og er utviklet av Mozilla. Den gir oss muligheten til å hente et dokument lagret som et Binary Large Object (BLOB) i databasen, omgjøre det til en Base 64-streng og dermed konstruere en forhåndsvisning av dokumentet på klientsiden. Lisensen til denne pakken er Apache 2.0 (open source).

4.1.4.6 PDFSharp

PDFSharp er server-ekvivalenten til PDF.js, og er utviklet av empira Software GmbH. Den tillater oss å benytte oss av dokumenter lagret som Binary Large Objects (BLOBs) i databasen, og konstruere en midlertidig fysisk fil lagret i RAM på server. Til forskjell fra PDF.js vil ikke filen brukes til å konstruere en forhåndsvisning, men heller for å utføre logikk på det. Eksempler på dette er for eksempel å hente hvor mange sidetall den har, og å kryptere og passordbeskytte den. Lisensen til denne pakken er Apache 2.0 (open source).

4.1.4.7 Sendgrid

Sendgrid er leverandøren av epost-funksjonalitet, utviklet av Twilio, som tillater kommunikasjon med Twilio sin API for sending av eposter programmatisk. Lisensen til kildekoden i pakken er MIT (open source), men for å sende flere enn 100 eposter i måneden, er oppdragsgiver nødt til å betale for det.

4.1.4.8 Bootstrap

Bootstrap er et gratis bibliotek for brukergrensesnitt utviklet for å gi ferdig designede komponenter som du kan bruke i ditt HTML-dokument. Bootstrap er et meget kraftig bibliotek som gir en rekke forskjellige design-”helpers” som har hjulpet med å gi nettsiden et profesjonelt og gjennomgående design.

4.1.4.9 UIKit

UIKit er et gratis bibliotek for brukergrensesnitt som kan ligne mye på bootstrap. UIKit tilbyr diverse komponenter som man kan importere inn i sitt grensesnitt. I vårt system har vi kun tatt bruk av pagination komponenten. Denne blir forklart grundigere i punkt [4.2.1.3.5](#). Lisensen til UIKit er MIT (open-source).

4.1.4.10 IbanNet

IbanNet er en enkel pakke for validering av IBAN-nummer (Internasjonale banknummer), utviklet av Martijn Bodeman. Da oppdragsgiver skal utbetale salgsgevinster til nettsidens selgere manuelt til deres private bankkonto, vil det være nødvendig å lagre disse bankkontoene etter datatilsynets anbefalinger (*Datatilsynet, u.å.*). Dersom brukeren ikke har en norsk bankkonto vil det være nødvendig å få IBAN-nummer fra de. IBAN-nummer fungerer litt som bankkortnummer, som vil si at de inneholder en checksum som lar de “teste seg selv”, for å vite om de er gyldige eller ikke. IbanNet lar oss validere at IBAN-nummer som kommer fra klient er gyldige eller ikke. Lisensen til denne pakken er Apache 2.0 (open source).

4.1.4.11 Stripe.net

Stripe er bedriften som har laget tjenesten som er benyttet for å la brukere betale for dokumenter med bankkortet sitt. Kode som er skrevet av de inkluderer koden på klient som oppretter skjemaet for betaling, og kode på server som validerer at alt stemmer som det skal, før en bruker har mulighet til å betale. Deretter brukes stripe sin Webhook-tjeneste for å opplyse systemet om at en betaling har gått igjennom, og om en bruker skal få tilgang på et opplastet dokument. Pakken sin lisens på kildekoden er Apache 2.0 (open source), men bedriften tar 2.4% + 2 kr per betaling for norske kort, og 2.9% + 2 kr for internasjonale kort.

4.1.4.12 System.Text.Encoding.CodePages

Denne pakken er utviklet av Microsoft og er benyttet for å kunne støtte code-page baserte encodings. Spesifikt Windows-1252, som er nødvendig for å generere, forhåndsvise og nedlaste PDF-dokumenter som er generert fra Binary Large Objects (BLOBs). Pakken sin lisens er MIT (open-source).

4.1.4.13 JQuery

JQuery er et javascript-bibliotek som endrer, forenkler og tilbyr nye funksjonaliteter. All kode som gruppen har skrevet ved hjelp i JQuery er vår egen, men da selve pakken sin kildekode med tilhørende metoder og objekter ikke er utviklet av oss vil vi inkludere JQuery i dette underkapittelet. Biblioteket sin lisens er MIT (open-source).

4.1.4.14 Ion Range Slider

Rangeslider er en komponent i den større pakken ION. Denne har blitt brukt på forsiden for å la brukeren velge et datointervall for ønskede dokumenter. Pakken har MIT lisens (open-source).

4.1.4.15 Daterangepicker

Dette biblioteket er benyttet for å kunne gi administratorer en til- og fra-datovelger i kalenderformat på siden hvor de ser oversikten over bedriftens kommende og tidligere utbetalinger i systemet. Biblioteket sin lisens er MIT (open-source).

4.1.4.16 FontAwesome

FontAwesome er et gratis toolkit for skriftyper og ikoner som lar utviklere enkelt benytte seg av en rekke forskjellige ikoner i prosjektet sitt. FontAwesome har to forskjellige lisenser for deres ikoner og for deres skriftyper. I vårt system har vi kun benyttet oss av FontAwesome sine ikoner, som har lisensen CC BY 4.0 (open-source)

4.1.4.17 Google Fonts

Google Fonts er et open-source bibliotek med mer enn tusen forskjellige skriftype-familier. Vi har benyttet oss av denne tjenesten for å finne og hente skriftypen til systemet. Skriftypen som er valgt for prosjektet er Nunito, som er

lisensiert under SIL Open Font License 1.1, og dermed gratis å bruke for kommersiell bruk.

4.1.4.18 Rater.js

Rater.js er et javascript-bibliotek som tilbyr funksjonalitet for å la brukere gi stjernevurderinger. Det har blitt brukt av oss for å la brukere gi omtaler og vurdere dokumentene de har kjøpt. Lisensen til dette biblioteket er MIT (open source)

4.2 Funksjonalitet

I dette kapitlet beskriver vi den mest sentrale funksjonaliteten i detalj. Både hvordan den brukes og hvordan den er utviklet. Vår kontrakt med way2easy spesifiserer at all kode skal holdes hemmelig for offentligheten og vi kan derfor ikke vedlegge skjermbilder av kildekoden. Funksjonaliteten blir heller beskrevet med skjermbilder av nettsiden og beskrivende tekst. Da systemet som er blitt utviklet er en nettside, vil dette kapittelet være strukturert ved gå igjennom hver av de forskjellige HTML-dokumentene som eksisterer på nettsiden, og forklare hva som foregår bak lerretet for å få hentet frem data og annen nødvendig informasjon for hvert HTML-dokument.

4.2.1 Anonyme brukere

Anonyme brukere er de brukerne av nettsiden som ikke er registrert/innlogget enda. Dersom en anonym bruker prøver å få tilgang på et avlukket område på siden (som for eksempel /User/UploadDocument) vil de bli bedt om å logge inn. I dette underkapittelet vil vi redegjøre hvilke sider på nettsiden som er offentlige og kan aksesseres av alle brukere som ikke er innlogget.

4.2.1.1 Registrere bruker

At brukere kan registrere seg og opprette en egen profil på nettsiden er ansett som en av de viktigste funksjonene i prosjektet, og derfor er det valgt å benytte seg av Microsoft sitt ASP.NET Core Identity-rammeverk for autentisering og registrering av brukere.

The screenshot shows a registration form titled "Registrer" with the subtitle "Opprett en ny bruker." The form includes fields for Epost, Brukernavn, Passord, and Bekreft passord. It also features a password field for document locking and a checkbox for accepting terms. A "Registrer" button is at the bottom, along with links for external logins via Facebook and Google.

FAQ Selg Handlekurv ⓘ

Registrer

Opprett en ny bruker.

Epost

Vi kommer aldri til å vise eposten din til andre brukere.

Brukernavn

Dette er navnet som andre brukere kommer til å se.

Passord

Bekreft passord

Passord for å låse opp dokumenter du har kjøpt

Kjøpte dokumenter er kryptert og passordbeskyttet for å forsikre kopibeskyttelse. Dette passordet er ikke sikkert, og kan ikke være det samme som du bruker til å logge inn.

Jeg har lest og aksepterer [brukervilkårene](#)

Registrer

Eller logg inn med ekstern bruker

[Facebook](#) [Google](#)

Figur 6, side for å registrerer en ny bruker

Fra /Identity/Account/Register kan brukere registrere seg og definere sin epost, brukernavn og passord. I tillegg til de vanlige krav om opplysninger under oppretting av en brukerprofil trenger Dokumarked også at brukeren definerer et dokumentpassord. Alle kjøpte dokumenter krypteres og passordbeskyttes av PDFSharp-pakken (punkt [4.1.4.6](#)), for å sikre kopibeskyttelse i henhold med Åndsverkloven. Fordi vi er nødt til å hente dette dokumentpassordet og la systemet benytte seg av det for å kryptere dokumentet, er vi nødt til å stille krav til brukeren at dette passordet er ulikt deres vanlige passord, fordi det ikke kan krypteres på samme måte som deres virkelige passord.

Dersom ASP.NET Core Identity-rammeverket validerer det som brukeren har skrevet inn, vil systemet sende en bekreftelesse-epost til brukeren, og denne nye brukeren bli

lagt inn i databasen, men de vil ikke kunne logge inn med brukeren før de har bekreftet sin epost.

4.2.1.2 Logge inn

Brukere skal ha tilgang til dokumenter de har lastet opp og/eller kjøpt tidligere. Etter de har registrert seg og bekreftet sin epost, vil innlogging til siden være mulig gjennom /Identity/Account/Login

The screenshot shows a login interface with the following elements:

- Header: FAQ, Søg, Handlekurv, a user icon.
- Title: **Logg inn**
- Slogan: Søg og kjøp studiemateriale, og del din kunnskap.
- Form fields:
 - Brukernavn (Username)
 - Passord (Password)
- Checkboxes:
 - Husk meg? (Remember me?)
- Buttons:
 - Logg inn (Log in)
 - Glemt passord? (Forgot password?)
 - Registrer deg her (Register here)
- Social login:
 - Logg inn med Facebook eller Google? (Log in with Facebook or Google?)
 - Facebook button
 - Google button

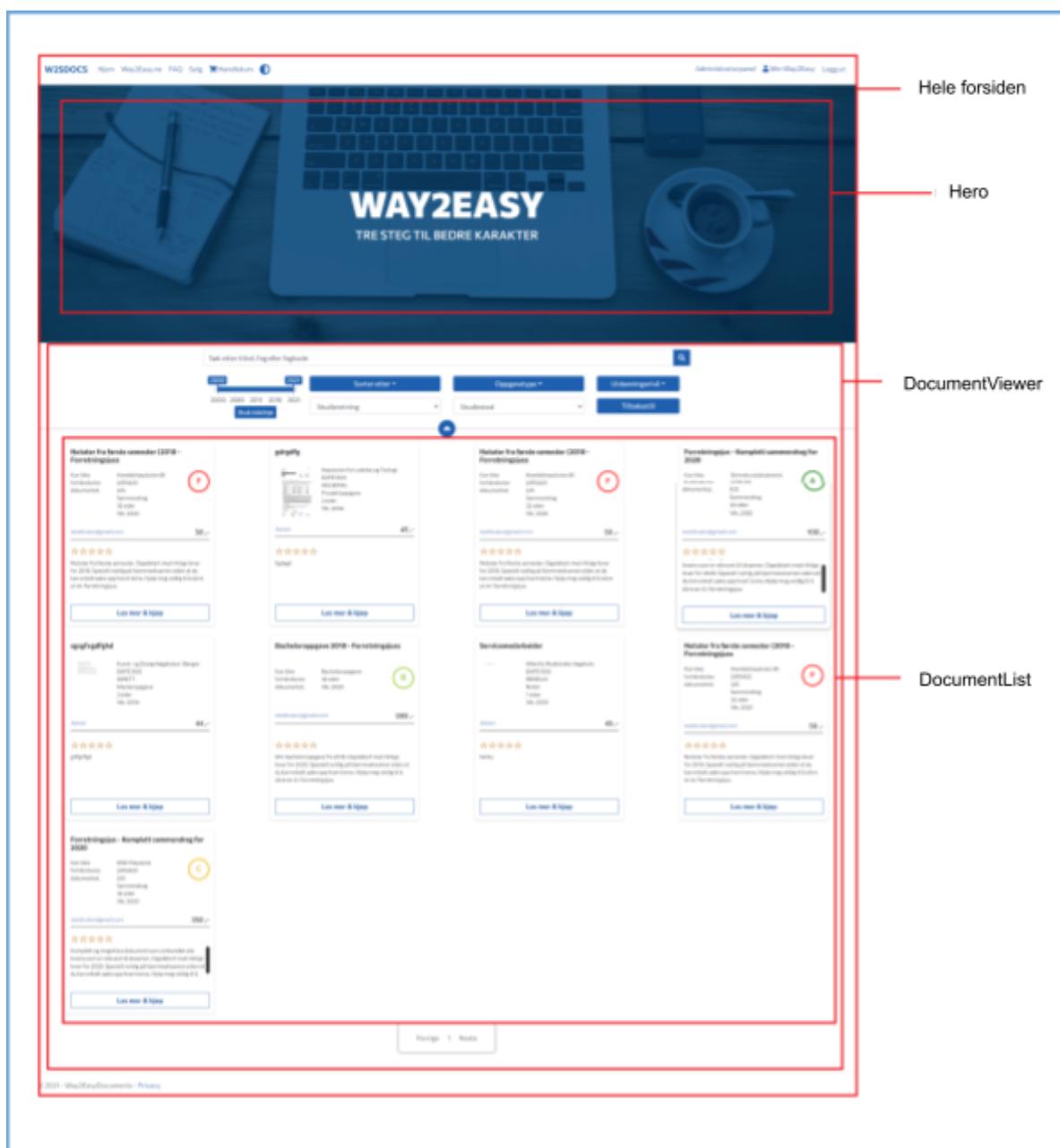
Figur 7, side for å logge seg inn med en registrert og bekreftet bruker.

4.2.1.2.1 Logge inn med ekstern bruker

Dotnet Core Identity-pakken inkluderer kode for å støtte innlogging med tredjeparts tjenester, gitt at systemet kommuniserer med Way2Easy sin bedriftskonto i disse tredjeparts tjenestene gjennom bruken av private, hemmelige nøkler i systemet. Det er lagt til to knapper for å la brukeren logge inn med enten facebook- eller google-kontoen sin, som vil lenke brukeren til den tredjeparts innloggingstjenesten sitt innloggingskjema. Dersom brukeren vil knytte denne tredjeparts-brukeren til en egen, ny konto på Dokumarked etter innlogging vil de få muligheten til det fra deres profilinnstillinger.

4.2.1.3 Forsiden

Forsiden er den siden som blir vist når en bruker først åpner nettsiden. Herfra kan brukere søke, filtrere og velge dokumenter som ligger ute til salgs. Forsiden består av tre forskjellige komponenter bygget gjennom partial views, som er en av komponentene for oppbygging av html-dokumenter som er tilgjengelig gjennom ASP.NET Core. (Microsoft, 2019). Under vil vi gå gjennom alle de forskjellige komponentene på forsiden, og hvilken funksjonalitet som er tilgjengelig for brukeren her.



Figur 8, Fullstendig bilde av forsiden med forklarende tekst for de ulike komponentene.

4.2.1.3.1 Hero

Første del av forsiden er et hero image. (Nest øverste komponenten i figur 8) Et hero image er ofte det første visuelle elementet en besøker møter på og det skal fange brukerens oppmerksomhet uten å være distraherende. (*Hero Image*, u.å.) På vår side består det av et bilde som representerer nettsidens generelle ide og en introduserende tekst.

4.2.1.3.2 DocumentViewer

Det er denne komponenten som er nest nederst i Figur 8, og det er denne som tilbyr søk, sortering og filtreringsmuligheter for utforsking av tilgjengelige dokumenter. Knappene for å velge hvilken side man vil se er også en del av denne komponenten. Dersom brukeren er logget inn vil de også ha muligheten til å velge mitt studiested og min studieretning i dropdown menyene for studiested og studieretning. Disse alternativene kan brukeren sette i Min Way2Easy. “Min Way2Easy” er forklart i punkt [4.2.2.3](#).

4.2.1.3.3 DocumentList

Det er denne komponenten som er nederst i Figur 8, og er den som faktisk inneholder dokumentene. Når nye filtrerings-alternativer blir valgt eller søker blir gjort lastes denne komponenten på nytt med nye dokumenter, uten at nettsiden trenger å lastes inn på nytt. Dette blir gjort ved hjelp av AJAX (Asynchronous Javascript and XML), som kommuniserer med systemets kontroller, som henter de tilhørende dokumentene fra databasen. JQuery blir deretter brukt for å erstatte innholdet i DocumentList med de nye dokumentene.

4.2.1.3.4 Filtrering, sortering og søker

The screenshot shows a search bar at the top with the placeholder "Søk etter tittel, fag eller fagkode". To the right of the search bar is a magnifying glass icon. Below the search bar are several filter and sort options. On the left, there is a "Bruk tidslinje" button and a timeline slider with markers for 2000, 2005, 2011, 2016, and 2021. Next to the timeline are two dropdown menus: "Sorter etter" and "Oppgavetype", both currently set to their default values. To the right of these are two more dropdown menus: "Utdanningsnivå" and "Tilbakestill". At the bottom of the interface, there is a footer bar with the text "Figur 9, Søk, sortering og filtreringsmuligheter på forsiden."

Figur 9, Søk, sortering og filtreringsmuligheter på forsiden.

Filtrering og sortering var ikke en del av den offisielle kravspesifikasjonen, men som beskrevet i punkt [2.5.2](#) så ble dette implementert i samsvar med bedriftens ønsker. De ulike alternativene er som følger:

Sorteringsvalg

Alfabetisk - A til Å
Alfabetisk - Å til A
Karakter - A til F
Karakter - F til A
Pris - lav til høy
Pris - høy til lav
Året dokumentet er skrevet - ny-gammel
Året dokumentet er skrevet - gammel-ny
Vurdering - Høyest vurderte dokumenter kommer først.

Oppgavetype

Bacheloroppgave
Masteroppgave
Eksamensbesvarelse
Arbeidskrav
Notat

Prosjektoppgave

Sammendrag

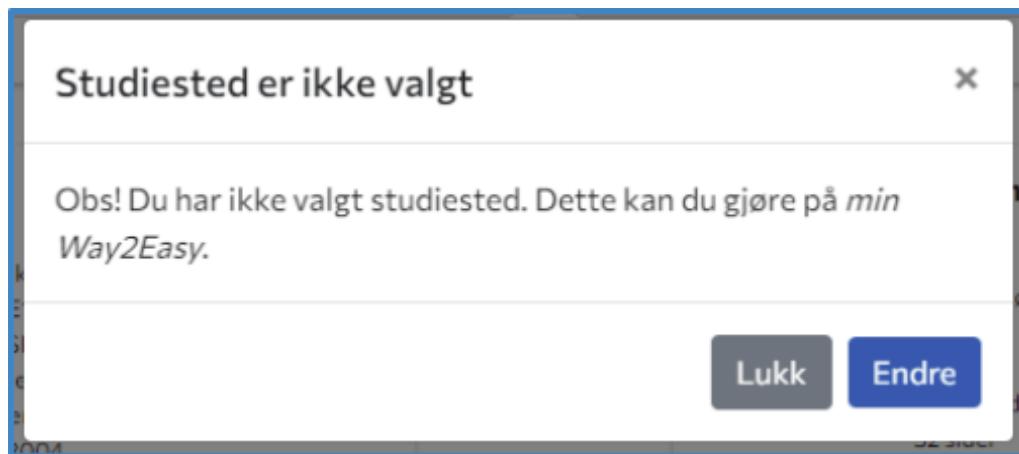
Utdanningsnivå

Høyere utdanning

Videregående skole

Studiested

En dropdown-meny som inneholder alle universiteter og høyskoler i Norge. Om brukeren er logget inn vises “mitt studiested” øverst som et valg. Dersom brukeren velger “mitt studiested”, men ikke har opplyst om studiestedet i profilen deres vil det vises en melding som ber de gjøre dette.



Figur 10, Feilmelding dersom bruker velger mitt studiested/studieretning og ikke har valgt tilsvarende.

Studieretning

En dropdown-meny som inneholder studierettingene som er tilgjengelige ved norske høyskoler. Eksempler på disse er Jus, Teknologi- og ingeniørfag eller Helsefag. Om brukeren er logget inn kommer “min studieretning” opp øverst som et valg. Det samme gjelder her som for studiested når det gjelder å vise en melding dersom brukeren ikke har valgt noe studieretning.

Dato intervall

En tidslinje hvor man kan dra begge punktene for å lage et intervall man ønsker å se dokumenter fra. Denne henter dokumenter skrevet i intervallet, ikke dokumenter lastet opp i intervallet. Selve tidslinjen er ikke utviklet av oss, men vi fant en gratis javascript plugin som tilbyr akkurat den funksjonaliteten vi ønsket. Pluginen heter ION.RangeSlider.

Nye dokumenter hentes først når “bruk tidslinje” knappen blir trykket. Dette er fordi pluginen sin standard er at en event fyres hver gang en av punktene endres. Dette førte til veldig mange databasekall da den prøvde å hente nye dokumenter hver gang et punkt endret seg.

Søk

Søkemotoren er ikke veldig komplisert. Man kan søke på tittel, fag, eller fagkode. Når søkerknappen trykkes eller enter trykkes på keyboardet vil søkerstrengen sjekkes opp mot dokumentene i databasen sine titler, fag og fagkoder. Dokumentene som inneholder søkerstrengen vises til brukeren.

4.2.1.3.5 Pagination

Med pagination menes det å kunne trykke neste og forrige side. Man skal også kunne se hvilken side man er på. Når dokumenter hentes fra databasen blir det hentet tolv dokumenter om gangen. Om brukeren trykker neste side vil de tolv neste dokumentene vises. For pagination ble det brukt UIKit sin dynamic pagination component.

Vår pagination er helt dynamisk som vil si at antall sider oppdaterer seg etter hvilke filtrerings-kombinasjoner som er valgt. Neste og forrige knappene blir også deaktivert dersom man er på enten første eller siste side.



Figur 11, pagination med side 7 som valgt side og 20 som maks antall sider.

4.2.1.4 Se detaljer om et dokument

Når en bruker har trykket på et dokument de er interessert i på f. eks. forsiden eller andre linker til dokumentet på nettsiden, blir brukerens ønske sendt til systemets controller som henter det tilhørende dokumentet i databasen. Deretter opprettes view-modellen for forhåndsvisningen av dette dokumentet. Derfra blir de sendt til [/Home/DocumentInfo/\[id-til-dokumentet\]](#) for å kunne se nærmere på det.

dokMarked Hjem Way2Easy FAQ Søg  Min Way2Easy Logg ut

Fullstendig oppsummering av LINUX og BASH-kommandoer fra labtider i Datasikkerhet ved OsloMet

Lastet opp: 05 May 2021 av Pakasone



Dette dokumentet er ikke karakterisert.

Dette er en oppsummering av alle linux og bash-kommmandoene som ble gjennomgått i labtimene i
datasikkerhet ved oslomet, høsten 2020.

Forskriftsverdi: **3.8857** nB Antall sider: **15**

Studieretning: **TEKNO - Teknologi, Ingeniørutdanning og økonomi**

Semester: **Høst**

Skrevet i: **2020**

Eksamenskode: **DAT53822**

Type Materiale: **Sammendrag**



22,-

[Legg i handlekurv](#)

Dette dokumentet sier du:

[Fortsett å handle](#)

Omtaler

© 2021 - Way2EasyDocuments - [Privacy](#)

Figur 12, siden for hvert enkelte dokument med forhåndsvisning og ytterligere informasjon om dokumentet. I dette tilfellet er forfatteren av dokumentet den som er på siden. Under "Legg til i handlekurv"-knappen står det "Dette dokumentet eier du", og da vil de ikke kunne legge det i handlekurven.

Det er også her hvor omtaler om dokumentet vises frem, og dersom brukeren er interessert i dokumentet de ser på, kan de velge å legge det i sin handlekurv. Det er verdt å nevne at selv om informasjonen som er oppgitt om dokumentet vises på denne siden, er ikke det faktiske PDF-dokumentet tilgjengelig. Fra denne siden blir det forhåndsvist minst en side av dokumentet, i tillegg til informasjon om karakter, beskrivelse av dokumentet, filstørrelse, studieretning, studiested, emnekode, type materiale - for eksempel "sammendrag" eller "eksamensbesvarelse", antall sider, format, semesteret det er skrevet og opplastingsdato.

4.2.1.5 Side for handlekurv

Dersom en bruker legger et dokument i handlevognen sin så blir dette lagret som JSON-format i en essensiell session-cookie som utløper etter en time. Flere dokumenter kan legges til i denne session-cookien, da JSON-koden som opprettes kan utvides med flere dokumenter.

The screenshot shows a user interface for a shopping cart. At the top, it says 'Min Handlekurv'. Below that is a table with columns: 'Dokument', 'Skrevet av', 'Format', and 'Pris'. There is one item listed: 'Eksamensinnslevering i Visualisering ved OsloMet' by 'Tester', in PDF format for 48 kr. A red box highlights the price '48'. Below the table, it says 'Total pris: 48 kr'. There are two buttons: 'Betalt med stripe' (highlighted with a blue border) and 'Fortsatt å handle'.

Figur 13, en bruker har lagt til et dokument med tittelen "Eksamensinnslevering i Visualisering ved OsloMet i sin handlekurv.

Fra handlekurven kan brukeren fjerne dokumenter fra handlekurvens session-cookie, forlate handlekurven og gå tilbake til forsiden, eller betale for dokumentene de har lagt til. For å betale for dokumentene i handlekurven vil brukere trenge å logge inn, eller registrere en ny bruker. Når brukeren er logget inn, vil neste steg fra handlekurven være punkt [4.2.2.2](#)

4.2.1.6 FAQ

FAQ eller "frequently asked questions" er siden hvor brukeren kan finne svar på ofte stilte spørsmål. Spørsmålene er delt inn i tre grupper: opplasting av dokumenter, kjøp av dokumenter og spørsmål om profil.

The screenshot shows a FAQ section with a title 'Ofte stilte spørsmål'. Below it, a sub-section title is 'Blir dokumentene kvalitetsskontrollert?'. A small note below says 'Dokumentene blir manuelt kvalitetsskontrollert og plagiatsjekket av Way2Easy før de gjøres tilgjengelig for salg.' An upward arrow icon is to the right of the question. Another question, 'Hvordan selger man dokumenter?', has a downward arrow icon to its right. At the bottom left, there are navigation links: 'Opplasting av dokument', 'Kjøp av dokumenter', and 'Profil'.

Figur 14, Skjerm bilde av FAQ med et spørsmål åpent.

4.2.1.7 “Om oss”

Dersom brukere har lyst på mer informasjon om Way2Easy, så ble det bestemt at det skulle finnes en lenke til Way2Easy sin bedriftsside på nettsiden. “Om oss”-knappen vil derfor ta brukeren til <https://www.way2easy.no>

4.2.1.8 Brukervilkår

Oppdragsgiver har produsert brukervilkår for å bestemme hvem og hva som skal ha tilgang til nettsiden. Brukervilkår er et vanlig dokument å ha på en nettside, spesielt nettsider som driver en tjeneste. Derfor skal brukervilkår være tilgjengelig for alle som aksesserer nettsiden

4.2.1.9 Offentlige profiler

Dersom en bruker er fornøyd med et dokument de har kjøpt kan det være naturlig at denne brukeren har lyst til å se andre dokumenter lastet opp av samme bruker. Det er derfor en offentlig profil tilgjengelig for alle, som viser oversikt over hvilke dokumenter en bruker har til salgs på siden. Det er i tillegg herfra en bruker kan rapportere andre brukere, dersom brukeren har, for eksempel, et upassende profilbilde.

The screenshot shows a user profile for 'Oskar_99'. At the top, there is a circular profile picture of a young man with short hair. To the right of the picture, the username 'Oskar_99' is displayed. Below the username, there are four lines of text: 'Studiested: OsloMet - Storbyuniversitetet', 'Antall dokumenter til salgs: 6', 'Studieretning: JUS - Jus', and 'Medlem siden: 27 Apr 2021'. Underneath this section, the heading 'Oskar_99 sine dokumenter' is shown. Three documents are listed in a grid:

- Forretningsjus - Komplett sammendrag for 2020**
Skrivekunstakademiet
JUR3420
JUS
Sammendrag
32 sider
Vår, 2020
Oskar_99 100,-
Komplett og meget bra dokument som omhandlet alle
A
- Forretningsjus - Komplett sammendrag for 2020**
NSKI Høyskole
JUR3420
JUS
Sammendrag
32 sider
Vår, 2020
Oskar_99 350,-
Komplett og meget bra dokument som omhandlet alle
C
- Bacheloroppgave 2018 - Forretningsjuss**
Bacheloroppgave
32 sider
Vår, 2020
Oskar_99 999,-
Min bacheloroppgave fra 2018. Oppdatert med riktige
B

Figur 15, Skjermbilde av en brukers offentlige profil.

4.2.1.10 Feilside

Det er alltid viktig at ingen sensitiv informasjon som kan forekomme i feilmeldinger på server blir vist til brukeren på klient. Derfor er det opprettet en feilside som brukere sendes til dersom en uforventet feil forekommer i systemet.

4.2.2 Brukere

Noe av funksjonaliteten på nettsiden er avgrenset for personer som ikke har registrert bruker på nettsiden. ASP.NET tilbyr en enkel måte å gjøre dette på ved hjelp av den innebygde [Authorize] attributten. Denne attributten kan brukes på to måter, enten å ha den over controller-klassen som gjør at alle metoder i klassen krever at brukeren er logget inn, eller å ha den over de spesifikke metodene som gjør at kun disse krever at brukeren er logget inn. Dersom man ønsker at alle metodene utenom noen få skal kreve autorisering så kan man ha [Authorize] attributten over klassen, og deretter skrive [AllowAnonymous] over metodene man ikke ønsker autorisering for å bruke. Eksempel fra Microsoft (2016a) sin dokumentasjon viser:

```
[Authorize]
public class AccountController : Controller
{
    [AllowAnonymous]
    public ActionResult Login()
    {
    }

    public ActionResult Logout()
    {
    }
}
```

Figur 16: Kodesnutt for autorisering tatt fra microsoft sin dokumentasjon av ASP.NET Core

4.2.2.1 Selg et dokument

Ett av virksomhetene som denne nettsiden tilbyr er muligheten for brukere å selge sine studiemateriale. Innloggede brukere har tilgang til å laste opp et nytt dokument til nettsiden fra /User/UploadDocument. For å komme til denne siden trykker de på "Selg" i headeren, som er synlig på hver eneste side (punkt [4.2.4.1.2](#)). Her finnes skjemaet som brukere må fylle ut for å få lastet opp dokumentet.

Legg ut et dokument til salgs

Hvilket utdanningsnivå tilhører dokumentet?

Tittelen på dokumentet*

Her er det vanlig å bruke tittelen du oppgir ved innlevering. Dersom dette studiematerialet ikke var en innlevering, kan en kort og beskrivende tittel være myltig ved salg.

Selv dokumentet*

Last opp dokumentet du vil selge Antall sider du vil forhåndsvise før betaling*

Filen du laster opp må være en pdf-fil eller et regnskav, og ende med .pdf, .xls eller .xlsx. Regnskav kan ikke forhåndsvises.

Beskrivelse av dokumentets innhold*

Her bør du oppsummere dokumentets innhold presist og konkis, med en grunne på 700 tegn. Detta er for å kunne effektivt formidle hvordan dit dokument kan myttiggjøres av studentene på nettsiden. Du er pliktig til å oppgi nøyaktig informasjon. Falsk informasjon kan medføre uteslenging fra siden, og at dokumentet fjernes.

Semesteret og året du skrev dokumentet*

Valg et semester Valg et år

Emnekode*

Hvor skrev du dette dokumentet?*

Valg studiested

Emnekode til faget dokumentet er slettet for. Eksempelvis vil faget 'BIO4200 - Molekylær evolusjon' her slettes som BIO4200. Bindstrek og mellomrom mellom bokstaver og tall er tilstatt. Dersom emnekode inneholder noen andre tegn, må du slette emnekode uten disse tegnene.

Finner du ikke ditt studiested? Send oss en epost ved way@easya.no, så skal vi legge det til!

Hva slags type studiemateriale er dokumentet?*

Valg studiemateriale

Denne informasjonen blir brukt for å filtrere søker, og for å kunne letttere gi anbefalinger til studenter fra samme studieplass.

Dersom dokumentet ble karakterisiert, kan du velge å inkludere det

Ingen karakter valgt.

Hvilken linje er dokumentet knyttet til?*

Valg studieområde

Denne informasjonen blir brukt for å kunne letttere gi studenter relevante studiemateriale under søker.

Er du sikker på hva som er en passende pris for dokumentet ditt? Du kan alltid endre prisen på dokumentet ved en senere anledning, dersom du føler den ikke tilgjør godt nok.

Jeg har lest og aksepterer [bruervilkårene](#)

Figur 17, skjemaet som lar brukere laste opp et dokument til siden.

Kravspesifikasjonen har definert et krav om å kunne verifisere karakteren brukeren oppgir på dokumentet, dersom dokumentet er karakterisert. Derfor vil 2 ytterligere felt i skjemaet vises dersom en bruker har valgt en karakter.

Dersom dokumentet ble karakterisert, kan du velge å inkludere det

A

Ved opplastning av karakterisert dokument er vi nødt til å kunne bekrefte det.
Verifiseringen av din karakter gjøres gjennom vitnemålsportalen.

Fra vitnemålsportalen.no kan du velge emnet denne karakteren tilhører, og du kan gi oss tilgang til å bekrefte karakteren du har oppgitt.
Velg 1 ukes visning av din karakter og send eposten til deg selv. Da vil du motta en lenke og en kode som vi kan bruke for å verifisere karakteren din, ved å legge de inn her:

Linken du har fått fra vitnemålsportalen

Lenken må starte med https://.

Koden du har fått fra vitnemålsportalen

Figur 18, dersom bruker oppgir karakter, vil 2 nye felt komme i skjemaet, for å fylle ut informasjon fra vitnemålsportalen.no

Dersom systemet brukerens input, vil dokumentet behandles av systemet for å legges inn i databasen som en BLOB for å kunne selges på siden. Dokumentet vil ikke være tilgjengelig for kjøp før en administrator manuelt godkjenner dokumentet (punkt [4.2.3.1](#)).

4.2.2.2 Kjøp et dokument

Brukere kan kjøpe dokumenter de har plassert i sin handlekurv. Det er tillatt å plassere dokumenter i handlekurven som en anonym bruker, men dersom de trykker på knappen for å betale for dokumentene vil de bli sendt til /User/StripePurchase, som brukere må logge inn for å aksessere. Det er tatt i bruk Stripe som betalingsløsning, og før dokumenter kan kjøpes vil Stripe prosessere handelen på server. Her sjekkes det om noen av dokumentene allerede har blitt kjøpt av brukeren, om noen av dokumentene eies av brukeren, og om en kobling til Stripe sine servere kan opprettes. Dersom alt går som det skal vil brukeren ha mulighet til å taste inn sine kortopplysninger og betale for dokumentene i handlekurven.

Betal med stripe

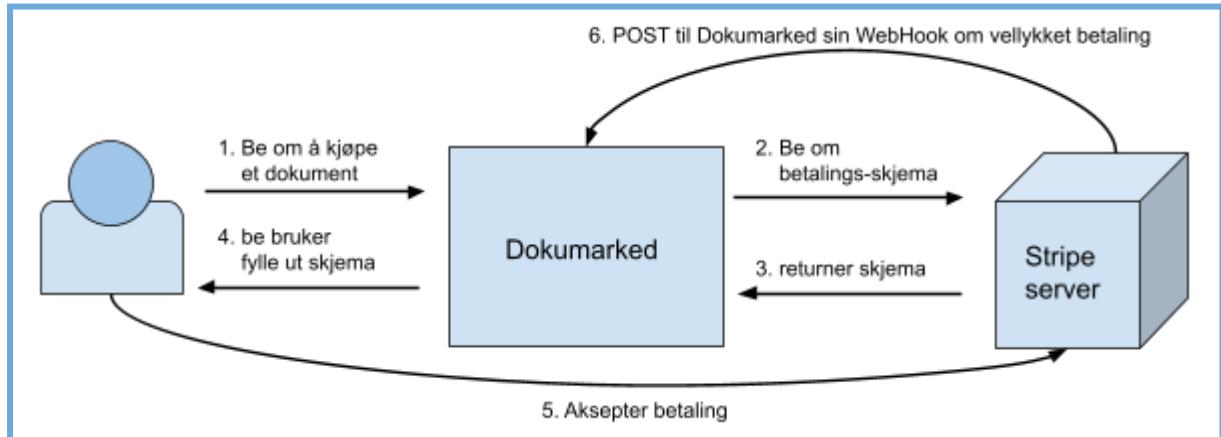
Tittel	Beskrivelse	Pris
Eksamensinnlevering i Visualisering ved OsloMet	Dette er del 1 av visualiseringeksamen ved OsloMet våren 2021	48

Betal 48 NOK

[Gå tilbake](#)

Figur 19, siden for å betale for dokumenter i handlekurven. Skjemaet for å taste inn kortopplysninger er produsert av Stripe, og systemet vi har utviklet vet aldri hva betalingsopplysningene til brukeren er.

Dersom betalingen går feilfritt vil Stripe varsle systemet ved bruk av en WebHook vi har utviklet, som vil iverksette prosessen som registrerer kjøpet, gir tilgang på dokumentet til brukeren som har utført handelen, og blant annet sender epost til forfatteren av dokumentet om at de har nettopp utført et salg. Etter at kjøpet er registrert hos brukeren av vår WebHook vil de kunne laste ned sine kjøpte dokumenter fra “Min Way2Easy” under “Mine kjøpte dokumenter” (punkt [4.2.2.3.1](#)).



Figur 20, et diagram som punktvis forklarer flyten fra når brukeren ber om å få kjøpe et dokument, til at Stripe sine servere gir varsel til Dokumarked om at en betaling har vært vellykket.

4.2.2.3 Min Way2Easy

Min way2easy fungerer som et dashboard for å kunne se oversikt over dokumenter man har kjøpt, dokumenter man har lastet opp til salgs og inntjeningshistorikk. Her kan man også redigere profilen sin. Mer detaljerte beskrivelser av de ulike funksjonalitetene blir gjort i de neste underkapitlene.

Dokument	Lastet opp	Pris	Status	Verifisert karakter	Promotert plass	Friere valg
gjørgjøf	29.04.2021	45	Ute til salgs	Dokumentet er uten karakter.	Nei	:
ngfvgfghd	29.04.2021	46	Ute til salgs	Dokumentet er uten karakter.	Nei	:
Servicemedarbeider	29.04.2021	45	Ute til salgs	Dokumentet er uten karakter.	Nei	:

© 2021 - Way2EasyDocuments - Privacy

Figur 21, brukere sin private profil, med tilgang til deres kjøpte dokumenter, opplastede dokumenter, og oversikt over inntjening

4.2.2.3.1 Laste ned et kjøpt dokument

Fra “Min Way2Easy” har brukerne mulighet til å laste ned dokumentene de har kjøpt, så mange ganger de vil, ved å trykke på “Mine kjøpte dokumenter”.

Dokument	Skrevet av	Pris	
Eksamensinlevering i Visualisering ved OsloMet	Tester	48	:
Fullstendig oppsummering av LINUX og BASH-kommandoer fra labtimer i Datasikkerhet ved OsloMet	Fakesome	22	:

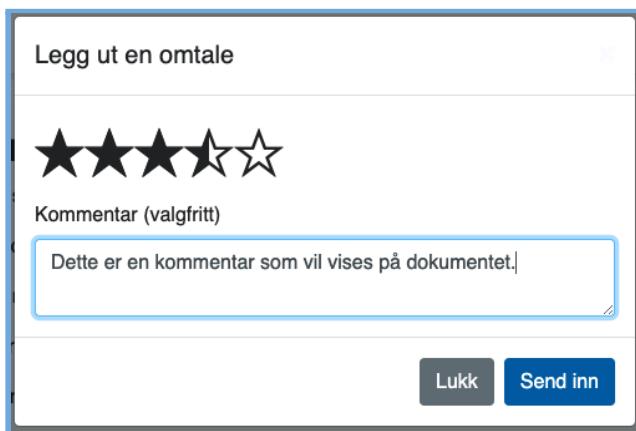
© 2021 - Way2EasyDocuments - Privacy

Figur 22, en bruker har kjøpt 2 dokumenter, og de er tilgjengelige for nedlastning under “Mine kjøpte dokumenter” i deres private profil.

Dette viser en tabell med alle dokumenter brukeren har kjøpt. Fra denne tabellen kan man laste ned dokumentet som pdf. Merk at dokumentene er passordbeskyttet og låses opp med brukerens dokumentpassord som ble oppgitt under registrering av brukeren og kan alltid endres ved å gå inn på profilens innstillinger. Da dokumentpassordet ikke er et reelt passord, og derfor ikke er kryptert, har vi latt brukerne se sitt dokumentpassord ved å gå inn på Min Way2Easy i tilfelle de glemmer hva passordet er.

4.2.2.3.2 Gi en omtale

Bruker kan legge til en ny omtale eller endre en eksisterende omtale på et dokument de har kjøpt, som vises på /Home/DocumentInfo (punkt [4.2.1.4](#)), og legges til i dokumentets gjennomsnittlige vurdering. Denne vurderingen blir brukt for å fylle stjernene på dokumentets fremvisning på forsiden og siden for å se mer om dokumentet.



Figur 23, modalen som vises når en bruker vil legge ut en omtale på et dokument de har kjøpt.

4.2.2.3.3 Endre brukerinfo

Fra Min Way2Easy kan brukeren trykke "rediger profil" eller "endre" ved siden av profilinformasjonen. Ved å gjøre dette blir brukeren sendt til /Identity/Account/Manage, hvor brukeren kan endre e-postadresse, passord, dokumentpassord, studieretning, studiested, telefonnummer og/eller profilbilde. Her kan brukeren også kontrollere sine eksterne brukere som er tilknyttet brukeren de har på dokumarked. Det vil si at de kan manuelt legge til eller fjerne enten facebook eller google som innloggingsmetode. Dersom en bruker velger et studested eller en studieretning vil de ha mulighet til å filtrere etter "mitt studiested" eller "min studieretning" på forsiden (punkt [4.2.1.3](#))

Rediger din profil

Profil Epost Passord Studier Dokumentpassord Ekstern innlogging	Profil Brukernavn <input type="text" value="Admin"/> Telefonnummer <input type="text"/> Profilbilde <div style="display: flex; align-items: center;"> <input type="button" value="Tryk her for å endre profilbilde"/> Browse </div> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Lagre"/> </div>
--	--

Figur 24, siden hvor brukere kan redigere deres profil. Til venstre vises de forskjellige typene informasjon de kan endre.

4.2.2.3.4 Endre et opplastet dokument

Dersom en bruker behøver å endre på for eksempel tittel, beskrivelse, semesteret, eller prisen til et dokument, vil de fra /User/ChangeDocument kunne gjøre dette.

Denne siden inneholder et skjema som ligner veldig på skjemaet i figur 17 i punkt [4.2.2.1](#), under opplastning av et nytt dokument, bare at noen av feltene er fjernet. Det er for eksempel verdt å nevne at dersom en bruker har lastet opp et dokument vil de ikke kunne endre den opplastede PDF-en, for å sikre at ingen brukere kan lure til seg å ha et dokument til salgs som ikke er godkjent på siden.

4.2.2.3.5 Se sin totale inntjening

Ved å trykke på “Min totale inntjening” på brukerens private profil vil de få oversikt over hvor mye penger de har samlet opp gjennom salg av dokumenter på siden. Her ser de informasjon om hvilken periode det er snakk om, hvor mye de får utbetalt, og når utbetalingen kan forventes. I tillegg til neste utbetaling, gir denne siden en oversikt over alle utbetalinger som har vært tidligere.

The screenshot shows a user profile for 'Fakesome'. At the top, there's a placeholder for a profile picture with the text 'Til Addressebok'. Below it, the user's name 'Fakesome' is displayed in bold. To the right is a link to 'Rediger Profil' (Edit Profile). Underneath the name, there are several status indicators: 'Studiested: Ingen' (Study location: None), 'Endre' (Change), 'Antall dokumenter du har til salgs: 1' (Number of documents for sale: 1), 'Studieretning: Ingen' (Study direction: None), 'Endre' (Change), 'Dokumentpassord: geir' (Document password: geir), 'Endre' (Change), and 'Medlem siden: 5 May 2021' (Member since: 5 May 2021).

Below this, there are three tabs: 'Mine kjøpte dokumenter' (Purchased documents), 'Mine dokumenter til salgs' (Documents for sale), and 'Min totale inntjening' (Total earnings). The 'Mine dokumenter til salgs' tab is selected.

Neste utbetaling	Kroner du får utbetaltt	Planlagt utbetaling
Periode 15/05/2021 - 15/06/2021	Kroner du får utbetaltt 16	Planlagt utbetaling 15/05/2021

Under the table, there's a section for 'Tidligere utbetalinger' (Previous payments) which states 'Ingen inntjening å vise.' (No earnings to show).

At the bottom left, there's a copyright notice: '© 2021 - Way2EasyDocuments - Privacy'.

Figur 25, en bruker ser sin neste utbetaling, med oversikt over hvor mye som blir utbetaltt, og når utbetalingen kan forventes. Under vises det at brukeren ikke har mottatt noen utbetalinger tidligere.

4.2.2.3.6 Betale for promotert plass

Fra "Mine dokumenter til salgs" kan brukeren velge å betale for promotert plass for sitt dokument. Etter å ha valgt dokumentet vil promotere blir de tatt til en betalingsside som også benytter seg av Stripe sin betalingsløsning, og fungerer identisk som i punkt [4.2.2.2](#), hvor vår WebHook blir opplyst om at en betaling for promotert plass har gått gjennom. Promoterte dokumenter vil bli prioritert før andre dokumenter på siden under søker, filtrering, og "surfing" på forsiden.

4.2.3 Admin

Systemet er utviklet med en egen seksjon som er kun tilgjengelig for brukere med administratorrettigheter. Herfra kan de moderere innholdet på siden, og utføre visse bedrifts-relaterte oppgaver. I dette underkapittelet vil vi gå gjennom hvilke funksjonaliteter vi har utviklet for dette.

4.2.3.1 Godkjenning og avvisning av dokumenter

Da oppdragsgiver stiller strenge krav til hva slags dokumenter som er tillatt på nettsiden, vil alle dokumenter som blir lastet opp kontrolleres og manuelt godkjennes

før de er tilgjengelige for kjøp. Etter å ha sett gjennom dokumentet, og eventuelt lenken til vitnemålsportalen brukeren har oppgitt dersom dokumentet er karaktersatt, kan dokumentet avvises eller godkjennes.

Fra /Admin/ReviewDocuments kan de ansatte i bedriften se alle dokumentene som ligger inne i databasen, men enda ikke er klarert for salg på nettsiden. Tabellen inneholder litt informasjon om hvert enkelt dokument, men dersom en administrator skal godkjenne eller avvise dokumentet er de nødt til å trykke seg inn på “Godkjenn eller avvis”-lenken på det dokumentet de har tenkt å vurdere.

Tilbake						
Forfatter	Tittel	Pris	Dato lastet opp	Sidetall	Karakter	Bekreft eller avvis
Tester	Testdokument for godkjenning	199	05/05/2021 16:33:27	105	A	Godkjenn eller avvis

Figur 26, et dokument trengs å godkjennes før det legges ut til salgs på nettsiden. En lenke til å se detaljene til dette spesifikke dokumentet er inkludert i tabellen.

Fra /Admin/ReviewSingleDocument/[id-til-dokumentet] kan administratorer godkjenne eller avvise spesifikke dokumenter, som vist i Figur 27. Dersom en bruker har oppgitt en karakter under opplastingen, er de nødt til å bevise denne karakteren gjennom vitnemålsportalen. Vitnemålsportalen er en tjeneste som lar personer velge å dele karakterer i relevante emner fra deres vitnemål, med en engangskode som lar noen se det. Dersom lenken og koden beviser karakteren, så vil dokumentet inspireres. Hvis alt stemmer, vil en administrator godkjenne dokumentet, som vil gjøre den tilgjengelig for kjøp. Dersom informasjonen ikke stemmer, vil administratoren avvise det, som vil slette det fra databasen.

The screenshot shows a document titled "Testdokument for godkjenning". The document details the following information:

- Oppgitt Karakter:** A
- Kode til vitnemålsportalen:** 456654345543
- Se vitnemål** (button)
- Filinformasjon:** 10.8667 mb, Studentring: INFOTEKN - Informasjonsteknologi og informatikk, Studienett: OsloMet - Storbyuniversitetet, Emnekode: EMNEKODE3002, Type Materiale: Arbeidskrav, Antall sider: 105, Format: .pdf, Semester: Høst, Skrevet i: 2020, Lastet opp: 5 May, 2021 av Tester
- Se dokumentet** (button)
- Beskrivelse:** Dette er dokument lastet opp for å vise hvordan en administrator kan godkjenne eller avvise dokumenter på nettsiden.
- 199,-**
- Godkjen dokumentet** (button)
- Avvis dokumentet** (button)

At the bottom left, it says: © 2021 - Way2EasyDocuments - Privacy

Figur 27, dette er siden som vises etter de har trykket "Godkjenn eller avvis"-lenken til et dokument, som vist i figur 26. Siden inneholder grundigere informasjon om dokumentet.

4.2.3.2 Side for å kunne utestile brukere

Dersom det blir oppdaget at en bruker har brutt brukervilkårene og det bestemmes at denne brukeren ikke lenger skal ha tilgang til nettsiden, kan administratorer utestile de fra /Admin/Ban.

Tester		
Brukernavn	Registreringsdato	Årsak til utestenging
Tester	05/05/2021 13:31:16	Brudd på brukervilkår

Figur 28, en bruker med navn "Tester" holder på å bli utestengt fra nettsiden, etter at brukernavnet deres er søkt opp.

Siden har et søkefelt som vil returnere brukere som har brukernavn som inneholder søkestrengen. Dersom det er treff på en eller flere brukere, vil en administrator ha

mulighet til å utestenge brukeren fra siden etter å ha oppgitt en årsak til utestengingen.

Utestengte brukere sine dokumenter vil bli satt tilbake til å være inaktive, som gjøres at de ikke lenger kan kjøpes, men de bevares fortsatt i databasen. Dette gjøres for å sikre at dersom en bruker har kjøpt et dokument vil de alltid kunne laste ned dokumentet fra siden.

4.2.3.3 Oversikt over utestengte brukere

Dersom en bruker har blitt utestengt ved en feil, eller bedriften bestemmer at de ikke lenger skal være utestengte, så kan de finne brukeren gjennom /Admin/BannedUsers.

Tilbake				Administratorpanel	Min Way2Easy	Logg ut
Brukernavn	Dato brukeren ble utestengt	Årsak til utestengning	Utestengt av			
Tester	05/05/2021 17:07:49	Upassende innhold i profilbilde	Admin		Gi tilbake tilgang	

Figur 29, etter at brukeren "Tester" har blitt utestengt, ligger brukeren på oversikten over utestengte brukere.

Herfra kan de gi tilbake tilgang til brukerne. Siden er i tillegg ment som en oversikt over utestengte brukere, uten å nødvendigvis gi tilbake noen tilgang.

4.2.3.4 Oversikt over rapporterte dokumenter

Way2Easy er avhengige av tilbakemeldinger og rapporteringer av dokumenter. Etter at et dokument er manuelt godkjent av Way2Easy kan det fortsatt forekomme at dokumentet ikke oppfyller kravene satt av Way2Easy.

Fra /Admin/DocumentReports kan administratorer se oversikten over hvilke dokumenter som har blitt rapportert, og hvor mange ganger hvert dokument er blitt rapportert.

Tilbake			
Tittel	Eier av dokumentet	Antall ganger dokumentet er rapportert	
Eksamensinnslevering i Visualisering ved OsloMet	Tester	1	Se dokumentets rapporter

Figur 30, et dokument eid av brukeren “Tester” har blitt rapportert en gang.

Herfra kan administratorer trykke seg inn på hvert av dokumentene som har mottatt rapporter, for å se hver enkel rapport og, dersom det viser seg at rapportene er sanne, deaktivere dokumentet for kjøp.

Tilbake			
Dato for rapportering	Rapportert av	Kategori	Kommentar
05/05/2021 17:37:55	Admin	Feilinformasjon i dokumentet	Dette dokumentet inneholder ikke det som stod i beskrivelsen.

Figur 31, en oversikt over hver av rapportene dokumentet til brukeren “Tester” har fått.

4.2.3.5 Oversikt over rapporterte brukere

På dokumarked.no er det et antall steder hvor brukere får bestemme innholdet på siden. Et eksempel på dette er omtaler. Etter at en bruker har kjøpt et dokument vil de få mulighet til å legge ut en omtale i form av “en til fem” stjerner. Her er det også mulighet for å skrive en valgfri kommentar. Dersom en bruker trakasserer eller benytter seg av upassende språkbruk vil det være mulig for andre brukere å rapportere denne brukeren. Fra /Admin/UserReports kan bedriften se oversikten over alle brukere som er rapportert, og hvor mange ganger hver bruker er blitt rapportert.

Tilbake			
Brukernavn	Antall ganger brukeren er rapportert		
Tester	1		Se brukerens rapporter

Figur 32, oversikt over alle brukere som er blitt rapportert, og hvor mange ganger brukerne har blitt rapportert

Ved å trykke på “Se brukerens rapporter” vil administratorer få oversikt over alle rapportene denne brukeren har mottatt.

Tilbake	Dato for rapportering	Kategori	Kommentar
	05/05/2021 17:33:55	Annet	Brukeren har et upassende profilbilde

Figur 33, alle rapportene som en enkel bruker har fått. I dette tilfellet har brukeren fått en rapport i kategorien "Annet", med kommentar: "Brukeren har et upassende profilbilde".

Dersom det viser seg at rapporteringene en bruker har mottatt stemmer og brukeren bryter nettsidens brukervilkår, kan en administrator uteslunge denne brukeren fra /Admin/Ban.

4.2.3.6 Oversikt over rapporterte omtaler

Dersom en omtale inneholder en kommentar som bryter brukervilkårene eller er upassende på en annen måte, har brukere av siden mulighet til å rapportere omtalen. Dette er lignende funksjonalitet som rapportering av brukere. Forskjellen er at om en bruker ellers ikke har brutt noen brukervilkår, men har lagt inn en upassende omtale, er det blitt bestemt at individuelle omtaler skal kunne slettes, uten å nødvendigvis uteslunge brukeren som har skrevet den.

Tilbake	Brukern som har gjort omtalen	Dokumentet som er omtalt	Selv omtalelsen	Antall ganger omtalen er rapportert	
	Admin	Eksamensinnlevering i Visualisering ved OsloMet	Dette er en rapportert omtale	1	Se omtalens rapporter

Figur 34, oversikt over alle omtaler som er blitt rapportert, og hvor mange ganger omtalene har blitt rapportert

Fra /Admin/ReviewReports kan bedriften se oversikten over alle omtaler som er rapportert, og hvor mange ganger hver omtale er blitt rapportert. Ved å klikke på knappen hvor det står "Se omtalens rapporter" kan administratorer se hver eneste rapportering som tilhører en omtale.

Tilbake			
Dato for rapportering		Kategori	Kommentar
05/05/2021 17:44:35		Upassende navn	
Slett omtale			

Figur 35, oversikt over alle rapporter en enkel omtale har fått. I dette tilfellet har den fått en rapport i kategorien "Upassende navn" uten noen kommentar fra brukeren som rapporterte den.

Dersom administratorene er enige med rapportene, og omtalen bryter brukervilkårene, kan de herfra slette omtalen fra siden.

4.2.3.7 Oversikt over kommende utbetalinger

Way2Easy Dokumarked er en handelsplattform, hvor bedriften er ansvarlig for å utbetale salgsgevinster til selgerne på siden. I punkt [2.5.1](#) i denne rapporten viser vi til diskusjonen gruppen har hatt med bedriften om utviklingen av utbetalingsløsningen, og hvordan det er kommet frem til den nåværende løsningen; en manuell utbetaling fra bedriften.

Oppdragsgiver behøvde en løsning som ville gi oversikt over hvor mye som er utbetalte, hvor mye som er tjent, hvilken periode utbetalingen tilhører, og hvilke utbetalinger som ikke er betalt enda. For å oppnå dette er det valgt en status-basert oversikt.

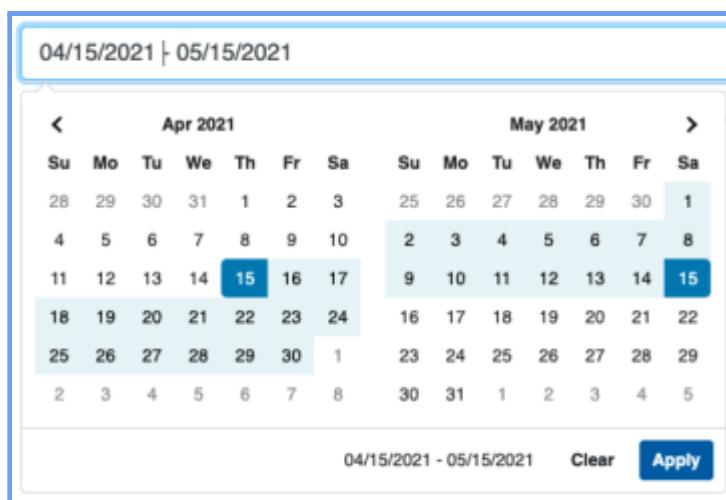
Dersom en bruker gjør et salg på ett av sine dokumenter den 17. januar, og ikke har gjort noen salg siden før den 15. januar, vil en ny utbetaling bli opprettet, med statusen "Utbetalt". Utbetalingen inneholder informasjon om hvor mye som skal utbetaltes til selgeren, og hvor mye bedriften kommer til å tjene på utbetalingen. Tilleggsinformasjon er også tilgjengelig for administratorer, som for eksempel datofristen for utbetaling. Datoen hvor utbetalingen er sist oppdatert lagres også. Dette er fordi dersom den samme brukeren gjør et nytt salg den 24. januar, og systemet finner utbetalingen som er opprettet, har statusen "Utbetalt" og datoen i dag er før fristen (15. februar), så vil den eksisterende utbetalingen sine beløper bli oppdatert.

Tilbake								Administratorpanel	Min Way2Easy	Logg ut
04/15/2021 - 05/15/2021										
Brukere	Netto (tar frakt)	Kroner som skal utbetales	Fortjeneste	Først opprettet	Sist oppdatert	Frist		Status		
Tester	48	36	12	onsdag 5. mai 2021	onsdag 5. mai 2021	lørdag 15. mai 2021	<button>Uten betaling</button>	Utbetalt		
Fakesome	22	16	6	onsdag 5. mai 2021	onsdag 5. mai 2021	lørdag 15. mai 2021	<button>Uten betaling</button>	Utbetalt		

Figur 36, viser oversikt over utbetalinger som enda ikke er betalt. Utbetalingerne er blitt automatisk hentet fra perioden 15. April til 15. Mai.

Etter at logikken for oppretting, oppdatering og behandling av utbetalinger var opprettet, ble siden /Admin/PendingPayments opprettet for å gi oppdragsgiver en oversikt over alle utbetalingerne som er opprettet. Når en i bedriften først kommer på denne siden henter systemet alle de førstkomende utbetalingerne innen perioden "forrige måned til neste måned", altså for eksempel 15. April til 15. Mai, dersom datoan i dag er før 15. Mai.

Siden vil også fungere som en oversikt for regnskapsformål, og administratorer kan derfor klikke på dato-boksen for å velge til- og fradatoen slik de selv ønsker, slik at de kan få en fullstendig oversikt over alle utbetalinger gjort tidligere.



Figur 37, En Til- og Fradatovelger som lar oppdragsgiver hente tidligere utbetalinger, dette er ment å fungere som en regnskapsløsning i tillegg som en utbetalingsoversikt.

Videre kan bedriften se oversikt over hver enkel utbetaling i detalj, og enten markere statusen som "Betalt" eller "Mislykket" etter at de har gjort en manuell utbetaling til brukerens personlige bankkonto. Dette bankkontonummeret må brukere oppgi før de kan laste opp dokumenter for første gang. Utbetalingen gjøres i bedriftens valgte tredjeparts banksystem. Etter at statusen på utbetalingen er endret, kan det dersom det trengs, reverseres tilbake til "Ubetalt".

Systemet sorterer listen av utbetalinger slik at de utbetalingerne som er mislykkede alltid kommer øverst, deretter kommer de som har status "ubetalt", og til sist de som er betalt.

Tilbake								
03/10/2021 - 05/19/2021								
Brukernavn	Netto (før trekk)	Kroner som skal utbetales	Fortjeneste	Først opprettet	Sist oppdatert	Frist		Status
Fakesome	22	16	6	onsdag 5. mai 2021	onsdag 5. mai 2021	lørdag 15. mai 2021	Utbetalt	Mislykkes
Tester	48	36	12	onsdag 5. mai 2021	onsdag 5. mai 2021	lørdag 15. mai 2021	Utbetalt	Utbetalt
Tester	20,-	15,-	5,-	mandag 5. april 2021	fredag 5. februar 2021	torsdag 15. april 2021	Se detaljer	Betalt

Figur 38, En av utbetalingerne ble mislykkes og en epost er blitt sendt til brukeren hvor de bes ta kontakt med Way2Easy så fort som mulig. En utbetaling med statusen "Betalt" vises også, og blir lagt nederst i tabellen.

En annen situasjon som kan oppstå er at en utbetaling sin frist har utgått. I dette tilfellet vil denne utbetalingen bli inkludert før ubetalte utbetalinger i tabellen, i tillegg til å utheve den med rød skrift i frist-feltet. Dette skjer selv om til- og fra datoene som er valgt ikke stemmer datoene for denne utbetalingen.

04/15/2021 - 05/15/2021								
Brukernavn	Netto (før trekk)	Kroner som skal utbetales	Fortjeneste	Først opprettet	Sist oppdatert	Frist		Status
Fakesome	22	16	6	onsdag 5. mai 2021	onsdag 5. mai 2021	lørdag 15. mai 2021	Utbetalt	Mislykkes
Tester	20	15	5	fredag 5. mars 2021	fredag 5. mars 2021	mandag 15. mars 2021	Utbetalt	Utbetalt
Tester	48	36	12	onsdag 5. mai 2021	onsdag 5. mai 2021	lørdag 15. mai 2021	Utbetalt	Utbetalt

Figur 39, en utbetaling har ikke blitt betalt innen fristen, som var 15. mars 2021. Derfor blir den inkludert selv om de valgte til- og fradatoene er fra 15. april til 15. mai.

4.2.3.8 Side for å endre roller på nettsiden

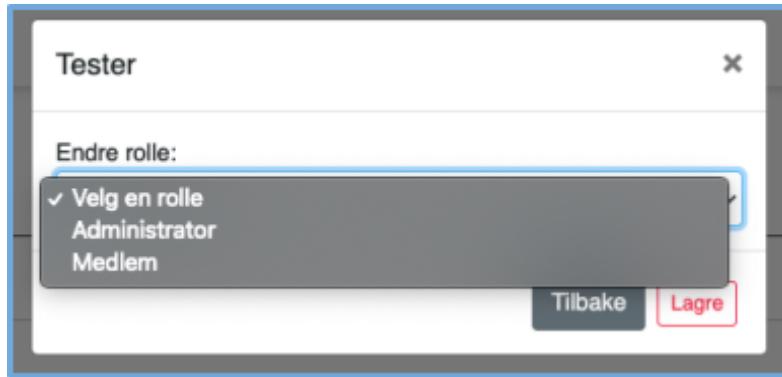
For å sikre sensitive data, og passe på at ingen uautoriserte brukere får tilgang på administratorsider er det tatt i bruk rollebasert autorisering.

The screenshot shows a web interface for managing user roles. At the top, there's a navigation bar with links like 'Hjem', 'Way2Easy.no', 'FAQ', 'Søg', 'Handlekurv', and a user icon. On the right, there are links for 'Administratorpanel', 'Min Way2Easy', and 'Logg ut'. Below the navigation, there's a search bar with placeholder text 'Søk etter brukernavn.' and a 'Søk etter bruker' button. The main content area is titled 'Rolle og rettigheter' and contains a table with the following data:

Brukernavn	Email	Rolle	Aksjoner
Admin	simenrenberg@gmail.com	Administrator	<button>Redigér</button>

Figur 40, side hvor administratorer kan registrere og oppdatere roller på nettsiden.

Dersom det er behov for administratorer på nettsiden, kan eksisterende administratorer legge inn en nøyaktig søkestreng på navnet til en eksisterende bruker, og dersom systemet finner brukernavnet de søker etter kan de gi brukeren rollen "Administrator". Det er også mulig å endre en eksisterende administrator tilbake til en normal bruker.



Figur 41, søkerstrengen fant brukeren "Tester", og administratoren kan nå redigere "Tester" sin rolle på nettsiden.

4.2.3.9 Side for å oppdatere kommisjonsprosent

Bedriften sin hovedinntektskilde er en prosentvis godtgjørelse på hvert salg gjort på siden, denne godtgjørelsen har vi definert som "kommisjonsprosenten". Bedriften vil kunne endre hvor stor/liten andel av salget de tar, da de vil tilby en gunstigere

salgsgevinst for selgerne på siden tidlig i prosjektet for å oppnå et godt antall aktive brukere. Deretter vil bedriften ha mulighet til å justere kommisjonsprosenten opp igjen etter at en stabil trafikk på siden er oppnådd. Det er også ment at denne funksjonaliteten skal kunne gi bedriften muligheten til å ha enkelte måneder hvor de justerer kommisjonsprosenten som en slags kampanje, hvor selgere får en større andel av salgsgevinsten enn til vanlig.

4.2.3.10 Side for å oppdatere prisen det koster å promotere et dokument

En annen inntektskilde for bedriften er betaling for promotert plass for dokumenter på siden. Årsaken til at oppdragsgiver ønsker muligheten til å endre prisen det koster å promotere et dokument er at det ikke er fastsatt hvilken sum som vil gjøre at nok brukere velger å promotere sine dokumenter, uten at det å promotere dokumentene er et selvfølge. Oppdragsgiver ønsker med andre ord at markedet skal være balansert nok til at man kan promotere dokumenter uten at det å promotere et dokument er den eneste måten å få dokumentet til å selge.

4.2.4 Andre sider

Til slutt vil vi i dette underkapittelet gå gjennom systemets komponenter/sider som ikke tilhører en spesifikk type bruker, men fortsatt bør redegjøres.

4.2.4.1 _layout.cshtml

Dette er en av de automatisk genererte “partial-views”-ene som blir opprettet av visual studio under opprettingen av prosjektet. _layout.cshtml er en server-generert HTML-fil som inkluderes i alle andre HTML-filer, så dersom noe skrives i denne filen vil dette inkluderes i alle andre filer på nettsiden. Det er i denne filen hvor cookie-varslingen er opprettet, for å passe på å opplyse brukere om sidens cookie-policy uansett hvor på siden de befinner seg. Et annet element som alltid er synlig er headeren.

4.2.4.1.1 Varsel om bruk av informasjonskapsler (cookies)

Ved første besøk på nettsiden, vil en stor del av siden dekkes av en cookie-varsling. Ved bruk av informasjonskapsler på nettsider stilles det visse krav for håndteringen

av samtykke. Nasjonal kommunikasjonsmyndighet sine anbefalinger om bruk av informasjonskapsler er som følger:

I 2013 ble det tatt inn en ny bestemmelse i ekomloven om vilkårene knyttet til lagring av opplysninger i kommunikasjonsutstyr - den såkalte «cookie paragrafen» i § 2-7b. Bestemmelsen er en lovfesting av det som tidligere var regulert i ekoforskriften § 7-3, - men i noe utvidet og endret form.

§ 2-7b Bruk av informasjonskapsler/cookies

Lagring av opplysninger i brukers kommunikasjonsutstyr, eller å skaffe seg adgang til slike, er ikke tillatt uten at brukeren er informert om hvilke opplysninger som behandles, formålet med behandlingen, hvem som behandler opplysningene, og har samtykket til dette. Første punktum er ikke til hinder for teknisk lagring av eller adgang til opplysninger:

1. *utelukkende for det formål å overføre kommunikasjon i et elektronisk kommunikasjonsnett*
2. *som er nødvendig for å levere en informasjonssamfunnstjeneste etter brukerens uttrykkelige forespørsel.* (Nkom, 2020)

Vi bruker informasjonskapsler (cookies)

Vår nettside benytter informasjonskapsler (cookies). Dersom de ikke er nødvendig for at nettsiden vår skal fungere, vil de ikke lagres på din enhet med mindre du samtykker til dette. Les om hvilke vi bruker og hvordan vi administrerer dem [her](#).

Nødvendige cookies

Disse blir brukt for å oppnå essensiell funksjonalitet på nettsiden. Uten disse ville ikke funksjonalitet som innlogging og handlekurv fungert. Vi har vurdert disse som nødvendige, og de lagres derfor uten foregående samtykke.

Andre cookies

Vi benytter oss ikke av noen cookies annet enn de som er nødvendige for funksjonalitet. Vi lagrer ingen ting av din oppførsel og bruk av vår nettside for analysering og reklame.

[Jeg forstår](#)

Figur 42, popup-vinduet som ber brukere av siden om å akseptere Dokumarked sin cookie policy. Ingen cookies på siden brukes til andre formål enn at siden skal fungere.

4.2.4.1.2 Header

En header er den globale navigasjonsmenyen til en nettside. Denne ligger plassert øverst på nettsiden og skal være synlig fra alle sider. Vår header har såkalt “sticky” posisjon, som vil si at dersom man blar nedover på siden så vil headeren følge etter og alltid ligge øverst på skjermen.

Figur 43, headeren til siden dersom brukeren ikke er innlogget.

Mens en anonym bruker er på siden vil headeren inneholde syv lenker og en knapp for å skru av/på dark mode. Dark mode forklares i punkt [4.3.3.2](#). Disse syv lenkene leder (fra venstre til høyre) til forsiden, way2easy sin bedriftsside, ofte stilte spørsmål, siden for å laste opp et nytt dokument, handlekurven, siden for å registrere seg, og siden for å logge inn.

Figur 44, headeren til siden dersom brukeren er innlogget

Dersom brukeren er innlogget vil lenken som sender brukeren til registreringssiden være byttet ut med en lenke som tar brukeren til deres private profil, "Min Way2Easy", som forklares i punkt [4.2.2.3](#). Lenken som tar brukeren til innloggingssiden blir byttet ut med en lenke som logger brukeren ut.

Figur 45, headeren til siden dersom brukeren som er innlogget er en administratorbruker

Dersom brukeren som er innlogget har administrator-rolle, så vil en åttende lenke dukke opp i headeren, som vil ta de til administratorpanelet. All funksjonaliteten som er implementert i administratorpanelet forklares i punkt [4.2.3](#). Administratorer skal også kunne bruke siden som normalt, så de har også en lenke til sin egen private profil, akkurat som vanlige innloggede brukere.

4.3. Design, UI og UX

Fra starten av utviklingen var oppdragsgiver klar over at vi som utviklere og teknologistudenter har kunnskap og erfaring i å designe et behagelig brukergrensesnitt. Vi fikk derfor frihet til å ta mange av avgjørelsene om designet. Under de ukentlige møtene fikk vi feedback om hva de likte og hva de gjerne ville endre litt på, men som oftest var de veldig fornøyde med våre valg og lot oss fortsette med det vi syntes var de beste løsningene. I kapitlene under går vi dypere

inn på hvilke valg vi har gjort for å skape et godt brukergrensesnitt og en god brukeropplevelse.

4.3.1 Design

Gruppen startet med planlegging av nettsidens utseende og brukergrensesnitt tidlig i prosessen. Ved bruk av figma ble det utviklet en rekke udetaljerte skisser og wireframes som ble presentert for bedriften. Disse ble deretter implementert på nettsiden med kode. I det ferdige produktet er det store forskjeller fra skissene og implementert kode. Figur 46 viser forskjellen fra første skisse for visning av et dokument og ferdig design.

Bedriften hadde 3 kriterier for utseendet og brukeropplevelsen av siden. Nettsiden skulle:

1. Se "bra" (i dette tilfellet definert ut fra oppdragsgivers ønsker) og moderne uten at det går utover funksjonaliteten
2. Være enkel å navigere
3. Være universelt utformet og responsivt designet for å sikre at brukere på forskjellige enheter får en like god opplevelse

Forretningsjus - Komplett sammendrag for 2020 - brukernavn	A	19.01.2021	Prototyping
Karakter	OsloMet	MET 21233	Høgskulen i Volda
Skole	JUR3420	Minst 100%	DAFE1300
Fagkode	Sammendrag	Sammenligning av 10 artikler	SAMFUNN
Oppgavetype	100,-	Sammenligning av 10 artikler	Bacheloroppgave
Pris		32 sider	11 sider
Se mer			Vår, 2020

testbruker 45,-

★★★★★

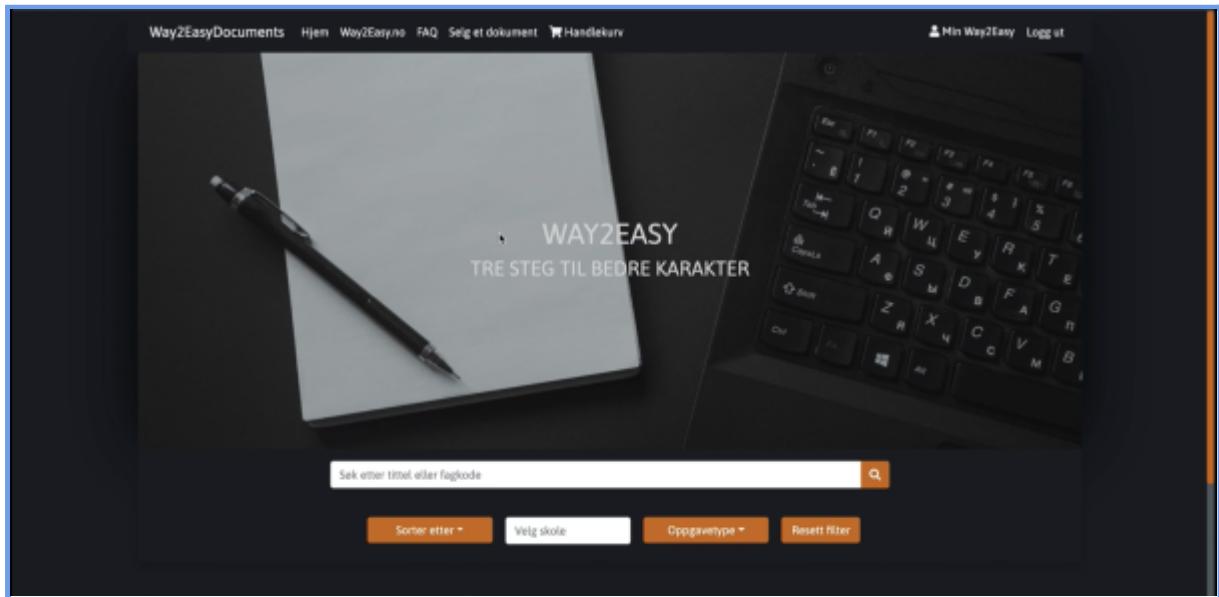
Min bacheloroppgave fra 2018. Oppdatert med riktige lover for 2020. Spesielt nyttig på hjemmeeksamen siden at du kan enkelt søke opp hvert tema. Hjelp meg veldig til å sikre en A i pototyping.

[Les mer & kjøp](#)

Figur 46, Første skisse for dokumentvisning til venstre. Ferdig dokumentvisning til høyre.

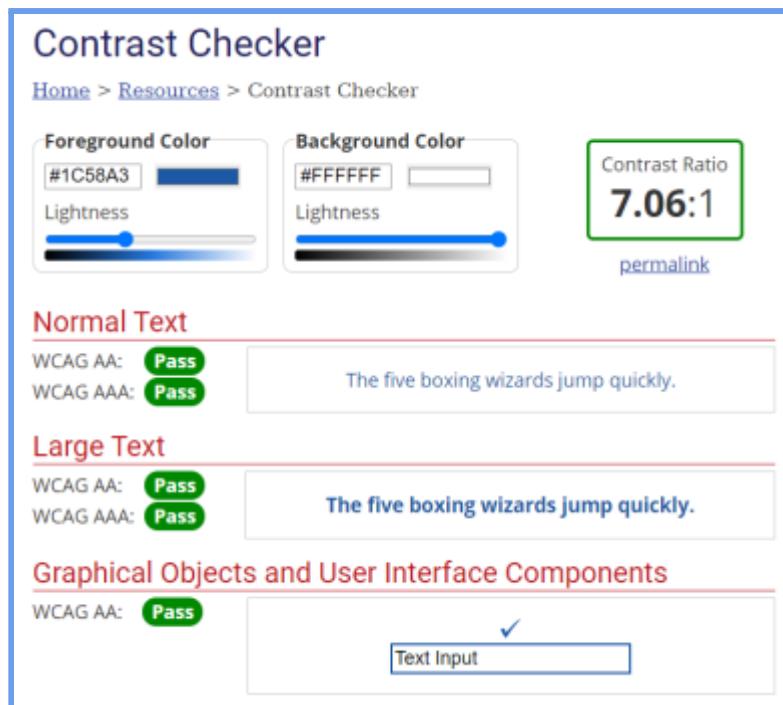
4.3.1.1 Farger

Når vi mottok oppgaven hadde ikke bedriften diskutert hvilke farger de ville ha på siden. Tidlig i utviklingsprosessen tok vi i bruk et svart og oransje fargetema, men etter tilbakemelding fra bedriften valgte vi å gå bort fra dette og mot et ”lysere” tema.



Figur 47, Skjermbilde av tidlig versjon av nettsiden.

Blåfargen som brukes som primærfargen på nettsiden ble bestemt sammen med bedriften. Bedriften ble presentert en rekke farger med forskjellig metning, valør og fargetone. Vi forsikret oss om at alle fargene de ble presentert med fulgte kontrast-kravene for tekst og andre elementer fra WCAG 2.1-standarden. WCAG 2.1 er en rekke retningslinjer og anbefalinger som er utviklet for å gjøre brukergrensesnitt mer tilgjengelig for folk med funksjonshemninger (W3, 2018). For å sjekke om fargene tilfredsstilte kravene brukte vi nettsiden WebAIM som sjekker kontrast-forholdet mellom bakgrunnsfargen og forgrunnsfargen.



Figur 48, Testing av kontrastforholdet mellom bakgrunnsfargen og forgrunnsfargen som brukes på nettsiden

Vi ble enige med bedriften om en blåfarge (heksadesimal fargekode #1C58A3) som kontrasterte nok med bakgrunnen, men som også var lys nok til å gi det positive og beroligende inntrykket til sluttbrukeren som bedriften ønsket. Blå forbides ofte med positivitet, kompetanse og troverdighet hos mennesker (*Bertram, Patra & Stone, 2017, S. 2*).

For å forsikre oss om at nettsiden er brukbar og tilgjengelig for alle ble det bestemt at nettsiden skulle være monokrom, og at vi heller skulle bruke forskjellige nyanser av primærfargen for å markere funksjonalitet og interaktivitet. Vi unngikk å bruke farger som eneste indikator for funksjonalitet, slik at det ikke er noe på siden som personer med nedsatt fargesyn ikke har mulighet til å benytte seg av. Et eksempel på dette ville vært å bruke rød og grønn som indikatorer for bekrefte og avkrefte. I disse tilfellene ble det heller brukt tydelig tekst med enten forgrunnsfargen på bakgrunnsfargen eller motsatt.

4.3.2 UI

UI (User Interface) eller brukergrensesnitt på norsk vil si det grafiske uttrykket som brukeren interagerer med. Med andre ord så er brukergrensesnittet det man kan se.

Dette inkluderer nettsidens innhold som tekst og bilder, skjemaer som skal fylles ut og knapper som kan trykkes. (*UX Og UI Design - Hva Er Egentlig Forskjellen?*, 2019)

4.3.2.1 Ikoner

Vi har brukt ikoner forskjellige steder på nettsiden der vi følte at det hjalp med forståelsen av funksjonaliteten ikonet representerer. Bruk av ikoner kan gjøre brukergrensesnittet enklere å forstå. Mennesker har nesten en perfekt evne til å gjenkjenne bilder. Sammenlignet med tekst går det mye forttere å se og forstå et ikon. Dersom brukeren har lite erfaring med nettsider generelt kan ikonene virke misledeende og være vanskelig å forstå, men målgruppen til Way2Easy er elever og studenter, og de har som regel god erfaring med nettsider og applikasjoner.

(F.E.Sandnes, *personlig kommunikasjon*, 2019, 09-ikoner.pdf, s.2-3).

Ikon	Forklaring
	Handlekurv. Lett gjenkjennelig. Brukes ofte på nettsider hvor kjøp kan gjøres.
	Et menneske. Ofte brukt for å symbolisere profil. Brukt hos oss i navigasjonsmenyen i sammenheng med min way2easy.
	Penn og papir. Brukt på nettsiden ved siden av "rediger profil" inne på min way2easy.
	Forstørrelsesglass. Fungerer som søkeknappen. Brukt på utallige nettsider og har blitt standarden for søk på nett.
	Darkmode knapp. Blir brukt på flere nettsider som tilbyr darkmode. Siden ikonet ikke nødvendigvis er selvforklarende så har det et tilsvarende

	tooltip dersom brukeren holder musen over.
≡	Hamburgermeny. Navigeringsmenyen på små skermstørrelser blir lagt inne i denne. Veldig mye brukt på samtlige nettsider.

4.3.2.2 Skrifftype

Skriftypen på en nettside er veldig viktig for utseendet og helhetsinntrykket. Dette gjelder særlig når det er snakk om en tekstbasert nettside med lite grafisk innhold. Vi har derfor brukt en del tid og ressurser på valget av skriftypen.

Når vi startet utviklingen var det ikke bestemt hvilken skrifftype som kom til å brukes på det endelige produktet. Vi testet derfor ut en rekke forskjellige underveis for å forsikre oss for at nettsiden fortsatt ser lik ut og fortsatt er responsiv når vi bytter til den endelige skriftypen.

Da vi hadde implementert det meste av funksjonaliteten i kravspesifikasjonen og startet å finpusse designet og stylingen på siden bestemte vi oss sammen med bedriften om en skrifftype. Gruppen fant 3 skriftyper de likte og som de syntes passet sammen med nettsidens design og formål; “Roboto”, “Montserrat” og “Nunito”. Etter diskusjon fra bedriften ble vi enige om å bruke “Nunito” (Figur 51).

Almost before we knew it, we had left the ground.

Figur 49, Skriftypen “Roboto”. En av skriftypene vi vurderte å bruke på nettsiden.

Almost before we knew it, we had left the ground.

Figur 50, Skriftypen “Montserrat”. En av skriftypene vi vurderte å bruke på nettsiden.

Almost before we knew it, we had left the ground.

Figur 51, Skriftypen "Nunito". Det er denne vi ble enige om å bruke på nettsiden.

4.3.3 UX

UX (User experience) eller brukeropplevelse på norsk handler om hvordan brukeren opplever det å interagere med brukergrensesnittet. Et godt brukergrensesnitt fører ofte til en god brukeropplevelse, men det er også noen grep man kan ta for å få det lille ekstra som forsterker opplevelsen.

4.3.3.1 Mikrointeraksjoner

En mikrointerasjon er en hendelse som har et enkelt mål, nemlig å skape et øyeblikk som oppleves behagelig eller hjelpsomt for brukeren. (Batchu, 2018). Mikrointeraksjoner vi har tatt i bruk beskrives under.

1. Tooltips

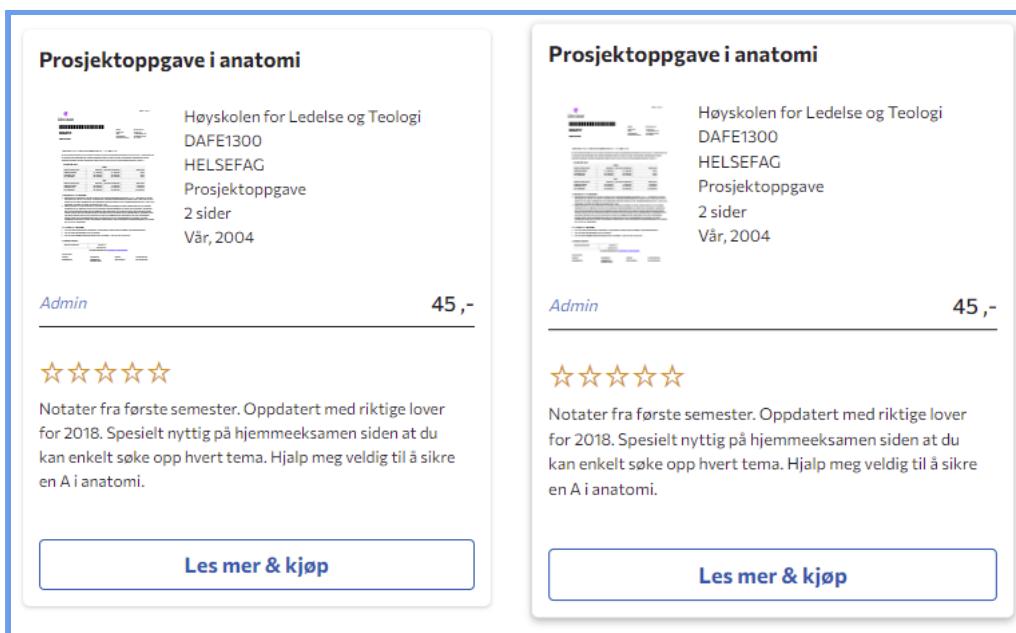
Dersom brukeren holder musen over visse elementer i brukergrensesnittet vil et tooltip (forklarende melding) vises. Tooltip-et forklarer hva brukeren faktisk holder musen over. På bildet under ser man hvor tooltips er tatt i bruk. Det er ikke åpenbart hva disse elementene faktisk gjør, så en forklarende tekst kan være til stor hjelp. For folk med synsutfordringer som bruker screen readers ville det faktisk vært umulig å vite hva disse knappene gjør uten en tooltip. En screen reader vil vanligvis lese opp tekstinneholdet til knappen, men knappene under har ingen tekst og dermed vil en skjermleser lese tekstinneholdet til tooltip-et.



Figur 52, Forskjellige tooltips på nettsiden som forklarer symbolene på siden.

2. Animasjoner på dokumentene

Når brukeren holder musen over et dokument vil tilsvarende dokument bli gjort større og få en mørkere skygge. Dette skaper en effekt som minner om det å faktisk plukke opp dokumentet for å ta en nærmere titt på det.



Figur 53, Viser forskjellen på et dokument når musen holdes over.

3. Fargeendring ved berøring

På flere av elementene på siden skjer det en fargeendring når brukeren holder musepekeren over. Denne fargeendringen er veldig subtil, men den er med på å separere statiske og interaktive elementer på siden. Denne

mikrointeraksjonen blir brukt på alle knapper og lenker. Knapper som allerede har bakgrunnsfargen får forgrunnsfargen ved berøring (se figur 55).



Figur 54, Viser forskjellen hvordan knapper ser ut til vanlig (til venstre) og når musen holdes over (til høyre)



Figur 55, Viser forskjellen hvordan knapper med hvit bakgrunn ser ut til vanlig (til venstre) og når musen holdes over (til høyre)

4.3.3.2 Dark mode

Under diskusjoner med medlemmene i Way2Easy ble det bestemt at dark mode skal utvikles for nettsiden. Dark mode er et alternativt design for en applikasjon hvor de lyse fargene blir gjort mørke. De siste årene har darkmode blitt meget populært innen design av applikasjoner og nettsider. Et mørkt fargetema er mer behagelig for øynene dersom det ikke er mye lys der man er eller om man sitter flere timer daglig foran skjermen. Det å gi brukeren kontroll over egen preferanse forbedrer også brukeropplevelsen. (Eisfeld & Kristallovich, 2020).

Dark mode funksjonaliteten ble utviklet ved hjelp av css custom properties. Variabler i css er gunstig dersom man skal bruke samme styling, som for eksempel farger, flere steder. I vårt tilfelle bruker vi en blåfarge mange steder. Vanligvis ville man skrive følgende css kode for å få en farge på et element:

```
.element {  
    background-color: [vår fargekode];  
}
```

Figur 56, Eksempel på css kode for å fargelegge et element

Dette fører til veldig mye duplisert kode, noe man ikke ønsker. Dersom man ønsker å endre fargen så må man endre fargekoden alle steder den er brukt. Det er her css custom properties er gunstig å bruke. For å gjøre eksemplet over på en bedre måte kan man gjøre følgende:

```
:root {  
    --main-secondary-color: [vår fagekode];  
}  
.element {  
    background-color: var(--main-secondary-color);  
}
```

Figur 57, Eksempel på css custom properties.

Dersom man nå ønsker å endre fargen trenger man kun å endre fagekoden i variabelen i root. Når dark mode knappen trykkes blir javascript brukt for å legge til en klasse kalt darkmode på <body> elementet. Denne darkmode klassen endrer verdien på variablene i root. Følgende eksempel viser css kode for å endre en lys farge til mørk:

```
:root {  
    --main-background-color: [vår lyse farge];  
}  
  
.darkmode {  
    --main-background-color: [vår mørke farge];  
}  
  
.element {  
    background-color: var(--main-background-color);  
}
```

Figur 58, Eksempel på bruk av css custom properties for dark mode funksjonalitet.

Eksempelet over viser at elementet alltid bruker den samme variabelen, men verdien endres ut ifra om <body> elementet har klassen darkmode eller ikke.

Bildene under viser forskjellen på innloggingssiden med og uten dark mode.

The screenshot shows a standard light-colored login form. At the top right are links for 'Registrer' and 'Logg inn'. Below that is the title 'Logg inn' and a subtitle 'Selg og kjøp studiemateriale, og del din kunnskap.' A horizontal line separates this from the input fields. The first field is 'Brukernavn' with the value 'testbruker'. The second field is 'Passord' with a masked value. There is a checked checkbox for 'Husk meg?'. Below the fields are buttons for 'Logg inn', 'Glemt passord?', and 'Registrer deg her'. Another horizontal line follows, with the text 'Logg inn med Facebook eller Google?' and two buttons for 'Facebook' and 'Google'. At the bottom left is a copyright notice: '© 2021 - Way2EasyDocuments - Privacy'.

Figur 59, Innloggingssiden uten dark mode.

The screenshot shows the same login form but with a dark background. The title 'Logg inn' and subtitle 'Selg og kjøp studiemateriale, og del din kunnskap.' are at the top. The input fields ('Brukernavn' and 'Passord') have black backgrounds with white text. The 'Husk meg?' checkbox is checked. The buttons for 'Logg inn', 'Glemt passord?', and 'Registrer deg her' are white with black outlines. The social media integration section at the bottom has a white background with black text and buttons. The copyright notice at the bottom left is also present.

Figur 60, Innloggingssiden med dark mode.

4.3.4 Universell utforming

Under design av nettsiden har universell utforming vært et fokus. I flere av delkapitlene diskutert under punkt [4.3](#), som valg av farger, kontrast, dark mode og mikrointeraksjoner har vi redegjort at mange av valgene tatt er gjort for å oppnå en universell utforming. Vi vil i dette underkapittelet gå igjennom de resterende

designelementene vi har benyttet oss av for å gjøre siden universelt utviklet. Noen av punktene som allerede er diskutert i punkt [4.3](#) vil dukke opp her også, fordi en konkret oppsummering av hvorfor de er universelt utformet er verdt å nevne.

4.3.4.1 Dark mode

Høykontrastmodus, eller “dark mode” er som allerede nevnt i punkt [4.3.3.2](#), men i det punktet har diskusjonen vært sentrert rundt hvordan det er blitt implementert. For å forklare godene dette designet gir i kontekst av universelt design vil vi referere til dette sitatet fra Angeles (2019):

Dark mode is not only beneficial for users with temporary or permanent light sensitivity (e.g. dilated pupils after an eye exam, or people suffering from photophobia), but is also more comfortable for non-visually impaired users, particularly when in low-light environments.

Med tankegangen presentert av Angeles gjøres det tydelig at valget om å tillate brukere å aktivere dark mode på nettsiden gjør nettsiden helhetlig mer tilgjengelig for både brukere med og uten f. eks. en synshemning.

4.3.4.2 Tabindex

Det er ikke alle brukere som har mulighet for å navigere nettsider på den tradisjonelle måten med bruk av musepeker. En rekke forskjellige grunner til dette eksisterer. Noen brukere kan være blinde, andre kan av forskjellige årsaker ikke ha mulighet til å operere en mus. På nettsiden har vi implementert såkalt “keyboard navigation” gjennom vår bruk av tabindex, som vil bestemme i hvilken rekkefølge elementer på nettsiden blir valgt/uthevet når brukeren trykker på “innrykks-knappen” (oftest kalt tab) på deres tastatur. På denne måten har vi som utviklere kunne prioritert hvordan brukerens flyt på nettsiden vil styres, slik at mer relevante knapper og lenker blir prioritert over andre, mindre viktige elementer.

4.3.4.3 Tooltips

Gruppen har ikke villet anta at alle personer vil få det samme inntrykket eller forståelse av et ikon på nettsiden. Vi har valgt å benytte oss av konvensjoner når det kommer til valg av ikoner for å minimere sjansene for at en bruker vil møte på et ikon

som har en betydning de ikke enda er kjent med. Likevel har vi passet på å inkludere såkalte tooltips som forklarer hva et hvert ikon betyr.

4.3.4.4 Aria labels og alt-tekst

Aria labels er tekst som er skjult for brukerne av nettsiden, men er tilgjengelig for skjermlesere for å kunne gi personer som er avhengige av dette en måte å lese hva knapper og ikoner uten tekst gjør. Eksempler på elementer på dokumarked med Aria Labels er modaler (en slags pop-up) som kan varsle om hendelser på siden, eller gi brukere et skjema til å taste inn for eksempel omtaler for et dokument. For å passe på at brukere som ikke kan se at denne modalen popper opp fortsatt får med seg dette har disse modalene et aria label som forklarer hendelsen.

Alt-tekst står for alternativ tekst. Dette er tekst vi som utviklere kan inkludere i bildene på nettsiden som vil leses opp av skjermlesere. Hvis bildet ikke lastes på grunn av treg internettforbindelse eller andre årsaker vil den alternative teksten for bildet vises i stedet.

5. Diskusjon om faglige utfordringer og valg

Under både planleggingen og utviklingen av systemet møtte vi på flere utfordringer og valg vi måtte ta. Dette gjelder både tekniske utfordringer angående implementering av funksjonalitet og design, samt mer bedriftsrettede valg. I kapitlene under går vi mer i detalj om hvordan vi gikk frem for å takle disse utfordringene og hvilke valg vi har gjort.

5.1 Fremgangsmåte når utfordringer oppstår

Vår fremgangsmåte for å takle utfordringer vi møtte på var følgende:

1. Anerkjenne utfordringen.
2. Diskutere innad i gruppen.
3. Komme med mulige løsninger.
4. Diskutere utfordringen og mulige løsninger med oppdragsgiver, samt få feedback om oppdragsgivers ønsker.
5. Implementere løsningen.

Ved bruk av denne fremgangsmåten ble vi aldri sittende fast på et problem i lang tid. Dersom et gruppemedlem hadde en utfordring kunne han alltid få hjelp av resten av gruppen og eventuelt finne en alternativ løsning i samarbeid med oppdragsgiver.

5.2 Viktige valg som har blitt gjort under oppbygging av systemet

5.2.1 Endring av rammeverk

Under planleggingsfasen av prosjektet ble det bestemt at gruppen skulle utvikle løsningen gjennom frontend rammeverket React. Det aller første utkastet av systemet er faktisk laget med React. Kun to av fire medlemmer på gruppen hadde noe erfaring med React så vi fant fort ut at det å lære seg denne teknologien ved siden av det å utvikle systemet var alt for tidkrevende og rett og slett vanskeligere enn forventet. Vi besluttet derfor å bytte ut React-rammeverket med Dotnet Core, som samtlige studenter i gruppen hadde erfaring med. I tillegg ville systemets arkitektur ligne mer på en tradisjonell nettside, noe som ville gjøre det lettere for et mindre erfarent medlem i gruppen.

5.2.2 Endring av database

Gruppen og bedriften ble tidlig enige om å bruke smarterASP.net som host. SmarterASP er et langt billigere alternativ enn Azure som er microsoft sin vertstjeneste for systemer laget i ASP.NET. Det var ikke før langt inn i utviklingen vi fant ut at smarterASP ikke støtter SQLite databaser, som var det vi hadde brukt. Vi måtte derfor overføre hele systemet til MSSQL og gjøre nødvendige endringer for å gjøre det kompatibelt med MSSQL.

5.3 Tekniske utfordringer

I løpet av vår utdanning ved OsloMet har vi bygget grunnlaget til å utvikle mindre systemer. Dette prosjektet er den første gangen gruppen har måttet kombinert all denne kunnskapen for å bygge et fullstendig og komplekst system. Dette medfører at for å kunne produsere kode med høy nok kvalitet var det nødvendig for gruppen å

tilegne seg ny kunnskap om både teknologiene vi brukte, og logikken i systemet vi utviklet. I dette underkapittelet vil vi gå gjennom de punktene som var mest teknisk utfordrende og forklare hvordan og hvorfor vi kom frem til løsningene vi gjorde.

5.3.1 Preview av dokument

Det å vise et bilde av dokumentet på forsiden viste seg å være langt vanskeligere enn forventet. Om det hadde vært et faktisk bilde av dokumentet i .jpeg eller .png format som brukeren lastet opp ved siden av .pdf filen ville det vært enkelt, men gruppen ble enige om at å lagre så mange bilder ville ta veldig mye lagringsplass og kunne fort bli dyrt for bedriften.

Vi benyttet oss derfor av en javascript plugin for å forhåndsvise pdf dokumenter kalt pdf.js. Denne pluginen bruker et <canvas> html element istedenfor et element. Canvas var noe vi ikke hadde erfaring med fra før og viste seg å ha en del uforventede egenskaper. Det å gjøre dette elementet responsivt, altså å få det til å justere seg etter nåværende skjermstørrelse var mindre intuitivt enn vi først antok. Disse årsakene gjorde at denne relativt enkle funksjonen tok lengre tid å implementere enn antatt.

Etter manuell testing av filtrering, søk og sortering fant vi ut at dersom en bruker trykker på ett filtrerings alternativ gjentatte ganger innen kort tid vil forhåndsvisningen av dokumentet vises flere ganger under hverandre. Vi brukte lang tid på å prøve å finne ut årsaken til dette og å finne en fiks. Dokumentasjonen til pdf.js gjorde oss ikke noe klokere, og ingen på utvikler-forumer hadde møtt på samme problem. Det endte med at vi aldri fant grunnen til problemet og bare antok at det var noe i kodebasen til pdf.js som førte til dette. Vi fikset problemet ved å deaktivere filtreringsmulighetene etter en filterknapp er trykket frem til dokumentene er vist. Dette gjør så man ikke kan trykke samme knapp flere ganger for å gjenskape problemet.

Figur 61, Filtreringsmuligheter med deaktiverte knapper.

5.3.2 Filtreringskombinasjoner

Etter alle filtrering, søk og sorteringsmuligheter var ferdig utviklet og testet, fant vi ut at ikke alle mulighetene var kompatible med hverandre. For eksempel dersom man søkte på noe og deretter valgte å kun vise dokumenter med sammendrag som oppgavetype, ville oppgavetype filteret overstyre søkeret. Dette førte til et unaturlig bruksmønster når det gjaldt å finne dokumenter.

Utfordringene kom fordi vi har såpass mange alternativer man kan velge, nemlig sorter, oppgavetype, studieretning, studiested, tidslinje og utdanningsnivå. Alle disse skal kunne velges og fungere med hverandre. Det tok lang tid med debugging og å prøve forskjellige løsninger før vi fikk løst problemet. Den endelige løsningen viste seg å være en del av LINQ som vi ikke var kjent med fra før. Måten vi hadde lært å bruke LINQ fra faget Webapplikasjoner vises i eksempelet under.

```
_db.Dokumenter.ToList();
```

Figur 62, Eksempel for å hente alle dokumenter i databasen.

Bruk av LINQ på denne måten førte til at algoritmen for sortering, filtrering og søker inneholdt en if-test for om hver enkelt filtreringsmulighet var valgt eller ikke. Dette førte til en algoritme på 240 linjer kode som var lite effektiv. Etter lesing av microsoft sin offisielle dokumentasjon om LINQ fant vi ut av det er mulig å skrive spørrengene som variabler av type IQueryable<T>. Disse spørrengene kan konkateneres til å danne en fullstendig spørring som inneholder alle valgte filtreringsmuligheter. Eksempelet under viser hvordan.

```
IQueryable<Dokument> Query = from dokument in _db.Dokumenter select dokument;  
  
if (et filter er valgt) {  
    Query = Query.where(dokument => dokument.Attributt.Equals(valgt_filter));  
}  
  
if (enda et filter er valgt) {  
    Query = Query.where(dokument => dokument.Attributt.Contains(enda_et_valgt_filter));  
}  
  
Query.ToList();
```

Figur 63, Eksempel for å hente alle dokumenter med ny LINQ metodikk.

Med bruk av LINQ på denne måten fikk vi kuttet lengden på algoritmen fra 240 til 75 linjer kode. Den nye versjonen viste seg å være raskere, krevde færre databaseaksesser og var lettere å lese og forstå.

5.3.3 Håndtering av page reload

Et problem vi brukte lang tid på å fikse var at dersom brukeren hadde valgt et filter eller søkt på noe og deretter trykket refresh siden, eller på hjem-knappen ville det tidligere valgte filteret fremdeles være valgt. Dette kunne føre til forvirring hos brukeren siden teksten på filtrering knappene var tilbakestilt og søkerfeltet var tomt, men dokumentene som vises er de som matcher det tidligere søker/filtreringen.

Løsningen for dette problemet var å implementere en metode som henter dokumenter som om ingen filter, søker eller sortering er valgt og resetter alle variabler assosiert med filtreringen på server-siden til deres opprinnelige verdi. Denne metoden kalles på i onPageLoad()-funksjonen i javascript. Det er også denne metoden som blir kalt dersom brukeren trykker på "tilbakestill".

5.3.4 Research

Noen av de tekniske utfordringene vi møtte på var godt dokumenterte, og vi visste at denne dokumentasjonen ville være nødvendig å lese før selve utviklingen begynte. I dette underkapittelet vil vi gå gjennom de høyest prioriterte punktene.

5.3.4.1 Betalingsløsning

Enhver nettside som lar brukere bruke sine penger på produkter er nødt til å stille strenge krav i teknologien som skal benyttes ved betalingsløsningen. Omfattende research ble gjort på lover og regelverk, samt "beste-praksis"-metoder som er dokumentert fra andre bedrifter. Det ble bestemt å benytte seg av den tredjeparts-betalingsleverandøren Stripe for betaling med bankkort. Deres bedrift har en "developer-first" filosofi, som gjør at deres dokumentasjon og anbefalte fremgangsmåte er meget godt egnet for oss, og etter omfattende testing av vår implementasjon anser gruppen betalingsløsningen til å være stabil og pålitelig.

5.3.4.2 Deployment

Gruppen har kun erfaring med å bygge lokale nettsider som ikke er tilgjengelig for allmennheten. Dette prosjektet skal kort tid etter bacheloroppgaven er levert ved

OsloMet ut på internettet og tilgjengeliggjøres for oppdragsgiver og allmennheten. Leverandøren av servertjenesten er blitt bestemt basert på gruppens research i henhold til bedriftens budsjett og vi har landet på en leverandør med gode omtaler og pålitelig service.

5.3.4.3 Fra .NET Framework til .NET Core

Teknologien som ble benyttet for utviklingen av løsningen er Microsoft ASP.NET Core (Forkortet til .NET Core). .NET Core er en videreutviklet versjon av Microsoft sitt originale rammeverk, Microsoft ASP.NET Framework (Forkortet til .NET Framework). .NET Framework er det rammeverket som ble lært da vi tok emnet ITPE3100 Webapplikasjoner høsten 2019. For dette prosjektet er det valgt å benytte .NET Core i stedet for .NET Framework. Strukturen og syntaksen i de to rammeverkene er nokså like, men det er fortsatt forskjeller som gjør at løsninger som fungerer i .NET Framework ikke fungerer i .NET Core. Disse forskjellene måtte læres før vi kunne begynne å skrive kode med høy nok kvalitet. Endringen fra .NET Framework til .NET Core ble gjort på grunn av anbefaling fra microsoft, og på grunn av at .NET Framework ikke fungerer i Mac OS X, som flere av gruppemedlemmene benytter seg av.

6. Refleksjon

I dette kapittelet vil vi gå gjennom gruppens læringsutbytte fra dette prosjektet, både rent teknisk og også hvilke ferdigheter og tanker om prosjektets fremtid gruppen har.

6.1 Læringsutbytte

Gjennom hele prosjektets løp har vi lært mye, både tekniske ting som programmeringsteknikker og mer bedriftsrettede ting som lover og regler rundt bruk av informasjonskapsler. Det har vært et utfordrende prosjekt hvor vi har brukt utallige timer for å få sluttproduktet så bra som mulig, men til tross for alle utfordringer og frustrasjoner når ting ikke har fungert som forventet, har vi også hatt det gøy med å se prosjektet gå fra ide til et faktisk system som skal ut i bruk. Det å utvikle et såpass komplekst system ser vi også på som god erfaring før vi skal ut i arbeidslivet og jobbe med slike systemer hver dag.

Vi har også forbedret våre egenskaper til å samarbeide på et prosjekt ved hjelp av git. Alle gruppemedlemmene hadde god kjennskap til git fra før, men vi har alltid syntes det har vært vanskelig og knotete å få det til å fungere. Etter over 400 git commits i dette prosjektets repository kan vi trygt si at dette er noe vi har blitt sterkere på.

Kommunikasjon og organisering innad i gruppen har også forbedret seg gjennom prosjektet. Det å faktisk planlegge et videomøte hvor vi samarbeider kan være lite motiverende dersom dårlig moral, dårlig kommunikasjon eller mangel på fremgang preger gruppen. Etterhvert som systemet begynte å virkelig ta form, oppstod det en stemning innad i gruppen hvor man gledet seg til neste møte for å kunne vise frem det man selv hadde gjort, og for å se hva resten av gruppen har fått til siden siste møte. Denne følelsen er kanskje ikke en tradisjonell ferdighet som har blitt lært, men likevel mener gruppen at ikke skal sløyfes fra kapittelet om læringsutbytte, da erfaringen av å ha en stolthet over hva man selv, og gruppen som helhet har oppnådd er en erfaring i og for seg selv, som vi vil ta med oss i arbeidslivet.

6.2 Videreutvikling

Det at systemet skal videreutvikles etter vi overleverer det til oppdragsgiver er noe vi har vært klar over siden begynnelsen av prosjektet. Vi har derfor hatt som mål å skrive vedlikeholdbar og skalerbar kode som kan utvides senere. Dersom ny funksjonalitet ønskes i systemet skal det være enkelt og forståelig å få til dette uten å måtte endre på vår eksisterende kode. I underkapitlene under går vi mer i detalj om hvilke grep vi har tatt for å tilrettelegge for videreutvikling og noen forslag for ny funksjonalitet som vi har kommet på.

6.2.1 Grunnlag

Det å skrive kommentarer i kildekoden som forklarer hvordan koden fungerer er et viktig tiltak for å gjøre det enklere for fremtidige utviklere å vedlikeholde og utvide systemet. Vi har benyttet oss av visual studio sin innebygde kildekodekommenterings-konvensjon. Dersom man skriver tre skråstreker etter hverandre vil en forhåndslaget mal for dokumentering skrives inn i kildekoden.

```
/// <summary>
///
/// The main method used for gathering documents for the document viewer.
///
/// </summary>
/// <param name="CurrentPage"></param>
/// <param name="UserId"></param>
/// <param name="CurrentFilterSettings"></param>
/// <param name="SearchQuery"></param>
/// <param name="DegreeIsSet"></param>
/// <param name="PlaceOfStudyIsSet"></param>
/// <returns>
/// A list containing documents according to which filters are set.
/// </returns>
```

Figur 64, Eksempel på en kildekodekommentar

Produktdokumentasjonen vi har skrevet i denne rapporten vil også være et grunnlag for videreutvikling. For at fremtidige utviklere skal kunne videreutvikle systemet er det viktig at de forstår hvordan systemet fungerer og er bygget opp.

Produktdokumentasjonen kan da leses og studeres av utviklere for å få den kompetansen de trenger for å begynne videreutvikling og utvidelse av systemet.

6.2.2 Forslag

Gjennom utviklingen av systemet har vi kommet opp med en rekke forslag til ny funksjonalitet som kan utvikles. Vi har latt være å implementere dette selv fordi det ikke var en del av kravspesifikasjonen og vi måtte heller fokusere på å få ferdig det nødvendige fra kravspesifikasjonen.

6.2.2.1 Komprimering av dokumenter

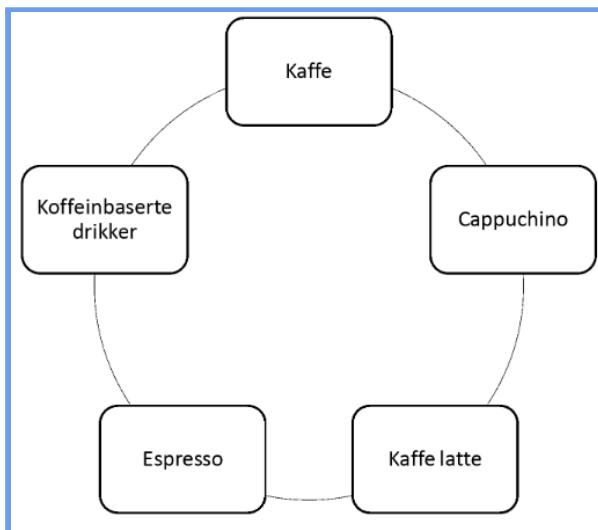
En ide vi hadde var å komprimere dokumentene som lastes opp ved hjelp av huffman-trær som vi har lært oss i faget DATS2300 - Algoritmer og Datastrukturer. Huffman komprimering er en type komprimering som kalles for “lossless compression”, det vil si at filstørrelsen blir vesentlig mindre uten at noe informasjon blir tapt i prosessen. Fordelene med å minimere filstørrelsen til dokumentene er både det å spare på lagringsplass og å minske lastetider når det kommer til henting av dokumenter fra DB.

6.2.2.2 Synonymringer

En synonymring er en liste med ord som er regnet som synonymer. Dette blir brukt i sammenheng med søkelogikk. Dersom et ord i listen blir søkt på vises treff for resten av synonymene også. Fordelen med dette er at det kan ofte gi brukerne flere søker treff. Ofte kan brukere som leter etter det samme bruke forskjellige termer. En ulempe med bruk av synonymringer er at brukere kan bli forvirret over søkeresultatene siden det vises resultater for andre ord enn det som ble søkt på.

(M.W Aasheim, *personlig kommunikasjon, 2019, Metadata & kontrollert vokabular.pptx*, s.7-10)

Om synonymringer skal implementeres i dette systemet sin søkermotor må derfor vurderes av de fremtidige utviklerne i samarbeid med bedriftens styre.



Figur 65, Eksempel på synonymring

6.2.2.3 Metadata

Det å implementere metadata til de forskjellige html-filene vil hjelpe søkermotorer som Bing og Google med å finne nettsiden. Dette er noe vi ikke har gjort, både fordi vi ikke er helt sikre på riktig måte å gjøre det på og fordi det ikke var noe krav om dette fra oppdragsgiveren. Med litt research vil det nok ikke være vanskelig for fremtidige utviklere av systemet og implementere dette.

6.2.2.3 Utvide til alle typer dokumenter

Etter utviklingen av dette systemet fant vi ut at Way2Easy nå hadde en plattform som med noen få justeringer vil fungere like godt for alle typer dokumenter. Dersom bedriften i fremtiden vil utvide sitt system til å støtte flere typer dokumenter er dette veldig mulig. Ideen var jo en markedslass for dokumenter relatert til skole, enten videregående eller høyere utdanning. Dette kan utvides til å være en markedslass for dokumenter generelt, enten det er akademiske skolerelaterte dokumenter, vitenskapelige forskningsartikler, intervjuer eller andre materiale.

6.3 Bachelorprosjektet under covid-19 pandemien

I dette kapitlet vil vi kort forklare hvordan det har vært å jobbe med et bachelorprosjekt hvor man er begrenset av covid restriksjoner.

Gjennom hele vår bachelorgrad har vi vært vant til å møtes på skolen for å arbeide med oppgaver, dette var også planen for bachelorprosjektet. Det fikk vi ikke gjort da campus har vært stengt grunnet pandemien. Det har vært utfordrende å kun sitte hjemme foran PC-skjermen og samarbeide via nettbaserte tjenester. Det å ikke kunne møte de andre gruppemedlemmene har også vært trist og ganske demotiverende til tider, men vi har kommet oss gjennom det fordi vi har klart å fokusere på hva som faktisk er målet; å levere et produkt med så høy kvalitet som mulig.

7. Konklusjon

Etter endt prosjekt mener vi selv at vi har levert et produkt av høy kvalitet, en mening vi deler med oppdragsgiver, som har gitt oss meget god tilbakemelding og er svært fornøyde med systemet vi leverer. Denne positive tilbakemeldingen er også i bakgrunn av hvorvidt vårt system fyller kravene i kravspesifikasjonen. Utviklingen av Dokumarked har vært en lang og krevende prosess, både rent teknisk og i konteksten av systemutviklingsprosessen. På grunn av dette har prosjektet vært spennende, og vi er glade for at vi har utviklet et system med slikt omfang.

Gruppens samarbeid har vært god gjennom hele utviklingsprosessen og vi er alle enige om at dette prosjektet har gjort vårt samarbeid bedre og mer effektivt enn noensinne før. Takket være at oppdragsgiver er en nyoppstartet bedrift har vi fått følt på “startup-ånden”, og har lært utsøkt mye om prosjektstyring på grunn av vårt store ansvar for prosjektets fremgang.

Vi mener selv at innsatsen vi har lagt inn i dette prosjektet overgår all forventning vi selv hadde før vi begynte. Da fremgangen i utviklingen virkelig ble tydelig, startet en slags snøballeffekt hvor vi bare ble mer og mer ivrige på å fortsette med å utvikle systemets mange funksjonaliteter. Ved hvert møte med oppdragsgiver fikk vi også ny motivasjon av å høre deres positive reaksjoner til det vi hadde gjort siden sist. Gruppen er enige om at systemet nå er klart til å leveres og til å tas over av neste utviklerteam for videreutvikling og drift. Vi føler at vår prosess i gjennom dette prosjektet reflekterer i hvor stor grad vi har veklagt god dokumentasjon og vedlikeholdbarhet av systemet. Denne rapporten er først og fremst ment å være en ressurs for Way2Easy sitt fremtidige utviklingsteam for å kunne få en dyp forståelse av hvordan systemet er strukturert og hvordan systemets prosesser er sydd sammen.

Referanser

NB! Referanser merket med “personlig kommunikasjon” i teksten er tatt fra forelesninger fra fag vi har hatt. Disse powerpointene er ikke tilgjengelige for andre enn studenter som har hatt faget og er logget inn på canvas. Slik APA-siteringsformatets konvensjoner bestemmer vil disse referansene ikke inkluderes i referanselisten under, fordi leserne vil ikke ha tilgang på disse.

Angeles, K. (2019, September 25). Stepping Out of the Light: Tips for the design and development of dark mode. Medium.

<https://medium.muz.li/stepping-out-of-the-light-tips-for-the-design-and-development-of-dark-mode-bb6f7a38043d>

Bartram, L., Patra, A., & Stone, M. (2017). Affective Color in Visualization. Tableau Research. Published.

<https://research.tableau.com/sites/default/files/Affective%20Color%20CHI%202017.pdf>

Batchu, V. (2018, April 12). Micro-interactions: Why, when and how to use them to improve the user experience. Uxdesign.

<https://uxdesign.cc/micro-interactions-why-when-and-how-to-use-them-to-boost-the-user-experience-17094b3baaa0>

Datatilsynet. (u.å.). Nettbutikker og kundeopplysninger. Retrieved May 6, 2021, from <https://www.datatilsynet.no/personvern-pa-ulike-områder/kundehandtering-handel-og-medlemskap/nettbutikker-og-kundeopplysninger/>

Dependency injection in Android |. (2020, June 10). Android Developers.

<https://developer.android.com/training/dependency-injection>

Eisfeld, H., & Kristalovich, F. (2020). The Rise of Dark Mode : A qualitative study of an emerging user interface design trend. Diva-Portal.

<http://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-50563>

Ellis-Chadwick, F. (2019, August 30). History of online retail. OpenLearn.
<https://www.open.edu/openlearn/money-management/management/business-studies/history-online-retail>

Hero Image. (u.å.). Optimizely. Retrieved May 20, 2021, from
<https://www.optimizely.com/optimization-glossary/hero-image/>

Microsoft. (2016, Oktober 14). Simple authorization in ASP.NET Core. Microsoft Docs.
<https://docs.microsoft.com/en-us/aspnet/core/security/authorization/simple?view=aspnetcore-5.0>

Microsoft. (2016b, Oktober 23). Async query and save - EF6. Microsoft Docs.
<https://docs.microsoft.com/en-us/ef/ef6/fundamentals/async>

Microsoft. (2019, June 12). Partial views in ASP.NET Core. Microsoft Docs.
<https://docs.microsoft.com/en-us/aspnet/core/mvc/views/partial?view=aspnetcore-5.0>

Nkom. (2020, March 6). Informasjonskapsler/cookies. Nasjonal kommunikasjonsmyndighet.
<https://www.nkom.no/internett/informasjonskapsler-cookies>

Palmer, S. R., & Felsing, J. M. (2002, Mars 22). Feature-Driven Development: Best Practices | Integrating Best Practices | InformIT. Informit.
<https://www.informit.com/articles/article.aspx?p=26059>

Patton, T. (2006, Mai 30). Use the data access layer to simplify architecture. TechRepublic.
<https://www.techrepublic.com/article/use-the-data-access-layer-to-simplify-architecture/>

UX og UI design - hva er egentlig forskjellen? (2019, April 27). Guilty.
<https://guilty.no/blogg/forskjellen-p%C3%A5-ux-og-ui-design>

W3. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1.*

<https://www.w3.org/TR/WCAG21/>

Liste over figurer

Figur	Forklaring	Side figuren brukes
Figur 1	Gruppens fremdriftsplan uke for uke, i form av et gantt-diagram.	19
Figur 2	Utviklingsprosessen i korte trekk. Etter tilbakemelding fra arbeidsgiver blir featuren implementert, eller videreutviklet. Dersom en feature blir implementert, går man til feature-listen for å finne neste feature som skal utvikles.	22
Figur 3	en forenklet modell av hvordan forholdet mellom entitet og view-modell sikrer kommunikasjon mot DB.	36
Figur 4	Pseudokode som viser hvordan en spørring for å legge inn en ny person i databasen kan se ut ved bruk av SQL.	40
Figur 5	den samme pseudokode-operasjonen som i Figur 4, gjennom bruk av Entity Framework	40
Figur 6	side for å registrer en ny bruker	47
Figur 7	side for å logge seg inn med en registrert og bekreftet bruker.	48
Figur 8	Fullstendig bilde av forsiden med forklarende tekst for de ulike komponentene.	49

Figur 9	Søk, sortering og filtreringsmuligheter på forsiden.	50
Figur 10	Feilmelding dersom bruker velger mitt studested/studieretning og ikke har valgt tilsvarende.	52
Figur 11	pagination med side 7 som valgt side og 20 som maks antall sider.	53
Figur 12	siden for hvert enkelte dokument med forhåndsvisning og ytterligere informasjon om dokumentet.	54
Figur 13	Handlekurv med et dokument.	55
Figur 14	Skjermbilde av FAQ med et spørsmål åpent.	55
Figur 15	Skjermbilde av en brukers offentlige profil	56
Figur 16	Kodesnutt for autorisering tatt fra microsoft sin dokumentasjon av ASP.NET Core	57
Figur 17	skjemaet som lar brukere laste opp et dokument til siden.	58
Figur 18	dersom bruker oppgir karakter, vil 2 nye felt komme i skjemaet, for å fylle ut informasjon fra vitnemålsportalen.no	59
Figur 19	siden for å betale for dokumenter i handlekurven. Skjemaet for å taste inn kortopplysninger er produsert av Stripe, og systemet vi har utviklet vet aldri hva betalingsopplysningene til brukeren er.	60
Figur 20	et diagram som punktvis forklarer flyten fra når brukeren ber om å få kjøpe et dokument, til at Stripe sine servere gir varsel til	60

	Dokumarked om at en betaling har vært vellykket.	
Figur 21	brukere sin private profil, med tilgang til deres kjøpte dokumenter, opplastede dokumenter, og oversikt over inntjening	61
Figur 22	en bruker har kjøpt 2 dokumenter, og de er tilgjengelige for nedlasting under ”Mine kjøpte dokumenter” i deres private profil.	61
Figur 23	modalen som vises når en bruker vil legge ut en omtale på et dokument de har kjøpt.	62
Figur 24	siden hvor brukere kan redigere deres profil. Til venstre vises de forskjellige typene informasjon de kan endre.	63
Figur 25	en bruker ser sin neste utbetaling, med oversikt over hvor mye som blir utbetalts, og når utbetalingen kan forventes. Under vises det at brukeren ikke har mottatt noen utbetalinger tidligere.	64
Figur 26	et dokument trengs å godkjennes før det legges ut til salgs på nettsiden. En lenke til å se detaljene til dette spesifikke dokumentet er inkludert i tabellen.	65
Figur 27	dette er siden som vises etter de har trykket ”Godkjenn eller avis”-lenken til et dokument, som vist i figur 26. Siden inneholder grundigere informasjon om dokumentet.	66
Figur 28	en bruker med navn ”Tester” holder på å bli uteslengt fra nettsiden, etter at brukernavnet deres er søkt opp.	66

Figur 29	etter at brukeren “Tester” har blitt utestengt, ligger brukeren på oversikten over utestengte brukere.	67
Figur 30	et dokument eid av brukeren “Tester” har blitt rapportert en gang.	68
Figur 31	en oversikt over hver av rapportene dokumentet til brukeren “Tester” har fått.	68
Figur 32	oversikt over alle brukere som er blitt rapportert, og hvor mange ganger brukerne har blitt rapportert	68
Figur 33	alle rapportene som en enkel bruker har fått. I dette tilfellet har brukeren fått en rapport i kategorien “Annet”, med kommentar: “Brukeren har et upassende profilbilde”.	69
Figur 34	Oversikt over alle omtaler som er blitt rapportert, og hvor mange ganger omtalene har blitt rapportert	69
Figur 35	Oversikt over alle rapporter en enkel omtale har fått. I dette tilfellet har den fått en rapport i kategorien “Upassende navn” uten noen kommentar fra brukeren som rapporterte den.	70
Figur 36	viser oversikt over utbetalinger som enda ikke er betalt. Utbetingene er blitt automatisk hentet fra perioden 15. April til 15. Mai.	71
Figur 37	En Til- og Fradatovelger som lar oppdragsgiver hente tidligere utbetalinger, dette er ment å fungere som en regnskapsløsning i tillegg som en utbetalingsoversikt.	71

Figur 38	En av utbetalingene ble mislykket og en epost er blitt sendt til brukeren hvor de bes ta kontakt med Way2Easy så fort som mulig. En utbetaling med statusen "Betalt" vises også, og blir lagt nederst i tabellen.	72
Figur 39	en utbetaling har ikke blitt betalt innen fristen, som var 15. mars 2021. Derfor blir den inkludert selv om de valgte til- og fradatoene er fra 15. april til 15. mai.	72
Figur 40	side hvor administratorer kan registrere og oppdatere roller på nettsiden.	73
Figur 41	søkestrengen fant brukeren "Tester", og administratoren kan nå redigere "Tester" sin rolle på nettsiden.	73
Figur 42	popup-vinduet som ber brukere av siden om å akseptere Dokumarked sin cookie policy. Ingen cookies på siden brukes til andre formål enn at siden skal fungere.	75
Figur 43	headeren til siden dersom brukeren ikke er innlogget.	76
Figur 44	headeren til siden dersom brukeren er innlogget	76
Figur 45	headeren til siden dersom brukeren som er innlogget er en administratorbruker	76
Figur 46	Første skisse for dokumentvisning til venstre. Ferdig dokumentvisning til høyre.	77
Figur 47	Skjermbilde av tidlig versjon av nettsiden.	78
Figur 48	Testing av kontrastforholdet mellom bakgrunnsfargen og forgrunnsfargen som	79

	brukes på nettsiden via https://webaim.org/resources/contrastchecker	
Figur 49	Skrifttypen “Roboto”. En av skrifttypene vi vurderte å bruke på nettsiden.	81
Figur 50	Skrifttypen “Montserrat”. En av skrifttypene vi vurderte å bruke på nettsiden.	81
Figur 51	Skrifttypen “Nunito”. Det er denne vi ble enige om å bruke på nettsiden.	82
Figur 52	Forskjellige tooltips på nettsiden som forklarer symbolene på siden.	83
Figur 53	Viser forskjellen på et dokument når musen holdes over.	83
Figur 54	Viser forskjellen hvordan knapper ser ut til vanlig (til venstre) og når musen holdes over (til høyre)	84
Figur 55	Viser forskjellen hvordan knapper med hvit bakgrunn ser ut til vanlig (til venstre) og når musen holdes over (til høyre)	84
Figur 56	Eksempel på css kode for å fargelegge et element	84
Figur 57	Eksempel på css custom properties.	85
Figur 58	Eksempel på bruk av css custom properties for darkmode funksjonalitet.	85
Figur 59	Innloggingssiden uten darkmode.	86
Figur 60	Innloggingssiden med darkmode.	86
Figur 61	Filtreringsmuligheter med deaktiverte knapper.	90

Figur 62	Eksempel for å hente alle dokumenter.	91
Figur 63	Eksempel for å hente alle dokumenter med ny LINQ metodikk.	91
Figur 64	Eksempel på en kildekodekommentar	95
Figur 65	Eksempel på synonymring	96

Vedlegg

1. Prosessdokumentasjon

Forord

I fra start til slutt i dette bachelorprosjektet har systemutviklingen fulgt en prosess som har blitt dokumentert i denne rapporten. Hovedsakelig inneholder denne prosessen først en planleggingsfase og deretter en utviklingsfase. Vanligvis ville utviklingsfasen etterfølges av en drifting- eller “deployment”-fase, men i vårt tilfelle har dette blitt byttet ut med en dokumentasjonsfase. Dette er fordi at når utviklingsfasen er konkludert skal systemet overgis til oppdragsgiver for å settes ut i drift uten oss, og derfor vil denne delen av systemets liv ikke dokumenteres her. Vi vil diskutere valg som er gjort og metodikk som er benyttet både i planleggingen av utviklingen og i utviklingen av selve systemet. En viktig del for at systemutviklingsprosessen har gått så bra som det har gjort har vært takket være et nært og godt samarbeid med oppdragsgiver, og vi har i denne rapporten også gjort et forsøk i å dokumentere hvorfor dette samarbeidet gikk så bra.

Innholdsfortegnelse

Prosjektets faser	111
<hr/>	
Planleggingsfasen	111
<hr/>	
Idemyldring	112
<hr/>	
Research for valg av teknologi	113
<hr/>	
Valg av teknologi	113
<hr/>	
Bestemmelse av prosjektets utviklingsmetodikk	113
<hr/>	
Budsjettering	116
<hr/>	
Oppretting av kravspesifikasjonen	117
<hr/>	
Oppretting av feature-listen	117
<hr/>	
Utviklingsfasen	117
<hr/>	
Utviklingsløpet	128
<hr/>	
Utvikling tidlig i prosjektet	120
<hr/>	
Utvikling midt i prosjektet	121
<hr/>	
Utvikling sent i prosjektet	123
<hr/>	
Dokumenteringsfasen	124
<hr/>	
Dokumentering av produktet	124
<hr/>	

Dokumentering for Prosjektoppgaven	124
<hr/>	
Dagbok	125
<hr/>	
Møtereferat	125
<hr/>	
Møter innad i gruppen	125
<hr/>	
Oppdragsgiveren og oss	125
<hr/>	
Faglig oppsummering	126
<hr/>	
Fremtid	127

Prosjektets faser

Prosjektet har hovedsakelig foregått gjennom to faser, hvor planleggingsfasen har dannet grunnlaget for utviklingsfasen. Vi vil i dette underkapittelet gå igjennom hvert av punktene i prosjektets faser fra starten til slutten, for å dokumentere hvilke beslutninger som har ledet til produktet som er dokumentert i prosjektets sluttrapport.

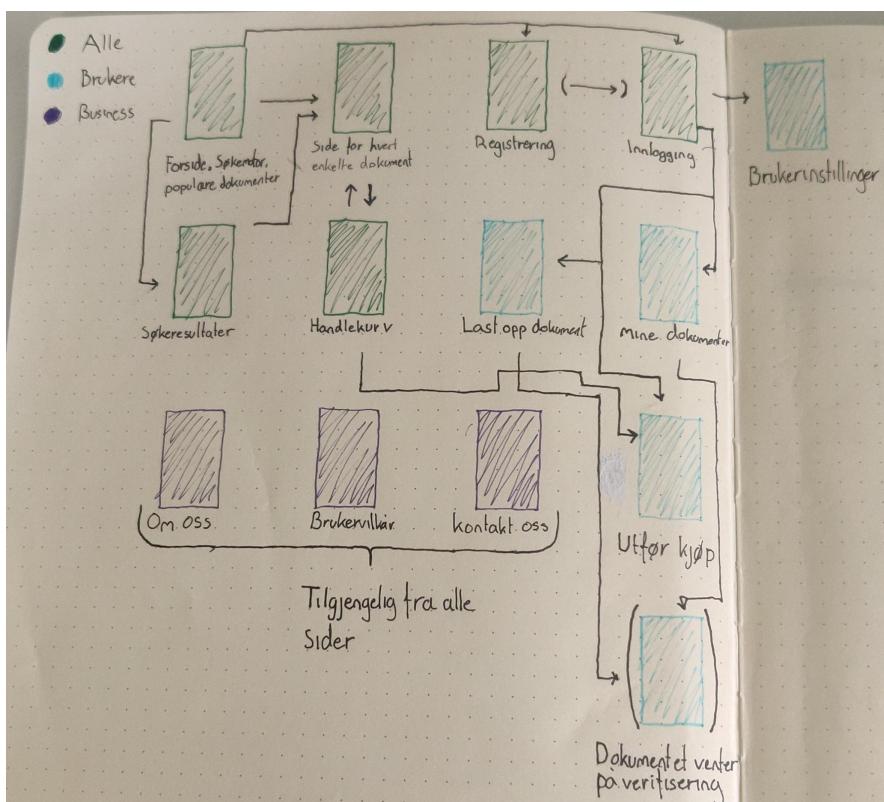
Planleggingsfasen

Vi ble kontaktet av Way2Easy før bachelorprosjektet hadde begynt. Dette gjorde at gruppens diskutering og beslutning at vi ville ta dette oppdraget skjedde i god tid, som gjorde at mye av planleggingen ble gjort før årsskiftet. I planleggingsfasen har vi utført budsjettering og idemyldringer sammen med oppdragsgiver, og researchet de teknologier som ville oppnå ønskene presentert i idemyldringen. Vi har besluttet utviklingsmetodikken og bestemt hvilke teknologier som skal benyttes for å utvikle produktet. De dokumentene som har vært mest vesentlige for systemets utvikling, nemlig kravspesifikasjonen og feature-listen har også i denne fasen blitt produsert.

Her vil vi gå igjennom hver av disse etappene og forklare gruppens tanker og avgjørelsene vi har gjort.

Idemyldring

I det første møtet gruppen hadde med oppdragsgiver den 22. Oktober 2020 ble vi presentert med deres visjon av produktet de håpet vi kunne levere. Allerede her begynte en slags muntlig kravspesifikasjon å utforme seg, som ville bli utdypt senere. Da dette møtet var over hadde gruppen takket ja til prosjektet, og diskuteringer innad i gruppen om valg av teknologi, utviklingsmetodikk og kartlegging av systemets arkitektur hadde begynt.



Figur 1, en tidlig skisse over hvilke sider nettsiden ville trenge, men en generell flyt til hver av de.

Under idemyldringen fikk gruppen gi tilbakemelding til bedriftens forslag om hva som var realistisk, hva som ikke var realistisk, og hvilke områder som ville være spesielt ressurskrevende under utviklingen av Dokumarked.

Det ble også diskutert hvordan arbeidsflyten til gruppen sammen med oppdragsgiver kunne virke. Oppdragsgiver var meget interessert i å ha et tett samarbeid med

gruppen for å kunne gi tilbakemeldinger på produktet mens det var i utvikling. Det ble dermed bestemt at gruppen skulle ha et møte med bedriften hver fredag hvor vi viste frem produktets fremgang, samt fikk tilbakemelding på funksjonaliteten som er blitt implementert uke for uke. Dette ble tatt i betraktning under valg av utviklingsmetodikk.

Research for valg av teknologi

Denne research-fasen gikk ut i fra idemyldringen for å grundig undersøke mulige teknologier, rammeverk og utviklingsmetoder. Det er også i denne delen av prosjektet hvor gruppen gjentok pensum og annen dokumentasjon for å spisse ferdighetene som ville være nødvendige for å utvikle systemet.

Valg av teknologi

Etter en rudimentær mental skisse av nettsiden var dannet ble gruppen enige om hvilke teknologier som kunne være gunstige å utvikle produktet med. Det ble bestemt at server-siden skulle utvikles i ASP.NET Core og det ble planlagt at klient-siden skulle utvikles som en webapplikasjon med React.

Bare noen dager senere gjorde en diskusjon innad i gruppen at vi bestemte å utvikle produktet med .NET Core MVC-rammeverket og ikke i React, grunnet oppgavens store omfang. Dersom oppstartsfasen i prosjektets løp skulle gå til å lære oss React godt nok til å kunne produsere kode av tilfredsstillende kvalitet ville vi antageligvis ikke fått god nok tid. Dette ble besluttet tidlig i prosjektet og er en beslutning gruppen er enige om at var den rette.

Bestemmelse av prosjektets utviklingsmetodikk

Ut i fra det generelle oppsettet av produktet som ble produsert etter idemyldringen ble det bestemt hvilken utviklingsmetodikk som skulle benyttes. Da Way2Easy er en nyoppstartet bedrift som ikke enda har fullstendig utviklet sine bedriffsstrategier, merkevarer eller visjon ble det påregnet at hyppige endringer i kravspesifikasjonen kunne forekomme.

Det var derfor et åpenbart valg at utviklingsmetodikken som skulle benyttes var av den smidige sorten. Gruppen lente seg først mot Kanban, da vi tenkte at denne metodikkens filosofi med ”jobb fra der du står nå” ville fungere godt med hvordan oppdragsgiver ønsket å holde møte med gruppen hver fredag for å vise frem prosjektets fremgang. Oppdragsgiver var positiv til Kanban, men det var klart at en tydelig oversikt over alt som ville være mulig å gjøre på nettsiden etter at prosjektet er fullført var å foretrekke.

Etter en diskusjon innad i gruppen ble det bestemt at en annen smidig metodikk med navnet ”feature-driven” utvikling kunne passe vår situasjon bedre. Dette er også en metodikk som tar utgangspunkt i der du er nå, med et større fokus på hvordan produktet vil se ut etter at alle disse feature-ene er utviklet. Følgende prinsipper i ”Feature”-drevet utvikling gjorde at vi valgte å gå for denne metodikken:

Prinsipp	Forklaring	Grunnlag for valg av metodikk
Feature-liste	En så fullstendig som mulig liste/oversikt over features som produktet/systemet vil inneholde ved prosjektets slutt.	Gi oss selv og ikke minst oppdragsgiver en oversikt over hva gruppen har forstått at skal utvikles.
Utvikling etter feature	Funksjoner som er for kompliserte til å implementeres innen 2 uker brytes ned til mindre prosesser som defineres som egne features i systemet.	Sikrer mot massive oppgaver som fort kan bli for tidkrevende å løse i ett sett. Å bryte opp oppgaver i mindre sett gjør det også enklere for to utviklere å kollaborere på samme oppgave.
Individuell kode- eller klasse-eierskap	Distinkte klasser og/eller kode i systemet eies til en hver tid av en av	Enhver av gruppemedlemmene er ansvarlige for å sikre god

	gruppemedlemmene, som har ansvar for å sikre at denne koden er “up-to-date” og klar for å bygges feilfritt.	kvalitet på koden de skriver, samt at en alltid vet hvem man skal spørre dersom kode du skriver kommer til å interagere med koden til noen andre.
Inspeksjoner	Regelmessige inspeksjoner av kode gjøres for å sikre en god kvalitet på design og kode. I vårt tilfelle er dette gjort ved pull-requests mot “main-branch”-en vår, ved bruk av GitHub.	Sikrer kode av høy nok kvalitet
Konfigurasjonsstyring	Oversikt over prosjektets kildekode og oversikt over alle de forskjellige “delene” som til enhver tid er i spill. Dette er oppnådd ved bruk av ClickUp, og ved bruk av GitHub - i tillegg til daglige møter og ukentlige møter med bedriften.	Gjør det lettere for gruppen å få innsikt i hva som bør prioriteres til en hver tid, i tillegg til å forenkle oppgaven om å dele prosjektets fremgang med oppdragsgiver under ukentlig møte.
Regelmessige builds	Systemet skal “alltid” være klar til å vises til oppdragsgiver, for å vise features som er utviklet, og for å få tilbakemelding på om disse features-ene opprettholder kravspesifikasjonen og kan implementeres.	Systemet vil alltid være klart til å vises frem ved det ukentlige møtet gruppen har med bedriften. Sikrer også at bugs som forårsaker byggings-feil i koden diagnostiseres og fikses hurtig.

Synlighet av fremgang og resultater	Kort fortalt er en oversikt over hva som må gjøres, hva som blir gjort, og hva som er ferdiggjort, er ett av de mest utbredte "beste praksisene" i smidig utvikling. Dette er oppnådd gjennom vår bruk av ClickUp.	Gir både gruppen og oppdragsgiver mulighet til å prioritere oppgaver og estimere tid til prosjektet er ferdig utviklet.
-------------------------------------	--	---

Budsjettering

Budsjettet til prosjektet har vært relativt lav, men fortsatt fleksibelt. Da Way2Easy - prosjektets klient, er en nyoppstartet bedrift som ikke enda har fått investorer eller andre stipend, har et fokus på lave kostnader så klart vært et tema. Noen utgifter slipper man ikke unna, og fordi gruppen aldri fikk et spesifikt antall kroner i budsjett, har disse utgiftene blitt diskutert med oppdragsgiver der de har oppstått, og enten blitt avslått eller godkjent.

Vertstjenesten

Det var planlagt å bruke Microsoft Azure som vertstjeneste men etter en inspeksjon av prisene til denne løsningen ble det bestemt at vi måtte lete etter et billigere alternativ. Gruppen landet på SmarterASP, som er en vertstjeneste som spesialiserer seg på å verte systemer utviklet i ASP.NET-rammeverkene, med vesentlig billigere planer. Dette selskapets omdømme var meget positivt og det ble dermed bestemt at vi skulle gå for det.

Betalingsløsningen

Det finnes ingen betalingsløsninger som er helt gratis, men det vil likevel ikke si at det nødvendigvis koster penger å benytte seg av de. Den vanligste måten en betalingstjeneste tar betalt for tjenesten, er ved å ta en prosentvis andel av alle kjøp som gjøres gjennom tjenesten. Betalingsløsningen som ble bestemt skal brukes for dokumarked er Stripe, som per dags dato tar 2.4% + 2 kr per betaling for norske kort, og 2.9% + 2 kr for internasjonale kort.

Epostleverandør

Da systemet skal kunne kommunisere med brukerne på siden under visse omstendigheter (som ved registrering og ved kjøp og salg av dokumenter), er en epostleverandør med et API som tillater sending av eposter programmatisk være nødvendig. Det ble bestemt å bruke SendGrid, som tillater 100 eposter gratis hver måned, også må bedriften selv anse om de behøver å oppgradere abonnementet deres.

Oppretting av kravspesifikasjonen

Da metode, budsjett og teknologi var bestemt, ble kravspesifikasjonen produsert på grunnlag av ideene som ble delt under idemyldringen. Kravspesifikasjonen har vært grunnlaget for all funksjonalitet gruppen har utviklet i systemet, og for alle features definert i feature-listen. Kravspesifikasjonen ble også skrevet i samarbeid med bedriften for å passe på at gruppen hadde forstått oppgaven riktig og at vi ville levere systemet slik det står beskrevet ved slutten av prosjektet.

Oppdragsgiver hadde produsert sitt eget dokument med frittflytende krav som hadde blitt lagt til og redigert fortløpende under idemyldringene. Dette dokumentet ble gjennomgått punkt for punkt, og beslutninger om hva som skulle inkluderes i kravspesifikasjonen, og hva som skulle sløyfes/diskuteres videre ble gjort.

Oppretting av feature-listen

Etter at kravspesifikasjonen ble ferdigstilt ble det opprettet en feature-liste som kan leses i sin helhet i vedlegg 3: [Prosjektets feature-liste](#). Feature-listen er dokumentet som forkorter kravspesifikasjonen ned til ett sett med funksjonaliteter/features som i løpet av prosjektet livsløp skal være utviklet. Selve dokumentet fungerer som en helhetlig oversikt, men i praksis har disse features-ene blitt overført til ClickUp for å enklere holde oversikt over hvilke av features-ene som blir utviklet av hvem.

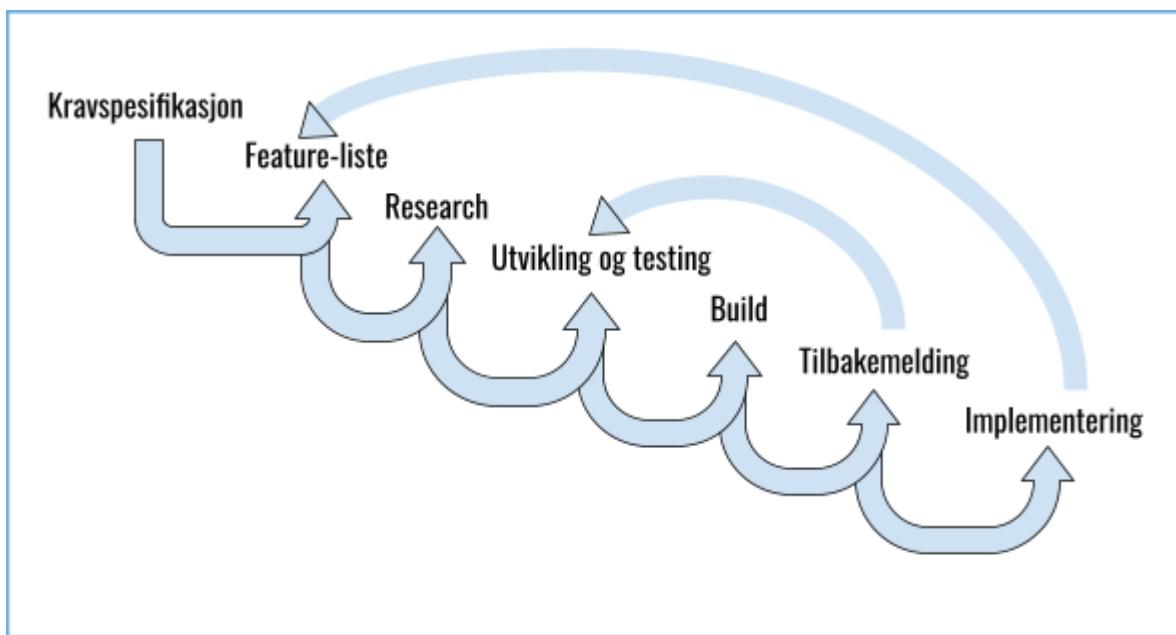
Utviklingsfasen

Da metodikken, kravspesifikasjonen, budsjett, valg av teknologi og feature-listen var nesten ferdigstilt var det klart for å begynne en meget tidlig utvikling av systemet. Dette begynte mandag 4. Januar 2021. Systemet har gått gjennom forskjellige faser,

og for enkelhets skyld er det valgt at i denne rapporten skal disse brytes opp i tidlig, mellom- og sen utvikling. I dette underkapittelet vil vi gå igjennom hver av disse etappene og forklare hva som er blitt produsert og hva som er blitt veklagt i hver av de. Vi vil også gå igjennom hvordan vi har strukturert utviklingsløpet som vi har benyttet oss av på en “dag-for-dag”-basis.

Utviklingsløpet

Gjennom hele utviklingsfasen har det blitt tatt grunnlag i arbeidsflyten som gruppen har bestemt i samarbeid med oppdragsgiver. Vi ville bruke en smidig metodikk som ga muligheten for oppdragsgiver å komme med innspill og gi avsluttende bestemmelser om produktet.



Figur 2, utviklingsprosessen i korte trekk. Denne figuren er den samme som figur 2 i sluttrapporten.

Hvert av de forskjellige stegene i prosjektets utviklingsløp er utdypt i punkt [3.2.2](#) i prosjektets sluttrapport. Mens i denne prosessdokumentasjonen vil flyten i den daglige utviklingen av systemet forklares

Etter at kravspesifikasjonen var ferdigstilt ble en feature-liste opprettet, som ville være gruppen og bedriftens oversikt over hva systemet skal inneholde innen prosjektet avsluttes. Denne listen er grunnlaget for hele utviklingsløpet, og gruppen

går tilbake til denne hver gang en feature er ferdig utviklet, og de skal begynne på en ny.

Etter en ny feature er valgt til å utvikles vil utvikleren, dersom det er nødvendig, utføre research innen hvilke programmeringsteknikker og/eller “beste praksiser” som finnes for like og/eller lignende løsninger.

Når utvikleren føler seg selvsikker om hvordan løsningen skal implementeres vil selve utviklingen utføres. Denne utviklingen innebærer alt fra oppretting av metoder, klasser, style sheets, markup-filer og javascript-filer. Dersom nye rammeverk eller utvidelser må legges til skal disse ha lisenser som er open-source for å sikre at ingen skjulte kostnader oppstår for oppdragsgiver. Dersom gratis teknologier ikke er gode nok, eller ikke finnes, vil dette tillegget diskuteres med oppdragsgiver.

Testingen av systemet har ikke vært en egen del av utviklingsprosessen, men har vært kombinert med utviklingen. Hver utvikler har vært ansvarlig for koden de har skrevet, og vært ansvarlig for å grundig utforske funksjonaliteten og tillegget av kildekode som utvikleren har lagt til i systemet. Dette innebærer både statisk kildekodeanalyse, og manuelle enhetstester. Når alle use-cases for funksjonaliteten er utforsket og ingen feil oppstår vil den nye koden kunne vises frem til oppdragsgiver ved førstkommende fredag.

I disse møtene med oppdragsgiver har gruppen fått tilbakemelding på funksjonaliteten som er implementert og fått tilbakemeldinger på om denne nye funksjonaliteten er god nok for å oppfylle oppdragsgiverens ønsker. I mens disse møtene først og fremst er for å gi oppdragsgiver en kanal for å påvirke utviklingen av deres system, er også møtet benyttet av gruppen til å gi tilbakemeldinger om eventuelle endringer i kravspesifikasjonen er realistiske og rimelige innen tidsrammen vi er gitt.

Dersom den nye koden er godkjent av bedriften vil den kunne foreslås til å merges gjennom en pull-request. For at koden skal smeltes sammen med resten av systemet vil en annen utvikler igjen utføre en statisk kildekodeanalyse og enten godkjenne koden eller foreslå endringer.

Systemet bygges deretter igjen for å utforske om noen andre feil oppstår etter at koden er smeltet sammen med "hoved-grenen", og dersom alt virker som det skal har koden vært antatt klar til å implementeres i produktet.

Deretter går utvikleren til feature-listen og velger neste feature som skal utvikles. For å gjøre det enklere å konfigurasjonsstyre prosjektet er andre verktøy enn feature-liste-dokumentet benyttet for dette, hovedsakelig ClickUp.

Denne flyten har fungert meget godt gjennom hele prosjektets livssyklus, fra den tidlige proof-of-concept-en til den fullstendig designede løsningen. Gruppen har alltid følt at til tross for at oppdragsgiver kan avslå endringer og tillegg i kildekoden, har oppdragsgiver vært positiv og forståelsesfull, og ikke minst hørt etter på våre tanker. Gruppen tror at dette er et av de viktigste årsakene for at vår utvikling av systemet har fungert så godt som det har, men at det ikke er en perfekt løsning for alle utviklingsteam.

Utvikling tidlig i prosjektet

I systemets tidlige utvikling ble det flere ganger opprettet et prosjekt i Visual Studio, på grunn av forskjellige årsaker. Den første årsaken var på grunn av gruppens beslutning om å utvikle løsningen i ASP.NET Core MVC i stedet for i React-rammeverket. Det første prosjektet ble skrapet og byttet ut etter noen få dager med utvikling. Heldigvis var noe av koden fortsatt relevant, og ble overført til det nye prosjektet.

En rekke forskjellige vansker ved å opprette et repository for kildekoden på github gjorde også at et blankt prosjekt måtte opprettes flere ganger. Feilen viste seg å være en bug mellom Visual Studio sitt innebygde GitHub-bibliotek og programvaren Github Desktop. Feilen ble rettet og etter et par dager var utviklingsmiljøet klart til å utvikle systemet.

Research for tidlig utvikling

I den tidlige utviklingsfasen var research for det meste fokusert på hvordan opprette en robust systemarkitektur, og for gruppemedlemmene som ikke hadde noe særlig

erfaring i bruk av ASP.NET Core, ble research av hvordan rammeverket fungerer vektlagt.

Proof of concept

Den tidlige utviklingsfasen var, mer enn noe annet, sentrert rundt å finne ut hva som ville la seg gjøre, og hva som ikke ville være mulig å oppnå. Systemarkitekturen ble påbegynt og systemets grunnmur gjennom bruk av dependency injection og databasekommunikasjon ble bygget. Videre i denne fasen ble de første funksjonalitetene utviklet uten noe fokus på brukergrensesnitt eller interaksjonsdesign. Mye av funksjonaliteten som er utviklet i dette stadiet er fortsatt i bruk den dag i dag.

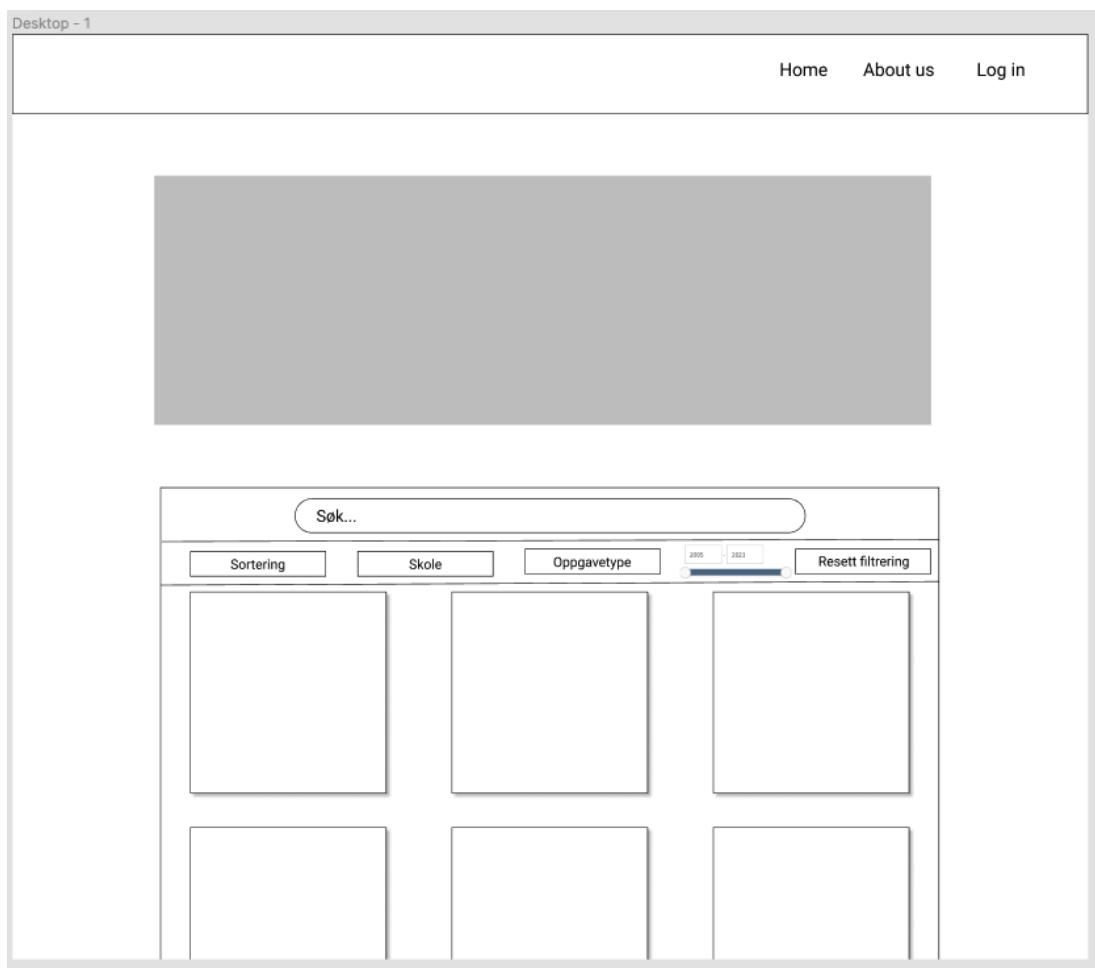
Ved slutten av “proof-of-concept”-stadiet var det tydelig at systemet slik det er beskrevet i kravspesifikasjonen ville være realistisk å oppnå. Utvikling fortsatte stødig i de neste ukene og etter hvert ble det tydelig at gruppen ikke lenger bare utviklet et “proof of concept”.

Utvikling midt i prosjektet

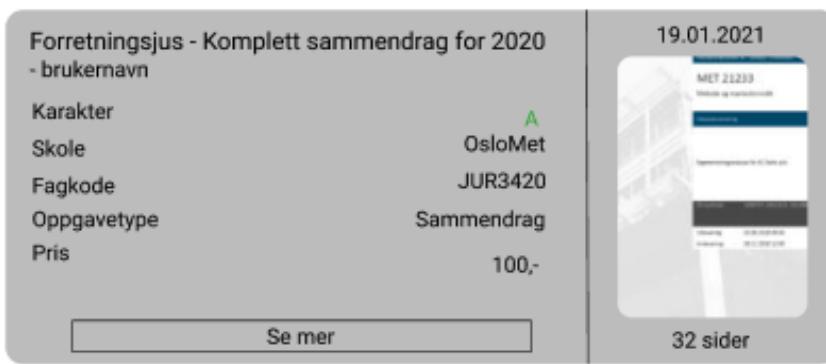
I mellom-utviklingen begynte et system med brukergrensesnitt å formas, til tross for et fokus på funksjonalitet over design. Dette er den desidert største utviklingsfasen, og det er her all funksjonalitet for siden ble utviklet. Hvordan vår hverdag så ut under denne utviklings-etappen er forklart under [Utviklingsløpet](#) i denne rapporten. Vi vil i dette underkapittelet undersøke hva gruppen hadde fokus på i mens vi utførte oppgavene med den flyten som er beskrevet i utviklingsløpet.

Funksjonalitet over design

Et tidlig og primitivt brukergrensesnitt ble først og fremst formet på grunn av bruken av bootstrap under utviklingen av markup-filene, men også gjennom skisser og wireframes, fordi gruppen ville ha noe å vise frem til oppdragsgiver under fredags-møtene.



Figur 3, en tidlig skisse av forsiden av dokumarked



Figur 4, en tidlig skisse av "kartene" på forsiden, som ville vise informasjon om hvert dokument.

Det var likevel et langt større fokus på å få funksjonaliteten på siden på plass. Det i denne fasen alt fra opplastning til betaling for dokumenter ble ferdigstilt og testet, og under dette løpet ble siden ferdig nok til å kunne gå "live", uten at det ville forårsake noen sikkerhetsbrudd eller andre kjøretidsavvik som ville gjøre siden ubruklig for en vanlig bruker. Siden var likevel ikke designet nok til å kunne oppfylle kravene satt i

kravspesifikasjonen, så når all funksjonalitet var på plass gikk utviklingen over i sitt sen-stadium.

Research for mellom-utvikling

Da utviklingen hadde nådd mellom-utviklingen ble det samtidig flere ting som måtte research-es, men det ble lettere å researche disse tingene enn i tidlig-utviklingsfasen. Dette er fordi i mellom-utviklingen visste vi nøyaktig hva vi trengte å researche for å oppnå det ønskede resultatet, mens i tidlig utvikling var det fortsatt en slags tåke over hva vi behøvde å vite for å utvikle systemet.

Research i denne fasen var i stor del sentrert rundt gode programmeringsteknikker og læring av nye rammeverk som det viste seg at systemet behøvde. Et godt eksempel på dette er at det var i denne fasen hvor utviklingen av betalingsløsningen begynte. Takket være en meget utdypt og utvikler-sentrert dokumentasjon fra Stripe ble deres betalingsløsning godt integrert i vårt system

Utvikling sent i prosjektet

Etter at utviklingen av funksjonaliteten på nettsiden begynte å bli ferdigstilt ble utviklingen av designet og interaksjonsdesignet prioritert. I tillegg til et fokus på brukergrensesnittet ble også møtene våre med oppdragsgiver hyppigere. Dette er fordi at design av nettsiden er en viktig del av nettsidens identitet, og i forlengelse bedriftens identitet.

Research for sen utvikling

Medlemmer av gruppen som har mindre erfaring med interaksjonsdesign og utforming av stylesheets var nødt til å utføre grundig research innen disse emnene, for å kunne designe nettsiden etter bedriftens ønsker med et fokus på universell og responsiv utforming. Bruken av bootstrap for strukturen i nettsidens design gjorde denne delen enklere for gruppen, da dette rammeverket er omfattende testet for disse områdene.

Utvikling

Utvikling av server-siden i denne fasen besto av en sjekkliste over alt som må gjøres før nettsiden er klar til å settes i produksjonsmiljøet sitt. Dette innebærer ting som

overføring av databaseleverandør til en som vertstjenesten støtter for rammeverket, endring av test-nøkler til private nøkler for alle API-er systemet benytter seg av, og andre forberedende endringer av systemet. Nye bugs oppstår som må fikses, for eksempel ble det funnet en måte brukere kunne sende flere forespørsler til serveren enn serveren kunne takle.

Dokumenteringsfasen

I gjennom hele prosjektets løp har dokumentering vært et fokus, både for å kunne skrive denne sluttrapporten, men også for å dokumentere produktet som er levert slik at det kan videreføres.

Fra starten av prosjektet førte alle gruppemedlemmene en prosjektdagbok. Etter hver dag med utvikling satt vi oss ned i 10-15 minutter for å fylle ut dagboken med hva vi hadde gjort denne dagen. Fra April begynte vi med rapportskriving. Vi hadde fremdeles mye funksjonalitet å utvikle så vi kodet mandag til torsdag og tok fredagene til kun rapportskriving. Fra midten av April ble gruppen og bedriften enige om at vi kunne fryse utviklingen av systemet frem til rapporten er levert. Dette ga oss et godt utgangspunkt til å få skrevet ferdig rapporten.

Dokumentering av produktet

Alle metoder vi har laget i systemet har en tilhørende kildekodekommentar som forklarer hva metoden tar inn, hva den gjør med dataen som den er gitt, og hva den returnerer. Alle klasser har en kildekodekommentar som forklarer klassens formål, og alle kontrollermetoder forklarer hvilken metode i programvarelageret den bruker.

Disse kildekodekommentarene har vært en viktig del av gruppens oversikt over kode-eierskapet for systemet, et prinsipp som er viktig innen feature-basert utvikling.

I tillegg har disse kildekodekommentarene vært skrevet med formålet om å la utviklerne som skal ta over systemet vite hvordan systemet fungerer, etter at vi overgir systemet til oppdragsgiver.

Dokumentering for prosjektoppgaven

I tillegg til å dokumentere produktet for videreføring og drifting, har gruppen benyttet seg av forskjellige dokumenteringsteknikker for å gjøre selve

rapportskrivingen enklere. Disse forskjellige teknikkene har vært uvurderlige under skrivingen av sluttrapporten samt annen dokumentasjon, som denne prosessdokumentasjonen. Følgende teknikker har gruppen benyttet seg av:

Dagbok

Prosjektdagboken har vært et av de viktigste dokumenteme under skrivingen av sluttrapporten. Den har latt oss se prosjektets livsløp dag for dag, og dette har latt oss utvikle en prosjektdagbok som tar for seg utviklingens løp uke for uke. Gruppens prosjektdagbok er vedlagt slik at den kan leses i sin helhet

I dagboken har vi dokumentert hva vi utviklet "i går", hva som skal utvikles "i dag" og hvor langt vi har kommet i utviklingen. Neste dag tar man utgangspunkt i det som ble skrevet "i går" og utviklingen fortsettes. Det har vært overraskende nyttig å se hva tankene var "i går", for å passe på at koden beholder kvaliteten sin. Dette har vært spesielt nyttig over helgene, da det er overraskende mye man kan glemme fra fredag til mandag.

Møtereferat

I de møtene hvor større beslutninger ble gjort som kunne påvirke hvordan utviklingsprosessen til gruppen var strukturert, ble det gjort møtereferat for å passe på at alle i gruppen fikk den samme beskjeden.

Møter innad i gruppen

Under skriving av sluttrapporten har gruppen sittet sammen i digitalt møte og skrevet for å passe på at dersom deler av utviklingen som ble gjort for lenge siden og kanskje ikke er blitt dokumentert da det ble utviklet, fortsatt blir dokumentert riktig i dette dokumentet.

Oppdragsgiveren og oss

Kommunisering

Det har under hele prosjektet vært et nært samarbeid med oppdragsgiver. Først og fremst gjennom ukentlige møter hvor gruppen setter seg ned og viser frem fremgangen i prosjektet for å få tilbakemeldinger fra oppdragsgiver, og for å diskutere

eventuelle løsninger og prioritere hvilke på features som bør utvikles “ neste uke”. Dersom det har vært små spørsmål som ikke behøver et møte så har det blitt sendt meldinger i messenger-gruppen vår.

Vi har underveis i arbeidet blitt gradvis bedre kjent med oppdragsgiver og deres ønsker, og de har også fått et innblikk i hvordan vi arbeider og hva slags ferdigheter vi har. I tillegg til et mer effektivt samarbeid med tanke på produktets utvikling, har gruppen også blitt kjent nok med bedriften til å ha det gøy på møtene, som har hatt mye å si for gruppens moral, spesielt senere i prosjektets livssyklus.

Samarbeidet

Det har vært viktig for gruppen at oppdragsgiveren skal føle at sin mening er hørt i gjennom utviklingen av produktet. Dette er fordi oppdragsgiver ikke har kunnskap innen IT-utvikling, men skal fortsatt være den som viderefører vårt system til de neste utviklerne som skal vedlikeholde og utbygge det. I tillegg er oppdragsgiveren den som benytter seg av systemets administrator-løsning, og det kan derfor være verdifullt å forstå hvordan denne løsningen er bygget opp, og derfor være med å utvikle den.

Gruppen har heller aldri følt seg presset til å utvikle løsninger vi ikke føler vi har god nok kompetanse for å utvikle. Dersom det har vært et stort tidspress eller at det har vist seg å være uoppnåelig å utvikle flere funksjoner innen tidsrammen har gruppen kunne formidle dette og det har alltid blitt respektert av oppdragsgiver. Det er også verdt å nevne at bedriften alltid har vært meget positiv til å jobbe sammen for å finne en alternativ løsning på funksjonaliteter det ikke er realistisk å få implementert slik de først er foreslått.

Faglig oppsummering

Da vår prosjektoppgave har vært å opprette et produkt for en nyoppstartet bedrift har ikke gruppen fått erfaringen av å utvikle et system i et tradisjonelt, profesjonelt miljø. Da dette ville også vært en verdifull erfaring, har gruppen satt pris på muligheten til å styre og ikke minst ha ansvar for prosjektets fremgang til en så stor grad som vi har hatt. Gruppen har takket være utviklingsmetodikken vår fått mye erfaring i bare det å

ta ansvar for kode vi har produsert, og vært ansvarlige for at det fungerer med alle andre komponenter i systemet når det vises frem til bedriften. Dette har krevd at vi har jobbet jevnt og profesjonelt på eget initiativ, og er en erfaring vi vil sette pris på lenge etter vi har begynt i arbeidslivet.

Muligheten til å få være med på “gründer”-miljøet som en konklusjon til vår skolegang ved OsloMet har vært en flott opplevelse som har gjort vårt samarbeid og kompetanse vesentlig sterkere enn den var før dette prosjektet begynte. Gruppen er stolte av produktet vi har klart å levere, og vi er enige om at alle valg som er gjort i felleskap har vært det riktige valget fra stedet vi har stått.

For å konkludere, er gruppen meget fornøyd med både valg av prosjekt, utførelsen av prosjektet og vi ser lyst på fremtiden som utviklere hvor enn vi ender opp i løpet av de neste årene. Vi har fått uvurderlige erfaringer i systemutvikling, samarbeid og koordinering og vil ta disse med oss i lang tid fremover.

Fremtid

Faktumet at dette produktet vil videreutvikles og vedlikeholdes av en fremtidig IT-avdeling i Way2Easy har vært klart siden før utviklingen begynte. Det har i gjennom hele utviklingsprosessen vært understreket at dokumentering av kildekode vil være av stor betydning for et fremtidig utviklingsteam.

Gruppen har hatt et vesentlig fokus på at produktet skal derfor være feilfritt nok til å kunne driftes av personer uten erfaring frem til inntektene er store nok til å ansette utviklere. Testing av systemet i normal bruk har vært en viktig del av utviklingsprosessen, og en håndtering av såkalte “edge-cases” har vært et selvfølge under opprettingen av nye metoder. Gruppen har også fått forklart systemets forskjellige komponenter til oppdragsgiver og etter at prosjektets sluttrapport er levert vil det også være et viktig dokument for bedriftens nye utviklerteam.

2. Kravspesifikasjon

Way2Easy Documents

Simen Renberg

Adrian Nilssen Nicolaisen

Eskil Nysether

Oskar Ruyter

Prosjektside

<https://oskrrt.github.io>

Presentasjon

Gruppemedlemmer - Gruppe 43

Navn	Studentnummer	Epost
Simen Nordgaard Renberg Gruppeleder	s331179	s331179@oslomet.no
Adrian Nilssen Nicolaisen	s333938	s333938@oslomet.no
Eskil Nysether	s330457	s330457@oslomet.no
Oskar Ruyter	s333993	s333993@oslomet.no

Bedriften

Oppdragsgiver	Prosjekttittel	Kontaktperson/Veileder
Way2Easy AS	Way2Easy Documents	Lars Erik Erlandsen (+47) 452 92 751 way2easyas@gmail.com

Bakgrunn

Way2Easy er en nyoppstartet bedrift med et fokus på produkter som skal hjelpe studenter å få et bedre utbytte av pensum. De har kjøpt domenenavnet dokumarked.no, men har ingen IT-avdeling for å utvikle dette nettsiden.

Det er gruppens oppgave å utvikle nettsiden hvor brukere skal kunne legge sine studiemateriale ut til salgs, og andre brukere skal kunne kjøpe disse materialene. Bedriften ønsker at nettsiden også skal fungere som en samling av alle dokumentene som er blitt kjøpt.

Forord

Denne kravspesifikasjonen er det siste utkastet av prosjektets kravspesifikasjon, og beskriver kravene oppdragsgiveren Way2Easy har til siden dokumarked.no, som skal utvikles av gruppen. Den uttrykker oppdragsgivers og gruppens felles forståelse av kravene har fungert som en kontrakt mellom partene.

Innhold

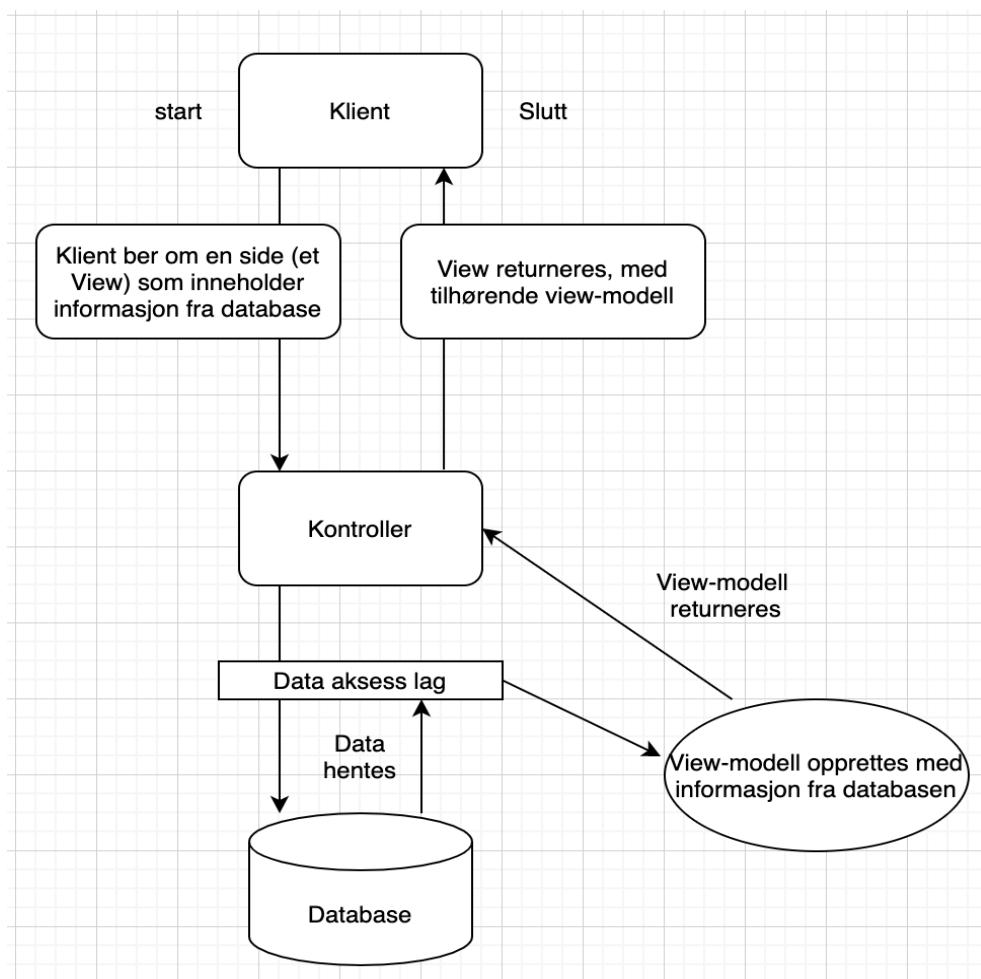
Kort systembeskrivelse	131
Funksjonelle krav	132
Registrering og innlogging	132
Opplastning av filer	133
Handlevogn	133
Kjøp av filer	133
Utbetaling til selgere	133
Søk	134
“Mine dokumenter til salgs” / “Mine kjøpte dokumenter”	134
Admin-side	134
Ikke-funksjonelle krav	135
Sikkerhet	135
Design	135
Rammekrav i systemet	135
Design	135
Lovverk	135
Systemkonstruksjon	136
Sider for alle brukere	136
Sider for innloggede brukere	137
Sider for bedriften (bedriften)	138

Kort systembeskrivelse

Denne løsningen vil kort fortalt tilby en nettside hvor brukere kan selge og kjøpe studiemateriale. Enhver bruker skal kunne både selge og kjøpe studiemateriale, og alltid ha tilgang på dokumentene de har kjøpt og selger.

For å oppnå dette behøver vi en database med fysiske filer omgjort til "byte-blobs" lagret med relasjoner til brukeren som eier dem. I tillegg trengs det å lagres hvilke studiemateriale som er kjøpt av hvem, til hvilken pris.

MVC-rammeverket i ASP.NET Core vil gjøre at henting og visning av nødvendige datafelt fra databasen gjøres gjennom view-modeller, som i tillegg sikrer at kommunikasjonen mot databasen og entitetene lagret der aldri forlater data-aksess-laget som en helhet.



Uformel skisse produsert av Simen N. Renberg for å forklare MVC (Model - View - Controller) rammeverket.

I databasen er det i tillegg tabeller for lovlige filformater (som .pdf, og .xlxs) og lovlige filstørrelser (20mb per dags dato). Disse er opprettet for å passe på at bedriften kan utvide, minke og ellers endre hva de tillater på nettsiden uten å måtte gjøre det programmatisk.

Betalingsløsningen til siden er det eneste som vil behøve refleksjon på hvilken løsning vi bør gå for. Kjøp av studiemateriale er enkelt nok, da vipps er villige til å gi oss tilgang til deres API. Utbetaling fra bedriften til kunder, derimot, er ikke støttet av vippes.

Dermed har vi kommet frem til, sammen med bedriften, to løsninger på utbetaling til nettsidens selgere.

Løsning 1

PayPal har et API som vil la oss programmatisk sende ut utbetalinger til privatpersoner, gitt at de har en paypal-konto.

Løsning 2

Oppdragsgiver er i kontakt med DNB om det er mulig å bruke et tredjeparts regnskapssystem for å automatisk utbetale det som skyldes til selgerne.

Funksjonelle krav

Registrering og innlogging

Det er nødt til å implementeres en innloggings-funksjonalitet på nettsiden, da dokumentene som lastes opp og kjøpes inn skal kunne lagres på brukeren som har gjort kjøpet, eller har lastet opp dokumentet.

.NET Core inneholder et system som heter ASP.NET Core Identity, som man kan velge å inkludere “out-of-the-box” under oppretting av et nytt prosjekt. Dette sikrer at

lagring av brukere overholder standarder for datasikkerhet. Anbefalte brukerinnstillinger, som endring av passord, validering av epostadresser og sletting av bruker er også inkludert.

For å spare tid og for å sikre at det er så sikkert som mulig, velger vi å gå for denne løsningen.

Opplastning av filer

Systemet skal legge til rette for opplastning og lagring av studiemateriale. Det er kun .pdf og .xlsx-formater som skal være mulige å laste opp. Under opplasting av filene vil det i tillegg til tittel, beskrivelse, karakter og pris behøves at en rekke andre detaljer oppgis for å enklere kunne finne igjen dokumentet ved å søke på forsiden. Emnekode, studiested, emnetittel, kategori - som jus, matematikk, informatikk, pedagogikk etc er eksempler på hvordan søkeord vil bli behandlet.

Dersom en bruker laster opp et dokument med en karakter, må dette dokumenteres. For å minske sjansen for juks vil brukeren trenge å sende bedriften et vitnemål med karakteren som gjelder gjennom vitnemålsportalen.no.

Handlevogn

Brukere skal kunne legge flere forskjellige studiemateriale i en handlevogn og kjøpe alle på en gang.

Kjøp av filer

Systemet skal legge til rette for kjøp av dokumentene som er lagt i handlekurven. Bedriften har vært i kontakt med Vipps og har kommet frem til at det vil være mulig for brukere å betale for studiemateriale gjennom Vipps sitt API.

Utbetaling til selgere

Bedriften ønsker at systemet skal kunne automatisk utbetale inntjeningen til privatpersoner som har solgt sitt dokument på nettstedet. PayPal har et API som vi

kan benytte oss av som vil la bedriften samle opp alle inntjeninger i en bankkonto, og en gang i måneden utbetale det som skyldes, minus et prosentvis beløp som går til bedriften.

I tillegg til denne programmatiske løsningen, så er bedriften i kontakt med DNB for å utforske om det er mulig å gi dette ansvaret til et tredjeparts regnskapsprogram.

Søk

Det skal være mulig for brukere å søke etter studiemateriale de leter etter. Oppslag i databasen basert på søkeord definert under opplastning av filer er første steg. Effektive algoritmer for å sikre kapasitet og hurtige oppslag vil bli vektlagt.

“Mine dokumenter til salgs” / “Mine kjøpte dokumenter”

Disse to kravene er slått sammen til ett, da de behøver to veldig like metoder på serveren. Det skal være mulig for brukeren å se hvilke studiemateriale de har ute på siden til salgs, og hvilke studiemateriale de har kjøpt, for å kunne laste de ned på nytt etter å ha kjøpt de. Dette vil være tilgjengelig på brukerens profil.

Det bør også være mulig for en bruker som selger studiemateriale å se sin totale inntjening ved oversikten over deres materiale til salgs.

Admin-side

Det samme nettstedet skal ha en administrator seksjon, med den funksjonalitet man kan forvente, bl. a. mulighet til å uteslunge brukere som har brutt brukervilkårene. I tillegg vil det også være denne siden hvor karaktersatte studiemateriale blir hentet frem for å enten verifiseres eller avvises.

Ikke-funksjonelle krav

Sikkerhet

Alt på nettsiden skal lagres på en sikker måte, og det skal aldri være mulig for en uautorisert bruker å få tilgang til sensitiv data.

Design

Nettsiden skal følge bedriftens retningslinjer angående produktets identitet. Med det menes det at fargevalg, design og utseende skal følge arbeidsgiver sin visjon. For å oppnå dette, vil gruppen være i et møte med arbeidsgiver en gang i uken.

Rammekrav i systemet

Design

Nettsiden skal være responsiv, slik at brukere kan ha en god brukeropplevelse uansett hvilken enhet de aksesserer nettsiden med.

Nettsiden må kunne bruker på alle enheter. også kunne eskaleres og deeskaleres.

Lovverk

Prosjektet skal utvikles i henhold til personopplysningsloven og personvernforordningen (GDPR).

Verk, altså studiemateriale og andre studiemateriale til salgs på nettsiden er beskyttet av åndsverkloven. Studiemateriale som lastet opp til systemet vil bli passordbeskyttet og kryptert for å kopibeskytte det.

Dokumentene skal være automatisk kryptering på opplastede dokumenter(mulig bruke offline) med lokal passordbeskyttelse.

Systemkonstruksjon

Sider for alle brukere

Side	Innhold
Forsiden	Oversikt over dokumentene på siden, søkemotor etc.
Side for hvert enkelt dokument	Siden bruker kommer til når de trykker på et dokument på forsiden, eller et annet hvor tittelen på et dokument dukker (For eksempel, ved "mine dokumenter til salgs").
Side for registrering	Brukere av siden skal ha mulighet til å opprette og logge inn i en profil.
Side for innlogging	Brukere av siden skal ha mulighet til å opprette og logge inn i en profil.
Side for søkeresultater	Brukere skal ha mulighet til å søke i databasen etter dokumenter de er interesserte i.
Side for handlekurv	Brukere skal kunne legge til dokumenter i handlekurv, og derfra betale for de.
FAQ	"Frequently Asked Questions". Brukere skal ha mulighet til å få svar på spørsmål som bedriften mener ofte blir stilt.
"Om oss"	Skal vi bare linke til Way2Easy.no?
Brukervilkår	Alle vilkårene som bruker godkjenner enten det gjelder som selger eller kunde

	etc
Andre profiler sin oversikt over dokumenter til salgs	Når du trykker på et brukernavn skal man se alle dokumentene den brukeren har til salgs
Feilside	Dersom noe kræsjer blir bruker sendt til en trygg feilside

Sider for innloggede brukere

Side	Innhold
Side for å legge et dokument ut til salgs	Brukter fyller inn all nødvendig informasjon om dokumentet og laster det fysisk opp.
Side for "mine dokumenter"	Dokumentene brukeren har kjøpt
Side for "mine dokumenter til salgs"	Dokumentene brukeren har til salgs
Side for å utføre kjøp av dokumenter	Betaler for dokumentet
Brukerinnstillinger	Brukter må kunne endre passordet sitt, slette brukeren sin om de ønsker det, etc etc.
Side for Endre dokument	Brukter skal kunne endre informasjon om et dokument etter de har lastet det opp på siden. Selve dokumentet og karakteren de har oppgitt skal ikke kunne endres.
Side for "Min inntjening"	Brukter skal ha oversikt om

	førstkomende utbetaling fra Way2Easy, i tillegg til en historikk for utbetalinger de har mottatt tidligere.
Side for å betale for promotert plass for deres dokument(er)	Brukerne skal ha mulighet til å betale for promotert plass på siden. Promoterte dokumenter blir satt først i "køen" når brukere søker/filtrerer forsiden.
Side for å rapportere brukere	Brukere skal kunne rapportere andre brukere på nettsiden, slik at bedriften kan (dersom brukeren har brutt sidens brukervilkår) stenge de ut fra siden.
Side for å rapportere dokumenter	Dersom en bruker har kjøpt et dokument som ikke inneholder det som ble lovet fra selger, skal brukeren kunne rapportere dokumentet og få refusjon.
Side for å rapportere omtaler	Dersom språket i en omtale bryter med brukervilkårene til siden, skal bedriften kunne få oversikt over hvilke, og kunne fjerne disse.

Sider for bedriften (admins)

Side	Innhold
Side for å godkjenne eller avvise dokumenters karakter-verifikasi	Selvfklarende.
Side for å kunne uteslunge brukere	Dersom brukere bryter brukervilkårene

	skal de kunne utestenges fra siden.
Oversikt over utestengte brukere	Dersom en bruker har blitt utestengt på grunnlag som viser seg å være feil, skal de kunne få tilbake tilgang til nettsiden.
Oversikt over rapporterte dokumenter	Bedriften skal kunne få en oversikt over hvilke dokumenter som er rapporterte, og hvor mange ganger de er blitt rapportert.
Oversikt over rapporterte brukere	Bedriften skal kunne få en oversikt over hvilke brukere som er rapporterte, og hvor mange ganger de er blitt rapportert.
Oversikt over rapporterte omtaler	Bedriften skal kunne få en oversikt over hvilke omtaler som er rapporterte, og hvor mange ganger de er blitt rapportert.
Oversikt over kommende utbetalinger	<p>Bedriften skal ha en oversikt over hvilke brukere som har solgt dokumenter innen en satt periode (fra siste 15. i måneden, til den neste 15. i måneden, for eksempel 15.01.2021 - 15.02.2021).</p> <p>Hver utbetaling skal ha en frist, og fra denne siden skal bedriften kunne registrere utbetalinger som "vellykket", "mislykket" eller "ubetalt".</p> <p>Siden er i tillegg ment som et regnskaps-verktøy, hvor bedriften kan justere dato-perioden.</p>

Side for å endre roller på nettsiden	Bedriften skal kunne gjøre andre brukere til administratorer.
Side for å oppdatere kommisjonsprosent	Bedriften skal ha mulighet til å justere prosentvis inntjening til bedriften for hvert salg som blir utført på nettsiden.
Side for å oppdatere prisen det koster å promotere et dokument	Bedriften skal ha mulighet til å justere mange NOK det koster å promotere et dokument.

3. Prosjektets feature-liste

Dette er den komplette feature-listen utviklet for bacheloroppgaven våren 2021 ved OsloMet. Hver av disse individuelle funksjonalitetene har blitt dekonstruert og brutt opp i flere, mindre “features” etter at en av gruppemedlemmene har tatt på seg oppgaven, spesielt dersom de er for kompliserte til å kunne implementeres på to uker.

Endring av kravspesifikasjon og/eller innspill fra oppdragsgiver kan medføre at nye features blir definert og utviklet uten at de blir lagt inn i dette dokumentet.

Funksjonalitet anonyme brukere skal ha tilgang på

1. Se forsiden og alle dokumentene på forsiden
2. Se nærmere på ett enkelt dokument dersom de velger det fra forsiden
3. Legge dokumenter i handlekurv
4. Søke etter dokumenter i database
5. Filtrere etter dokumenter i database
6. Se brukervilkår
7. Registrere ny bruker
8. Logge inn i en bruker
9. Se offentlige profiler sin “butikk”

Funksjonalitet for innloggede innloggede brukere

1. Laste opp dokument
2. Kjøpe dokument
3. Endre opplastet dokument
4. Se oversikt over opplastede dokumenter
5. Se oversikt over kjøpte dokumenter
6. Se oversikt over inntjening for hver periode
7. Se sin private profil
8. Promotere et opplastet dokument
9. Rapportere dokument
10. Rapportere bruker

11. Rapportere omtale
12. Få epost dersom brukeren har fått godkjent dokument
13. Få epost dersom brukeren har fått avvist dokument
14. Få epost dersom brukeren har mottatt en utbetaling
15. Få epost dersom utbetaling mislykkes
16. Få epost for å bekrefte kontoen under opprettelse
17. + Alt som anonyme brukere kan

Funksjonalitet for administratorbrukere

1. Utestenge brukere
2. Se oversikt og gi tilbake tilgang til utestengte brukere
3. Godkjenne dokumenter
4. Avvise dokumenter
5. Se rapporter på dokumenter
6. Se rapporter på brukere
7. Se rapporter på omtaler
8. Se antall brukere på siden
9. Se antall dokumenter på siden
10. Se oversikt over kommende utbetalinger til selgere på siden
11. Registrere utbetalinger som "betalt" eller "mislykket"
12. Se oversikt over utbetalinger langt tilbake i tid
13. Endre roller på nettsiden
14. Oppdatere kommisjonsprosent
15. Oppdatere promoteringspris
16. Deaktivere dokumenter etter de er godkjent
17. Aktivere deaktiverte dokumenter
18. Må autentisere seg med en tofaktor-kode sendt på epost.
19. + alt som innloggede brukere kan

4. Prosjektdagbok

Way2Easy Documents / DokuMarked

Denne prosjektdagboken er en sammensatt versjon av utviklernes personlige dagbøker som ble ført ukentlig gjennom prosjektet.

Uke	Arbeid
53 (2020)	<p>Prosjektet ble opprettet i Visual Studio og startet med planlegging av databasestrukturen. Generelle databasemodeller ble skrevet(ved bruk av entity framework). Opprettet litt av de repository-klassene vi trengte og skrev et par metoder, som senere ikke er blitt brukt.</p> <p>Prosjektet som ble opprettet var et prosjekt-template for en React applikasjon med dotnet core API - i backenden.</p> <p>Flere forskjellige repository-er på github ble opprettet, dette på grunn av kluss mellom versjonshåndteringen i visual studio og github desktop.</p>
1 (4. - 10. januar)	<p>På grunn av komplikasjoner med github så ble det siste github-repositoryet opprettet 12.01.21.</p> <p>Mellan 02.01 til 12.01 ble vi enige om at vi skulle gå vekk fra React i front-enden, og heller bruke Dotnet Core 3.1 MVC -</p>

	<p>rammeverket. Dette var fordi vi var bedre kjent med denne teknologien. Begynte på logikken mot databasen, og fikk fullført opprettingen av databasemodellene.</p>
2 (11. - 17. januar)	<p>Utviklet HTML-sidene vi så for oss at nettsiden skulle inneholde, med fokus på de som var viktigst for nettsidens hovedfunksjonalitet. Funksjonalitet for opplasting av dokumenter til databasen i back-enden ble også ferdigstilt, samt validering på opplastnings-skjemaet på serverside.</p> <p>Fikset et problem med .gitignore som gjorde at feil filer ble tracket. La til logikk mot databasen slik at documentInfo.cshtml henter dokument fra databasen når en bruker trykker inn på det. Opprettet DBInit.cs som initialiserer databasen for testformål.</p> <p>Under opplastning av dokumenter til databasen blir filnavnet, beskrivelse og tittel HTML-encoded for å sanitere inputs. I documentInfo.cshtml blir denne teksten HTML-decoded igjen, for å vise spesielle tegn (æøå <>, etc) riktig.</p>
3 (18. - 24. januar)	<p>Funksjonalitet for å legge inn noe i en handlekurv og vise det frem på shoppingCart.cshtml ble implementert.</p>

	<p>Utvidet funksjonaliteten ved å laste opp et dokument og introduserte nye felter på dokument-tabellen. De nye feltene er semester, yearofwriting, placeofstudy og degree. placeofstudy og degree er egne tabeller i databasen slik at de kan oppdateres, og også fordi de hentes ut fra databasen for å populere select-en i skjemaet. Dette for å sikre at brukerne velger riktig sted, og også for å hindre skrivefeil dersom det var et vanlig inputfelt.</p>
	<p>API-controller ble opprettet for å håndtere database-logikk som ikke trengte kommunikasjon med kontroller-metoder ment for navigering av siden (enkle GETs, etc). Skrev inn noen assignmenttypes til DBInit.cs</p>
	<p>AutoMapper ble importert inn i prosjektet. Det er en extension som automatisk mapper objekter av type a til objekter av type b. DTO-modeller ble også opprettet. Arbeidet på _DocumentViewerPartial.cshtml og filtrering av dokumenter ble startet på.</p>
	<p>Brukerprofiler ble utviklet.</p>
	<p>Sider for innlogget brukers kjøpte og solgte dokumenter ble utviklet. Dokumentene</p>

	<p>hentes og vises i <code>_DocumentViewerPartial.cshtml</code> på brukerens side.</p> <p>Sider for brukerinnstillinger ble startet på. En drop-down meny for innlogget bruker som for brukernes egne sider ble laget slik etter oppdragsgivers instruksjer. Sider for de ulike menyvalgene ble opprettet. Drop-down menyen linker til brukerinnstillinger som identity rammeverket har fra før. Denne vil personaliseres senere.</p>
4 (25. - 31. januar)	<p>Fullført klientvalidering på opplastning av dokumenter. UserDTO ble opprettet; ment til PRIVAT profil og PublicProfileDTO, ment til OFFENTLIG profil. Alle Degrees ble skrevet inn i DBInit. Endret på innlogging slik at man logger inn med brukernavn i stedet for epost. Jobbet på forbedret funksjonalitet på PublicProfile.cshtml, men den mangler fortsatt styling.</p> <p>Alle identity feilmeldinger som er generert av IdentityErrorDescription er nå oversatt til norsk gjennom LocalizedIdentityErrors.resx.</p>

	<p>Begynt på logikken for å hente dokument ut fra database-blobben og laste det ned på klient. Også gjort at dokumenter nå kan bli returnert som et byte-array i kontrolleren.</p> <p>Videre utvikling av nedlasting av dokumenter. Usikkert om det skal åpnes i en ny fane eller lastes ned lokalt på PCen.</p> <p>Sortering og filtrering av dokumenter er ferdig implementert. Implementert innlogging med facebook og google ved hjelp av OAUTH 2.</p>
5 (1. - 7. februar)	<p>Funksjonalitet for å kryptere og låse opp dokumenter etter kjøp er ferdig. Filen lastes ned lokalt på brukerens datamaskin. Fikset noen bugs som oppstod.</p> <p>Forhåndsvisning av dokumenter i DocumentInfo.cshtml viser nå første side av dokumentet som et preview gjennom bruk av PDF.js, et open-source bibliotek.</p> <p>Implementerte Administrator-rollen og startet utviklingen av administrator panelet.</p> <p>Funksjonalitet for deaktivering av dokumenter for administratorer ble implementert, samt funksjonalitet for å uteslunge eller "banne" brukere.</p>

	<p>Justert sorterings-logikken og implementert søk.</p> <p>Implementert funksjonalitet for administratorer til å endre roller for andre brukere på administratorpanelet. Fikset en bug ved aktivering av deaktiverte dokumenter.</p> <p>Noen bugs ved innhenting av innlogget brukers kjøpte dokumenter og dokumenter brukeren har til salgs ble fikset.</p>
6 (8. - 14. Februar)	<p>Lagt til funksjonalitet for å laste opp et profilbilde. Bruker samme metode som opplastning av dokumenter. Blob-kolonne i bruker-tabellen.</p> <p>Også lagt inn rapportering av brukere, oversikt over utestengte brukere, fikset litt på utestenging av brukere, og lagt til mulighet for å gi tilbake tilgang til utestengte brukere.</p> <p>Fått inn visning av profilbilder der det passet (documentinfo, publicprofile, privateprofile),</p>

	<p>og fikset litt bugs i logikken for reversering av utestenging.</p> <p>Lagt til oversikt over rapporterte brukere, og et detalje-view med hver eneste rapport for en bruker. Ingen særlig action fra dette viewet enda. Man må søke i ban-viewet for å faktisk uteslunge en bruker som er rapportert mye.</p>
	<p>Lagt til testmetode for å utføre kjøp av dokumentene i handlekurv. Skal IKKE være med i produksjon. Fikset litt på knappen i documentInfo.cshtml dersom man er admin.</p>
	<p>Gjort at knappen for å legge dokument i handlekurv er disabled hvis det er du som har lastet det opp.</p>
	<p>Optimert shoppingcart, fikk OutOfMemoryException fordi session-objektet var for stort. Bruker nå strings i stedet for objects. Stylet også dokumentopplastning.</p>
	<p>Gjorde at oversikten for "Mine kjøpte dokumenter", "Mine dokumenter til salgs" og "Min fortjeneste" er partial-views som lastes gjennom et jquery load()-kall til kontrolleren fra MyWay2Easy.cshtml.</p>
	<p>Fikset utseendet/styling globalt på siden. Implementerte også css custom-properties (variabler) for enkel endring av ting som farger på siden. Implementert funksjonalitet</p>

	for å kunne redigere profilen sin (endre passord, endre epost, endre studieinfo, endre dokument passord osv.).
7 (15. - 21. Februar)	<p>Lagt inn omtaler i DocumentInfo.cshtml, men vi har enda ikke funksjonalitet for å legge inn omtaler gjennom nettsiden. Også lagt til mulighet for admins å slette omtaler.</p> <p>Lagt til mulighet for å rapportere dokumenter etter de er kjøpt</p> <p>Oversatte diverse feilmeldinger på nettsiden til norsk og fikset diverse bugs. Også jobbet med feilhåndtering.</p>
8 (22. - 28. Februar)	<p>Fått inn funksjonalitet for å betale for dokumenter med bankkort gjennom Stripe.</p> <p>Opprettet view etter handlevogn-viewet som tar totalprisen og sender det til Stripe sine servere. Ved å bruke en webhook som lytter etter stripe-aktivitet vil den kunne håndtere business logikk ved eventuelle fullførte kjøp.</p> <p>Dersom en betaling går gjennom blir betalingen lagret i databasen for regnskapsformål og for å utbetale summen til selgeren. Deretter blir dokumentet satt</p>

	<p>som eid av {bruker som kjøpte dokumentet}, som gjør at den er tilgjengelig til nedlastning fra “Min Way2Easy”. Har ikke stylet kjøp vinduet enda. La også til inputfelt for karakter-verifikasiing. Dokumenter kan lastes opp uten karakter, men da kommer to nye input-felter i skjemaet, som bruker må fylle inn med informasjon fra vitnemålsportalen for å verifikasi denne karakteren. Fikset også en bug med profilbilde som gjorde at den runde bildeboksen skalerte feil ved bruk av portrettbilder.</p> <p>Det skal være mulig å laste opp dokumenter fra både høyere utdanning og videregående. Startet på implementering av dette da det per dags dato er det kun mulig å laste opp dokumenter fra høyere utdanning. Pga. dette inneholder skjemaet felter som “emnekode” og alle studieområdene er fra høyere utdanning. Et nytt skjema som lastes etter bruker først har valgt hva slags type dokument som skal lastes opp vil inneholder input-felter relatert til videregående, med linjer, karakterer etc spesifisert til videregående.</p>
9 (1. - 7. Mars)	Funksjonalitet for opplastning av enten videregående- eller høyere utdanningsdokumenter er ferdig. Stylet betalingsvinduet for stripe.

	<p>Implementert funksjonalitet som gjør at sidetall regnes ut i backenden ved å bruke PDFSharp, og antall sider som skal forhåndsvises velges av selgeren (fra 1-10 sider).</p> <p>Funksjonalitet for å endre informasjon om dokumenter som allerede er lastet opp ble implementert.</p> <p>Fått inn et popup-vindu som kommer under opplastning av dokument dersom brukeren ikke har lagret et kontonummer de vil få utbetalinger for salgene deres til.</p> <p>Bruker IBAN.js for å validere riktig IBAN-format dersom de har en internasjonal konto.</p> <p>Utviklet en oversikt for administratorer for hva bedriften skylder hvilken bruker i intervallet 15. til 15. Dette vil feks. si ”hva skylder vi (bedriften) denne personen for salg gjort fra 15.01.2021 til 15.02.2021”. Her skal det være en oversikt over hva som skal utbetalles til hvilken konto, også skal registrering av vellykkede utbetalinger og mislykkede utbetalinger utføres. Dersom en utbetaling mislykkes settes statusen til ”mislykket”, og denne utbetalingen vises øverst i rødt, slik at de kan fikse det. Ellers blir det satt til ”Betalt” i grønt. Det skal også være en datepicker som bestemmer</p>
--	--

	<p>utbetalinger som skal gjøres/har blitt gjort fra “denne datoен” til “denne datoen”.</p> <p>Implementert dark mode på siden.</p> <p>Implementert sortering etter gjennomsnittlig vurdering og at vurdering vises på dokumentene på forsiden. Fikset bugs med dark mode og implementert toggle knapp som skjuler/viser filtreringsmulighetene. Fikset andre diverse bugs som ble oppdaget.</p>
10 (8. - 14. Mars)	Gjort at oversikten over betalinger fra den 15 forrige måned til den 15 denne måneden vises ved første lasting av siden. Mulighet for å registrere en utbetaling som vellykket eller mislykket er ferdig. Startet arbeidet på datepicker.
11 1(5. - 21. Mars)	Endret utseendet på dokument “kortene” som vises på forsiden. Diverse justeringer for en mer responsiv forside ble gjort. Kun aktiverete og verifiserte dokumenter hentes fra db.

	Ga administratorer muligheten til å endre brukerens rolle via søk etter brukernavn.
12 (22. - 28. Mars)	Små bug-fixes og dokumentering
13 (29 Mars - 4. April)	Små bug-fixes og dokumentering
14 (5. - 11. April)	<p>Endret utseendet på dokument "kortene" som vises på forsiden. Diverse justeringer for en mer responsiv forside ble gjort. Også gjort at kun aktiverete og verifiserte dokumenter hentes fra db.</p> <p>Arbeidet med styling av siden.</p>
15 (12. - 18. April)	<p>Implementerte preview av dokumentet på dokument-kortene i <code>_DocumentViewerPartial.cshtml</code>. Også fikset en del bugs med filtreringslogikken.</p> <p>Nå går fokus over på skriving av sluttrapporten.</p>