



Høgskulen
på Vestlandet

BACHELOROPPGAVE

QR-koder — styling, generering og innsikt i hva som påvirker lesbarheten

QR-codes — styling, generation and insight into scannability

Frede Berdal

Magnus Gjørund

Even Sleire

DAT191 - Bacheloroppgave

Fakultet for ingeniør- og naturvitenskap (FIN)

Veileder: Sven-Olai Høyland

23.05.2022

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 10.

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> QR-koder — styling, generering og innsikt i hva som påvirker lesbarheten	<i>Dato:</i> 23.05.2022
<i>Forfatter(e):</i> Frede Berdal, Magnus Gjøsend, Even Sleire	<i>Antall sider u/vedlegg:</i> 52
	<i>Antall sider vedlegg:</i> 9
<i>Studieretning:</i> Dataingeniør/Informasjonsteknologi	<i>Antall disketter/CD-er:</i> 0
<i>Kontaktperson ved studieretning:</i> Sven-Olai Høyland	<i>Gradering:</i> Ingen
<i>Merknader:</i>	

<i>Oppdragsgiver:</i> Favrit AS	<i>Oppdragsgivers referanse:</i>
<i>Oppdragsgivers kontaktperson:</i> Aslak Hollund	<i>Telefon:</i> 47304656

<i>Sammendrag:</i> Rapporten beskriver prosessen for utviklingen av en prototype til et QR-kodesystem for Favrit. Selskapet er et teknologiselskap som leverer programvare til serveringsbransjen. Favrit tilbyr en løsning hvor blant annet kundenes gjester skanner en QR-kode og får tilgang til en digital meny. Prototypen tillater styling og generering av QR-koder. Den gir også innsikt i hva som påvirker en QR-kode sin lesbarhet. I tillegg er det utviklet en algoritme som gir tilbakemelding på lesbarheten til stylingen.

Stikkord:

QR-kode	QR-kodestyling	Lesbarhet til QR-koder
---------	----------------	------------------------

FORORD

Vi ønsker å rette en stor takk til Favrit for oppgaven, representant fra Favrit - Aslak Hollund og alle interessenter hos Favrit som har vært involvert i prosjektet. Disse har i stor grad vært med å forme resultatet gjennom gode tilbakemeldinger underveis.

Even Sleire er ansatt hos Favrit, som er oppdragsgiver. Han har tidligere jobbet for supportavdelingen og besitter derfor forkunnskaper om dagens løsning for QR-kode styling og generering.

Vi ønsker å takke Sven-Olai Høyland, intern veileder, for gode råd og samarbeid gjennom prosjektet.

I tillegg ønsker vi å takke alle som har testet og evaluert produktet. Disse har gitt verdifulle tilbakemeldinger.

Frede Berdal, Magnus Gjørund, Even Sleire

Ordliste/Akronymer

HVL - Høgskulen på Vestlandet

QR - Quick Response

REST-API - Representational state transfer Application Programming Interface

URL - Uniform Resource Locator

Branding - Merkevarerbygging

Rebranding - Endring av navn, logo og fargeprofil

Styling - Redigering av komponenters utseende

Favrit - Oppdragsgiver, tilbyr en ende-til-ende plattform for selskaper innenfor serveringsbransjen

Kunde - Selskaper innenfor serveringsbransjen (Restaurant, bar, hotell osv.)

Gjest - Brukerne av tjenesten ute hos kundene

Skanne - Å lese av en QR-kode

QR-code-generator - Tredjepartsløsningen Favrit bruker for styling og generering av QR-koder

Link-redirect - En tjeneste som videresender en bruker fra en URL til en annen

Frontend - Klientsiden av en webapplikasjon

Backend - Serversiden av en webapplikasjon

BLOB - Binary large object

MVP - Minimum viable product

JSX - JavaScript XML

INNHALDSFORTEGNELSE

1 INNLEDNING	1
1.1 KONTEKST	1
1.2 MOTIVASJON	1
1.3 PROSJEKTEIER	2
1.4 Problembeskrivelse og mål	3
1.5 Oppbygging av rapporten	4
2 PROSJEKTBEKRIVELSE	5
2.1 PRAKTISK BAKGRUNN	5
2.2 AVGRENSNINGER	5
2.3 RESSURSER	6
2.4 LITTERATUR OM PROBLEMSTILLINGEN	7
2.4.1 Grunnleggende om QR-koder	7
2.4.2 STATISK VS DYNAMISK QR-KODE	9
3 DESIGN AV PROSJEKTET	10
3.1 FORSLAG TIL LØSNING	10
3.1.1 Alternative løsninger for styling av QR-kode	10
3.1.2 Alternative løsninger for generering av QR-kode	10
3.1.3 Alternative løsninger for lagring av QR-kode	11
3.1.4 Alternative løsninger for å få innsikt i lesbarhet	11
3.2 DISKUSJON AV ALTERNATIVENE	12
3.2.1 Styling	12
3.2.2 Generering	12
3.2.3 Lagring	12
3.2.4 Innsikt i lesbarhet	13
3.3 VALGT LØSNING	13
3.4 Valg av teknologi	15
3.4.1 Frontend	15
3.4.2 Backend	16
3.5 PROSJEKTMETODIKK	17
3.5.1 UTVIKLINGSMETODIKK	18
3.5.2 PROSJEKTPLAN	21
3.5.3 RISIKOVURDERING	22
3.6 Evalueringsplan	23
4 LØSNING	25
4.1 Arkitektur	25
4.2 Styling av QR-koder	26

4.3 GENERERING AV QR-KODER	30
4.4 INNSIKT I LESBARHET	31
4.4.1 Utstyr	31
4.4.2 Forsøksoppsett	32
4.4.3 Ensfarget QR-kode	33
4.4.4 QR-koder med tydeligere positioning detection markers	34
4.4.5 QR-koder med logo	35
4.4.6 Konklusjon	35
4.4.7 Algoritme	36
4.4.8 Informasjonsside	37
5 EVALUERING OG RESULTAT	38
5.1 Evalueringsmetode	38
5.1.1 Evaluering av produktutvikling	38
5.1.2 Evaluering av kvalitet	39
5.2 Evaluering av sluttprodukt	40
5.3 Prosjektresultat	41
5.4 Prosjektgjennomføring	42
5.4.1 Arbeidsmetodikk - Lean startup	43
5.4.2 Prosjektplan - Critical Chain	43
5.4.3 Samarbeid	43
6 DISKUSJON	45
6.1 Tilnærming	45
6.2 Konsekvenser av tilnærming	46
6.3 Drøfting av resultat	46
6.4 Refleksjon	47
7 KONKLUSJON OG VIDERE ARBEID	48
7.1 Konklusjon	48
7.2 Videre arbeid	49
8 REFERANSER	51
9 VEDLEGG	53
9.1 Risikoanalyse	53
9.2 Brukertesting instruksjer	54
9.3 Brukertesting spørreundersøkelse	54
9.4 Resultat av brukertesting	55
9.5 Prosjektbeskrivelse	56
9.6 Resultater fra eksperiment	59

1 INNLEDNING

Følgende delkapitler skal ta for seg kontekst, motivasjon og introduksjon av prosjekteier. Videre presenteres prosjektets problemstilling og mål.

1.1 Kontekst

Favrit er et teknologiselskap som tilbyr en ende-til-ende plattform for serveringsbransjen. Favrit har i dag 96 ansatte, 900 kunder og er tilgjengelig i fire ulike land (Favrit, 2022). Kundene deres er bedrifter som hoteller, restauranter og barer. Selskapet tilbyr en løsning hvor kundenes gjester skanner en QR-kode og får tilgang til en digital meny der de legger inn bestilling og gjennomfører betaling gjennom tjenesten. Kunden får gjestens bestilling i en digital kø. Køen gir kunden oversikt over alle bestillingene. Favrit gir også kunden mulighet til å opprette og redigere menyer, samt integrasjon mot regnskapssystemer. Kundene får tilsendt QR-koder som gjestene bruker for å aksessere tjenesten.

I dag brukes en tredjepartsløsning for å style, generere og lagre QR-kodene. Favrit ønsker en løsning som tilbyr de samme mulighetene for styling som dagens løsning, men samtidig ha eierskap til produktet. Prosjektet vil derfor gå ut på å lage en prototype for dette.

1.2 Motivasjon

QR-kodene er et viktig element for Favrit og for deres kunder. QR-koden er gjestens første møte med tjenesten. Favrit sin markedsføringsavdeling, som vil bli en av brukerne av systemet, bruker QR-koder i markedsføringssammenheng. Av den grunn er det viktig at QR-kodene er estetiske og unike, slik at gjestene assosierer QR-kodene med Favrit. Markedsføringsavdelingen har i dag ansvar for stylingen av QR-kodene. Stylingen omfatter blant annet redigering av farger, mønster, størrelse og legge til logo. Gruppens prototype må derfor legge til rette for slik styling.

Ett av problemene rundt dagens løsningen er at systemet ikke gir tilbakemelding på hvor lesbar en QR-kode er etter styling. Favrit har vært gjennom en rebranding som har ført til ny styling på QR-kodene. De er usikker på om den nye fargen, det nye mønsteret eller logoen har gjort QR-koden vanskeligere å lese enn tidligere. Dette satte fokus på et behov som ikke dagens løsning

dekker. Favrit ønsker dermed et system som gir en form for evaluering av stylum QR-kode. Dette skal sikre god lesbarhet på QR-kodene som genereres. Lesbarhet vil si en QR-kode sin evne til å bli skannet.

Figurene under viser hvordan QR-kodene var stylum før og etter rebranding (se Figur 1.1 og Figur 1.2). Favrit sitt tidligere merkevareravn var Ordr.



Figur 1.1. Norefjell Golfklubb (2021) BETALING MED ORDR. Tilgjengelig fra: <https://www.norefjell-golf.no/aktuelt/nytt-selvbetjent-betalingsystem> (Hentet: 8. mars 2022).



Figur 1.2. Favrit (2022) Skann koden og prøv. Tilgjengelig fra: <https://web.favrit.com/> (Hentet: 8. mars 2022).

Et annet problem er at Favrit sin oppetid er avhengig av en tredjepartsløsning. Dersom denne løsningen er nede, fører det til at QR-kodene ute hos kundene ikke fungerer ettersom dette er dynamiske QR-koder. Det er ønskelig at dagens system er uavhengig av eksterne løsninger som Favrit ikke har kontroll over. Dette er hovedmotivasjonen for å eie et eget system.

1.3 Prosjekteier

Favrit vil ha alle rettigheter til løsningen. Hensikten med prosjektet er å utvikle en prototype som Favrit kan overta og videreutvikle. Gruppen har ikke eierskap over løsningen, bortsett fra rettighetene til å demonstrere løsningen i rapporten og på EXPO Bergen. EXPO Bergen er et arrangement som arrangeres årlig på campus Bergen der avgangstudentene ved HVL presenterer bachelor- og master-prosjektene sine (HVL, 2022a).

1.4 Problembeskrivelse og mål

Problemstilling

Under COVID-19 pandemien oppstod det et nytt fenomen i serveringsbransjen – gjestene trengte ikke lengre å forholde seg til servitører for å bestille. Ved hjelp av en QR-kode får gjestene tilgang til en digital meny, og kan legge inn bestillinger på egenhånd. For Favrit er gjestens inntrykk av QR-koden viktig fordi dette er styrende for oppfatningen av tjenestes kvalitet. Det er derfor viktig at QR-koden er estetisk fin og har god lesbarhet. Favrit bruker i dag en tredjepart for styling og generering. Denne løsningen skaper noen utfordringer og dekker ikke bedriftens behov tilstrekkelig.

Det første sentrale problemet er at markedsføringsavdelingen ikke får en form for tilbakemelding på lesbarheten til QR-kodene når de blir stylet. De vet altså ikke om de forverrer lesbarheten til QR-koden når de tilpasser den.

Et annet sentralt problem er at ved å bruke dagens løsning er Favrit avhengig av en tredjepart. Avhengigheten kan føre til at Favrit sine QR-koder ikke lenger fungerer når tredjepartens systemer er nede. For å ivareta Favrit sine interesser skal denne bacheloroppgaven bli besvart med hensyn til følgende problemstilling:

Hvordan kan Favrit få eierskap over sine QR-koder, og samtidig sikre unikt design og god lesbarhet?

Mål

Målet med prosjektet er å utvikle en prototype av et system for QR-kodegenerering som dekker de viktigste behovene til Favrit sin markedsførings- og supportavdeling. Gruppen har definert de tre viktigste behovene gjennom brukerintervju med markedsføringsavdelingen. Disse behovene er:

- Mulighet til å style utseende til en QR-kode
- Mulighet til å generere en QR-kode med gitt styling
- Få innsikt i hva som påvirker lesbarheten til en QR-kode

Prosjektet vil bestå av tre hoveddeler. Hver del skal ta for seg ett av behovene. Disse tre hoveddelene vil fungere som delmål for prosjektet.

Forskningsspørsmål

Det skal være en sammenheng mellom forskningsspørsmålene, mål og problemstilling. Gruppen har formulert følgende forskningsspørsmål for å hjelpe å besvare delmålene og problemstillingen.

- Hvordan kan løsningen tilby ulike muligheter for styling?
- Hvordan påvirker de ulike elementene innenfor styling lesbarheten til en QR-kode?

1.5 Oppbygging av rapporten

Rapporten er bygd opp av syv kapitler som omtaler prosessen rundt utviklingen av QR-kodesystemet. Avslutningsvis oppgis referanser og vedlegg.

Kapittel 1 *Innledning* tar for seg kontekst, motivasjon og presentasjon av prosjekteier. Prosjektets problemstilling og mål legges fram.

Kapittel 2 *Prosjektbeskrivelse* skal presentere prosjektets opprinnelige krav, avgrensninger, og hvilke ressurser prosjektet benytter seg av. I tillegg blir det gitt en grunnleggende innføring i oppbyggingen av QR-koder.

Kapittel 3 *Design av prosjektet* skal presentere og diskutere alternative løsninger til problemstillingen. I tillegg vil kapittelet ta for seg prosjektmetodikk og evalueringsplan.

Kapittel 4 *Design og utvikling* skal beskrive prototypen i detalj ved å gjøre rede for brukergrensesnittet og den tekniske løsningen.

Kapittel 5 *Evaluering og resultater* skal ta for seg hvordan resultatene evalueres, hvilke metoder og framgangsmåter som er brukt og en konklusjon av evalueringen.

Kapittel 6 *Diskusjon* skal ta for seg tilnærmingen til hvordan prototypen skulle bli utformet og konsekvensene av dette. Kapittelet skal også drøfte resultatene av prototypen.

Kapittel 7 *Konklusjon og videre arbeid* skal ta for seg konklusjon av prosjektet og forslag til videre arbeid.

2 PROSJEKTBEKRIVELSE

De følgende delkapitlene presenterer prosjektets opprinnelige krav, avgrensninger, og hvilke ressurser prosjektet benytter seg av. I tillegg blir det gitt en grunnleggende innføring i hvordan QR-koder er bygd opp.

2.1 Praktisk bakgrunn

Oppdragsgiver har satt initielle krav i prosjektbeskrivelsen ([se vedlegg 9.5](#)). Disse kravene har som hensikt å skape et utgangspunkt for prosjektet. Det er ellers frihet rundt kravene og hvordan problemet løses. Dette er de initielle krav fra prosjektbeskrivelsen:

- Løsningen må kunne lage QR-koder som representerer strenger.
- Det må være mulighet for å lage QR-koder både gjennom et brukergrensesnitt og gjennom et API.
- Det bør være et skille mellom public-API (som brukes av andre klienter eller system hos oppdragsgiver) og internal-API (som brukes i bachelorgruppens applikasjon).
- Løsningen må kunne ta i mot kunde-ID og lage QR-koder som linker direkte til lokasjonen.
- Løsningen bør kunne lage QR-koder som kan styles etter Favrit sitt behov.
- Løsningen bør kunne gi en vurdering av valgt styling og si noe om lesbarheten til QR-koden.

Gruppen vil forholde seg til de initielle kravene som et rammeverk for funksjonalitet som systemet skal inneholde. Dersom det avdekkes nye behov hos interessentene, som krever justeringer av krav, skal gruppen ta høyde for dette og tilpasse systemet etter de nye behovene.

2.2 Avgrensninger

Systemet skal fungere som en prototype. Det skal ikke implementeres mulighet for generering av dynamiske QR-koder ettersom disse er avhengig av en link-redirect tjeneste. Link-redirect er en tjeneste som videresender en bruker til ønsket URL. Dynamiske QR-koder blir redegjort for i [Kapittel 2.4.2](#). Grunnet at systemet skal fungere som en prototype og et internt system vil det ikke bli tatt høyde for universell utforming og globalisering.

2.3 Ressurser

For å gjennomføre prosjektet trenger gruppen tilgang til enkelte ressurser. Favrit har kontorer i Oslo. Av den grunn kommer gruppen til å bruke HVL Campus Kronstad som arbeidsplass. Alt utstyret er gruppens eget.

Ved behov for innhenting av ny kunnskap skal gruppen hovedsakelig benytte seg av forskningsartikler, dokumentasjon, og andre relevante ressurser. Dersom det blir nødvendig skal gruppen utføre eksperimenter.

Krav og behov skal innhentes fortløpende gjennom møter med interessenter. Gruppen skal i tillegg kontakte intern og ekstern veileder ved behov. Dersom prosjektet vil omhandle maskinlæring, er det ønskelig å komme i dialog med professor Jerry Chun-Wei Lin ved HVL, som er anerkjent på dette området (HVL, 2022b).

Samarbeidsverktøy

Ulike verktøy som er benyttet for prosjektets utvikling og kommunikasjon er oppramset nedenfor.

Messenger - Kommunikationsverktøy for kommunikasjon innad i gruppen.

Slack - Kommunikationsverktøy for kommunikasjon med oppdragsgiver.

Github - Versjonskontrollverktøy brukt for å ha tidligere versjoner av kildekode tilgjengelig og dele koden mellom gruppemedlemmene.

Microsoft Teams - Møteverktøy brukt for å ha digitale møter med veileder, oppdragsgiver og andre interessenter.

Google documents - Samkjøringsverktøy for at gruppen kan skrive rapporter og andre dokumenter sammen på tvers av datamaskiner.

Miro - Verktøy for å lage visuelle komponenter til rapport og dokumenter. For eksempel visualisere konsepter, ideer og løsninger.

Trello - Verktøy for å systematisere og visualisere oppgaver som skal gjøres, oppgaver som blir utført og oppgaver som er fullførte, ved hjelp av en kanban tavle.

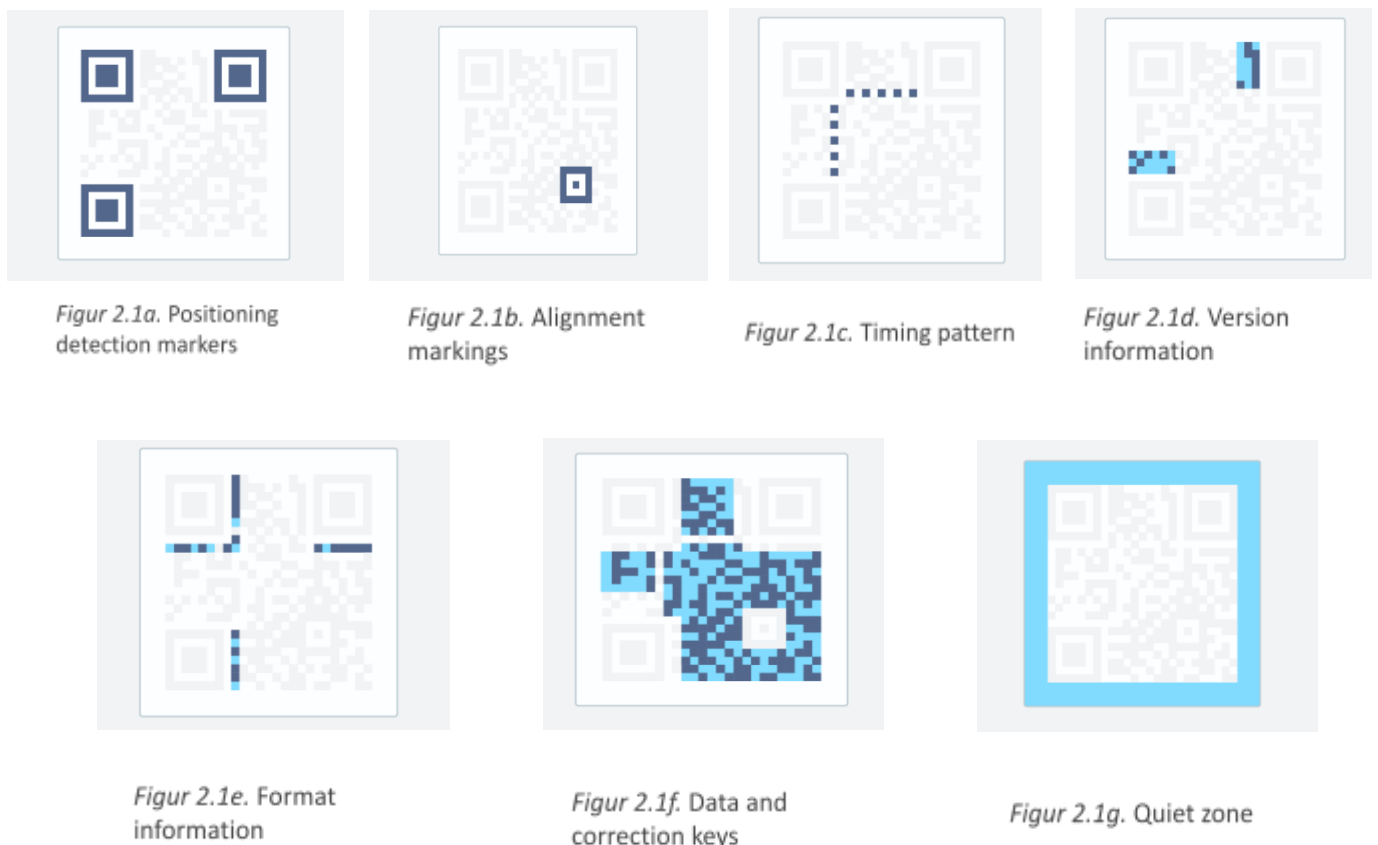
2.4 Litteratur om problemstillingen

Dette kapittelet har som hensikt å skape innsikt i domenet og gi en grunnleggende forståelse for tematikken prosjektet tar for seg.

2.4.1 Grunnleggende om QR-koder

En QR-kode er en todimensjonal strekkode som inneholder data i form av bits (Digital didaktikk, 2022). Det gjør at en QR-kode kan inneholde en betydelig større mengde data enn en strekkode. En QR-kode kan inneholde opptil 7089 tall eller 4296 tegn, inkludert tegnsetting og spesialtegn. Antall datamoduler vil øke etter mengden data som lagres i QR-koden.

Moderne QR-koder er bygget opp av syv ulike deler (QR Code Generator, 2022). Hver del illustreres i Figur 2.1. Delene skaper et pixel-mønster som kan minne om et puslespill. Hver av disse delene har en spesifikk hensikt.



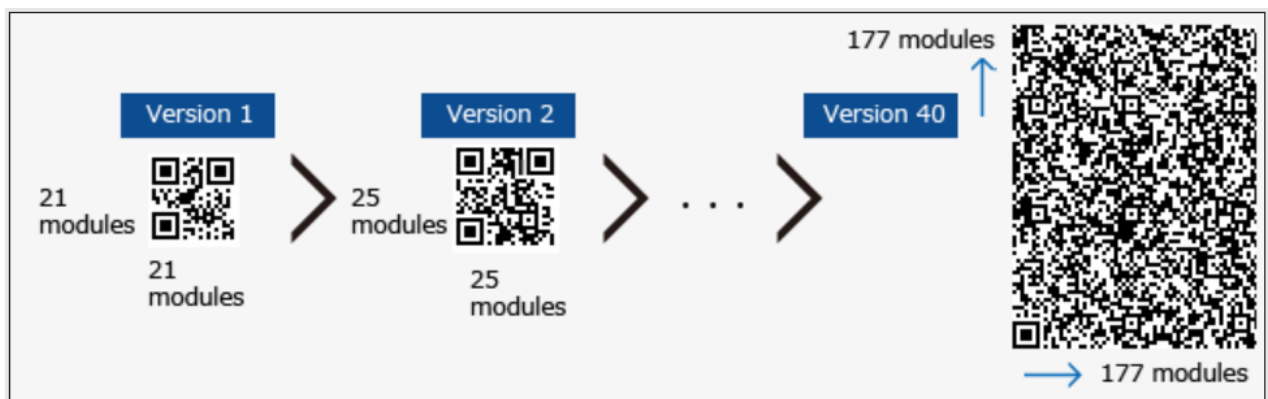
Figur 2.1. QR Code Generator (2022) *The anatomy of a QR Code* Tilgjengelig fra: <https://www.qr-code-generator.com/qr-code-marketing/qr-codes-basics/> (Hentet: 10. januar 2022).

Positioning detection markers er tre kvadrater som er lokalisert i hjørnene av QR-koden (se Figur 2.1a). Hensikten er at en telefon skal raskt gjenkjenne at det er en QR-kode. Årsaken til at det er nøyaktig tre kvadrater er for at telefonen skal kunne identifisere hvilken retning den skal lese QR-koden.

Alignment markings har som hensikt å gjøre det lettere å skanne QR-koder som befinner seg på en buet overflate (se Figur 2.1b). Hvis QR-koden inneholder en større mengde data vil QR-koden ha flere av disse.

Timing pattern har som hensikt å definere hvor matrisene som inneholder data befinner seg i QR-koden og størrelsen på dem (se Figur 2.1c).

Version information sier noe om hvilken versjon QR-koden er (se Figur 2.1d). Det brukes i dag 40 ulike versjoner av QR-koder. Versjonen sier noe om antall datamoduler QR-koden inneholder (se Figur 2.2).



Figur 2.2. Qrcode.com (2022) *Information capacity and versions of the QR code*. Tilgjengelig fra: <https://www.qrcode.com/en/about/version.html> (Hentet: 10. mars 2022).

Format information inneholder informasjon om *error tolerance* og *data mask pattern* (se Figur 2.1e). Dette gjør det enklere for en telefon å skanne QR-koden til tross for skade på koden.

Data and error correction keys er datamoduler som inneholder alt av lagret informasjon i QR-koden (se Figur 2.1f). Her finnes også informasjon om hvilket nivå av *error correction* QR-koden er satt til. *Error correction* nivået sier noe om hvor skadet en QR-kode kan være og

fortsatt fungere etter sin hensikt. Dette gjør det mulig å skanne en QR-kode til tross for en rift eller søl på koden.

Det finnes fire nivåer av *error correction*:

Nivå L – opptil 7% skade

Nivå M – opptil 15% skade

Nivå Q – opptil 25% skade

Nivå H – opptil 30% skade

Quiet zone er en sone rundt QR-koden som har til hensikt å skille koden fra omgivelsene (se Figur 2.1g).

2.4.2 Statisk vs dynamisk QR-kode

Systemet skal fungere som en prototype, og tar derfor kun utgangspunkt i generering av statiske QR-koder. Oppdragsgiver ønsker at det legges til rette for at systemet skal generere dynamiske QR-koder på et senere tidspunkt.

Strengen statiske QR-koder peker på kan ikke endres etter de er generert. Dynamiske QR-koder inneholder ofte en kortere URL med en ID. Ved scanning av en dynamisk QR-kode blir man sendt til en link-redirect tjeneste. ID-en som er lagret i QR-koden vil peke på en URL man videresendes til. Fordelen med det er at URL-en lagret i link-redirect tjenesten kan endres på.

3 DESIGN AV PROSJEKTET

De følgende delkapitlene skal presentere og diskutere alternative løsninger til problemstillingen. I tillegg vil kapittelet ta for seg prosjektmetodikk og evalueringsplan.

3.1 Forslag til løsning

3.1.1 Alternative løsninger for styling av QR-kode

Gruppen har tenkt å lage en løsning for styling av QR-koder som minner om dagens løsning. Ved hjelp av brukerintervju med markedsføringsavdelingen har gruppen kartlagt at dagens løsning dekker nesten alle behov. Det som mangler er at bakgrunnsfargen ikke kan endres og at man ikke får tilbakemelding på lesbarheten. Stylingen skal omfatte mulighetene til å endre forgrunnsfarge, bakgrunnsfarge, legge inn logo, justere størrelsen til logo, endre fysisk størrelse på QR-koden og endre formen på datamodulene. Etter ferdig styling kan brukeren lagre den som en mal slik at framtidige QR-koder kan se tilsvarende ut.

En løsning vil være å implementere et eksternt bibliotek i klienten som gir brukeren muligheten til å gjøre nødvendige endringer på QR-koden.

Et annet alternativ vil være å utvikle et bibliotek som er tilpasset behovene til markedsføringsavdelingen.

3.1.2 Alternative løsninger for generering av QR-kode

Dette underkapittelet tar for seg alternativene rundt generering av QR-koder, samt hvordan de skal bli hentet etter generering. Gruppen har identifisert to mulige løsninger.

Det første alternativet er å generere ny QR-kode etter behov. Dette blir gjort ved at systemet tar inn en kunde-ID eller URL, for så å generere en QR-kode som peker på kundens meny. Denne prosessen blir gjentatt hver gang det er behov for å hente eller generere ny QR-kode. En slik løsningen er mindre kompleks siden det ikke er behov for å lagre data i en database.

Det andre alternativet går ut på å lagre QR-koden i en database, med hensikt om å gjenbruke QR-koden. Med denne løsningen er det tenkt at en "last ned" knapp vil kunne hente QR-koden fra databasen.

3.1.3 Alternative løsninger for lagring av QR-kode

Utfra hvilken løsning gruppen velger for generering av QR-koder, kan det oppstå ulike utfordringer. Hvis gruppen velger å lagre data i en database, kan det bli gjort på flere forskjellige måter. Dette underkapittelet tar kun for seg hvilke data som er relevant å lagre i databasen for å kunne hente QR-koden på et senere tidspunkt.

Gruppen har kommet frem til to alternative løsninger på hva som bør lagres. Det som skiller løsningene er hvilke data som lagres som igjen påvirker hvordan man skal hente QR-kodene.

Det ene alternativet er å lagre hele QR-koden som et BLOB (Binary large object) i databasen. Det andre alternativet er å lagre en URL og/eller en kunde-ID. Neste gang support skal hente QR-koden, henter de strengen, for så å generere QR-koden. Begge handlingene blir gjort ved å trykke på "last ned" knappen.

3.1.4 Alternative løsninger for å få innsikt i lesbarhet

Det finnes flere alternativer for å løse utfordringene rundt innsikt i lesbarhet. Den ene løsningen går ut på å lage et informasjonsskriv som tydelig forklarer hvordan ulike komponenter ved styling kan påvirke lesbarheten. Det er også ønskelig at gruppen klarer å samle nok data om ytre faktorer som kan påvirke lesbarheten. For eksempel hvilket materiale QR-koden er på? Hvordan er lyset i rommet? Er QR-koden laminert, med tanke på gjenskinn? Informasjonsskrivet er tenkt som et støttedokument til markedsføringsavdelingen for å få generell innsikt i hva som påvirker lesbarheten.

Neste løsning er å gi tilbakemelding på hvilke sentrale komponenter tilhørende en QR-kode som er dekket eller ikke dekket basert på gitt styling. Hver komponent til QR-koden vil få en tilbakemelding. Løsningen vil ta inspirasjon fra situasjoner der man lager et nytt passord - er passordet svakt, nøytralt eller sterkt? Utfra denne tilbakemeldingen kan markedsføringsavdelingen gjøre endringer på stylingen.

En annen løsning er å utvikle en algoritme som gir en indikasjon på helhetlig vurdering av lesbarheten til gitt styling. Algoritmen skal basere seg på de sentrale egenskapene tilhørende en QR-kode. Det er tenkt at algoritmen skal gi en poengsum, og basert på poengsummen gi en indikasjon på om den er svak, nøytral eller sterk.

Den siste løsningen vil være å bruke maskinlæring til å gi tilbakemelding på lesbarheten til gitt

styling. For at maskinlæringsmodellen skal gi tilfredsstillende tilbakemelding er det nødvendig å samle stor nok mengde data for å trene opp modellen.

3.2 Diskusjon av alternativene

3.2.1 Styling

De alternative løsningene for styling av QR-koder har ulike fordeler og ulemper. Ved å implementere et eksternt bibliotek kommer gruppen til å spare betydelig mengde tid. Ulempen med å ta i bruk et slikt bibliotek kan være at det blir en begrenset mulighet til styling, slik at Favrit sine viktigste behov ikke blir møtt. Fordelen med å utvikle et bibliotek selv er at biblioteket kan bli utviklet utfra behovene, slik at Favrit får de stylingmulighetene de ønsker. Dessuten gir det mulighet til videreutvikling for å møte framtidige behov. Utviklingen av et slikt bibliotek fra grunnen av kommer til å være svært tidkrevende. I tillegg er det en risiko rundt at biblioteket ikke blir ferdig utviklet, eller utviklet godt nok.

3.2.2 Generering

Det er fordeler og ulemper med begge alternativene til generering av QR-koder. Ved lagring av QR-kodene i en database kan en unngå å måtte regenerere QR-kodene. Dette er en mer tidkrevende implementasjon enn det andre alternativet. Fordelen med å generere ny QR-kode etter behov er at systemet blir mindre komplekst. Som følger av dette kan det i større grad fokuseres på andre områder som kan skape stor verdi, for eksempel innsikt i lesbarhet.

3.2.3 Lagring

Ved å lagre QR-koden som et BLOB i databasen trengs det ikke å generere ny QR-kode. Ulempen med å lagre QR-koden som BLOB er at det vil kreve stor mengde lagringsplass i databasen. Hvis det skjer en endring i styling på QR-kodene, vil det være behov for å regenerere alle eksisterende QR-koder med gitt ny styling, som igjen vil kreve stor mengde lagringsplass. Det andre alternativet, hvor det lagres en kunde-ID, vil kreve mindre lagringsplass i databasen, men det vil være et behov for å regenerere QR-koden. Det koster lite å regenerere QR-koder, det ansees ikke som et stort problem.

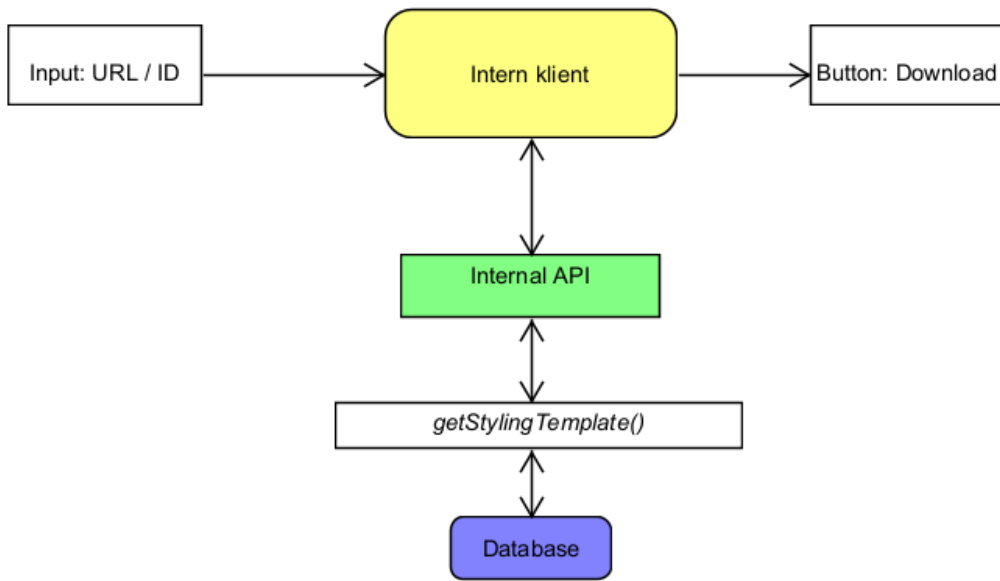
3.2.4 Innsikt i lesbarhet

For å gi best mulig tilbakemelding på lesbarheten er det ønskelig å implementere alle fire alternativene. Det har vært utfordrende å finne tidligere forskning på hva som påvirker lesbarheten til en QR-kode. Som følge av det må gruppen utføre selvstendig forskning og utvikle en algoritme utfra resultatene. Det er derfor stor usikkerhet rundt omfanget av arbeidet og hvor langt gruppen kommer med denne funksjonen. Gruppen har som forventning å klare å implementere minst to av alternativene.

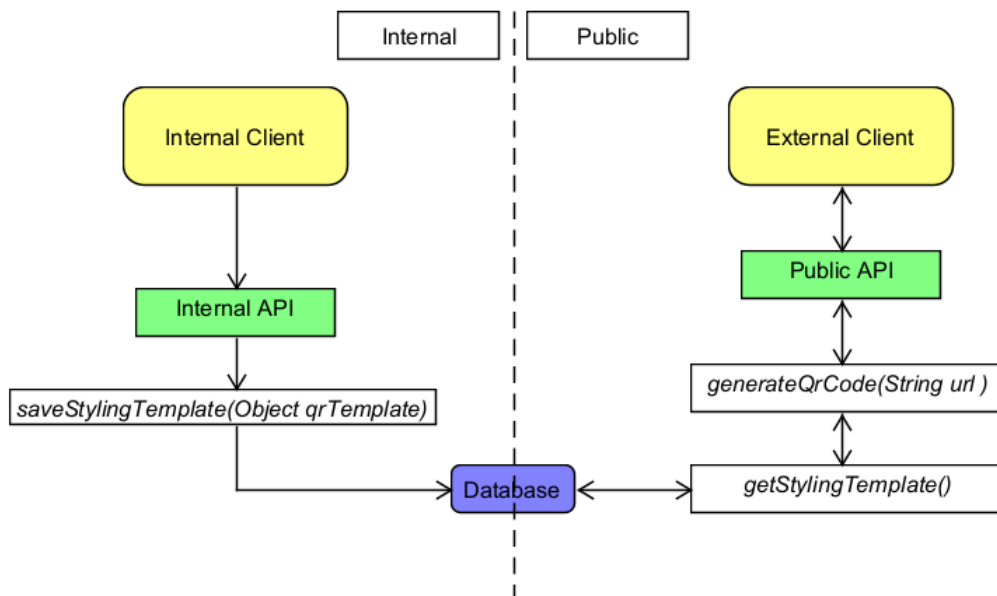
3.3 Valgt løsning

Løsningen for styling av QR-koder vil implementere biblioteket `qr-code-styling`, som er et JavaScript-basert bibliotek. Denne løsningen vil i utgangspunktet dekke alle de viktigste behovene til markedsføringsavdelingen i Favrit. Ved lagring av en stilet mal vil alle attributtene tilknyttet malen bli lagret i en database. QR-kodene vil bli generert etter behov ved å benytte seg av bibliotekets nedlastningsfunksjon. Systemet skal generere QR-koder gjennom et eget brukergrensesnitt (se Figur 3.1) og gjennom et API slik at andre klienter kan generere QR-koder. Følgelig skal det implementeres et internal-API som kun brukes av systemets klient og et public-API som kan brukes av eksterne klienter (se Figur 3.2).

Begge API-ene vil bli implementert som REST API. Dette er API som må følge visse krav til utforming (IBM, 2021). Kravene er at klient og server skal være uavhengig av hverandre, hvert endepunkt skal ha sin egen URI (Uniform Resource Identifier) og det skal være en lagdelt arkitektur. API-ene skal i tillegg være tilstandsløse, og om mulig skal ressursene kunne lagres i hurtigminnet. Heretter vil REST API-ene bli omtalt som API.

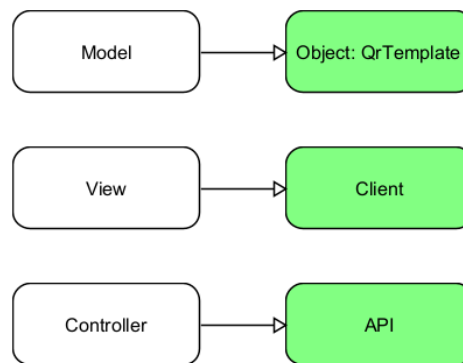


Figur 3.1. Arkitektur for generering gjennom systemets klient



Figur 3.2. Arkitektur for generering gjennom ekstern klient

Arkitekturen av systemet vil følge designmønsteret MVC (Model-View-Controller). Modellen har som ansvar å holde og håndtere data som typisk kommer fra kontrolleren. Viewet har som ansvar å presentere data for brukeren, som i dette tilfellet vil være klienten. API-ene i systemet vil fungere som kontrollere. Kontrolleren har ansvar for å ta imot data fra viewet og sende ansvaret videre til modellen. (se Figur 3.3).



Figur 3.3. Illustrasjon - anvendelse av MVC

3.4 Valg av teknologi

Prototypen skal utvikles som en webapplikasjon. Det er derfor naturlig å vurdere teknologier for frontend og backend.

3.4.1 Frontend

Brukergransnittet vil utvikles ved å benytte et grunnlag basert på HTML, CSS og enten JavaScript eller TypeScript. Her vil det diskuteres hvilket rammeverk eller bibliotek systemet skal anvende. Begge språkene som diskuteres baserer seg på teknologiene nevnt over.

TypeScript vs JavaScript

TypeScript er et objektorientert programmeringsspråk som er et superset av JavaScript (GeeksforGeeks, 2022). Det vil si at TypeScript tilbyr alt JavaScript tilbyr, samt noen flere funksjoner. En viktig egenskap i TypeScript er statisk typing (*static typing*). Dette gjør at typesjekking blir tillatt, noe som ikke eksisterer i JavaScript. Fordelen med statisk typing er at feil tildeling av typer kan bli oppdaget mye tidligere i utviklingen.

Angular

Angular er et rammeverk og en utviklingsplattform for utvikling av mobil- og webapplikasjoner, bygget på TypeScript (Angular, 2022). Rammeverket er mest brukt for å bygge komplekse applikasjoner (Dhaduk, 2022). En av fordelene ved å benytte seg av Angular er robusthet i store applikasjoner og ytelse.

React

React er et bibliotek som er basert på JavaScript for utvikling av brukergrensesnitt (React, 2022). Biblioteket er komponentbasert som med fordel gjør det lett å bygge komplekse brukergrensesnitt. React tillater bruk av JSX (JavaScript XML). Dette gjør at en komponents logikk og tilhørende HTML ikke trenger å være i separate filer. JavaScript-funksjonalitet kan altså kombineres med HTML.

Valgt teknologi

Gruppen besitter forkunnskaper med objektorientert programmeringsspråk, noe som gjør det lettere å benytte TypeScript. I motsetning til Angular, har gruppen erfaring med React. Systemet vil ikke være i størrelsesorden hvor robusthet i større applikasjoner er viktig. Dette taler for å velge React hvor det fokuseres mer på å bygge et brukergrensesnitt på en effektiv måte.

Basert på diskusjonen har gruppen valgt programmeringsspråket TypeScript i kombinasjon med React for utvikling av frontend.

3.4.2 Backend

Kapittelet tar for seg en diskusjon om hvilket programmeringsspråk og bibliotek som vil benyttes. Programmeringsspråkene som diskuteres er Java og Kotlin i kombinasjon med rammeverket Spring Boot.

Java

Java er et godt egnet språk for backend i en webapplikasjon. Gruppen har mye erfaring med Java ettersom det brukes gjennom hele studieløpet på HVL. Java er et objektorientert programmeringsspråk som har et rikt økosystem (Smith, 2017). Det vil si at det er mange verktøy

og rammeverk som er kompatibel med språket.

Kotlin

Kotlin, i likhet med Java, er et objektorientert programmeringsspråk. Kotlin er et nyere programmeringsspråk og en utvidelse av Java. Dette gjør at alt som er skrevet i Kotlin kan kjøres i alle miljøer Java kan kjøres. Kotlin, i motsetning til Java, håndterer null-verdier på en bedre måte (Ardito, 2020). Uten spesifikk håndtering av nullverdier i Java kan det forekomme nullpeker unntak (*NullPointerException*). Dette er en av fordelene ved Kotlin som gjør at systemet ikke nødvendigvis krasjer dersom det forekommer en uforventet nullverdi.

Spring Boot

Spring Boot tilbyr en enkel og rask måte for oppsett av en webapplikasjon (Spring, 2022). Spring Boot er bygget på Java, men er også kompatibel med Kotlin. Rammeverket setter opp det meste av infrastruktur slik at utviklerne kan fokusere på å bygge funksjonalitet. Noen av de viktigste fordelene er enkelheten rundt *dependency-management* og implementering av API.

Valgt teknologi

Gruppen har mye erfaring med utvikling i Java, men ønsker å utforske nye teknologier. Kotlin er en utvidelse av Java, noe som gjør en overgang til Kotlin lettere. Kotlin krever mindre *boilerplate-code*. Det vil si at det krever mindre kode for å oppnå det samme. Kotlin har i motsetning til Java bedre håndtering av nullverdier. Det var også et ønske fra veileder å benytte Kotlin.

Basert på diskusjonen har gruppen valgt programmeringsspråket Kotlin i kombinasjon med Spring Boot for utvikling av backend.

3.5 Prosjektmetodikk

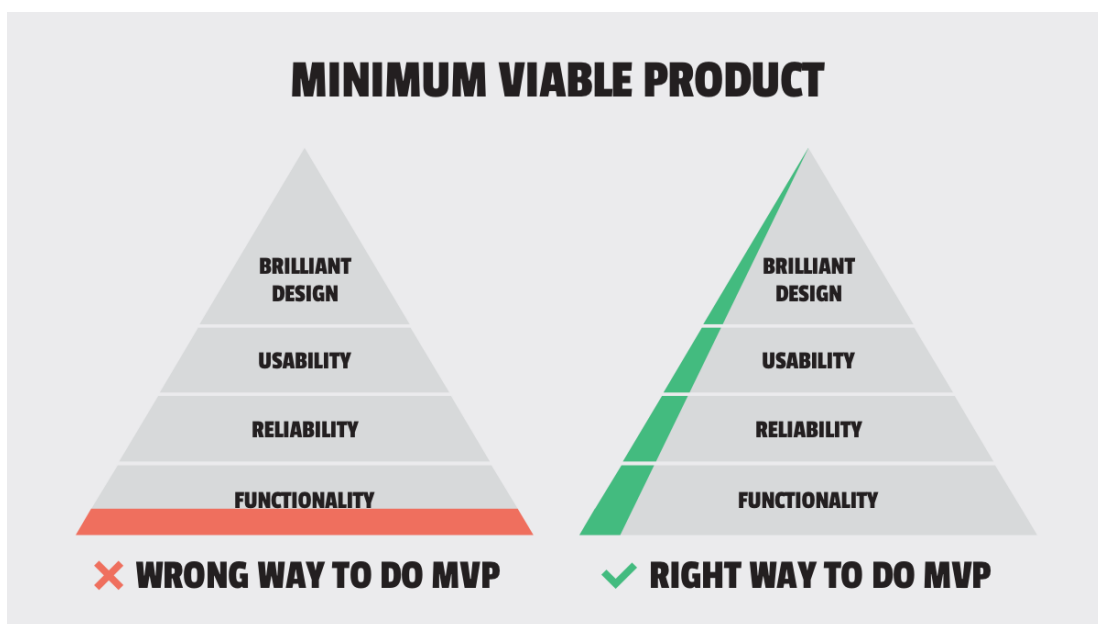
I denne delen vil ulike metodikker bli redegjort og diskutert opp mot hverandre. Til slutt vil det bli konkludert hvilken metodikk gruppen har bestemt å benytte seg av, og begrunnelse for valget.

3.5.1 Utviklingsmetodikk

Lean startup

Lean startup er en innovasjonsmetodikk som bygger på prinsipper fra Lean produksjon (*Lean manufacturing*) med mål om å redusere ressursbruk og gi økt verdi for kundene (PwC, 2018). Metodikken går innenfor kategorien “smidig utvikling”. Tre viktige prinsipper innenfor Lean startup er hypotesedrevet utvikling, kontinuerlig leveranse og brukerens interesser. Metodikken baserer seg på en iterativ utviklingsprosess hvor hver utviklingssyklus består av fasene bygg-mål-lær (*build-measure-learn*). Hensikten med iterasjonene er å skape en tilbakemeldings-syklus (*feedback-loop*).

I byggefasen utvikles det et MVP (Minimum Viable Product). Et MVP er en utgave av tjenesten eller systemet, som kun dekker de viktigste behovene for kunden. Figur 3.4 illustrerer sammensetningen til et MVP. Til høyre på figuren vises hvordan et MVP skal utformes, der en kun implementerer nødvendig funksjonalitet, brukervennlighet og design til produktet for å evaluere framgangen.



Figur 3.4. Kovalchuk, O. (2018) *What is Minimum Viable Product And How To Make It Right*. Tilgjengelig fra: <https://medium.com/anoda-mobile-development-agency/what-is-minimum-viable-product-and-how-to-make-it-right-8b885001c6f5> (Hentet: 10. mars 2022).

Målingsfasen handler om å teste MVP-et sammen med kunden. Her vil hypoteser som er satt opp på forhånd, bli evaluert. Kunder eller brukere vet ofte ikke hva de faktisk vil ha eller har behov for,

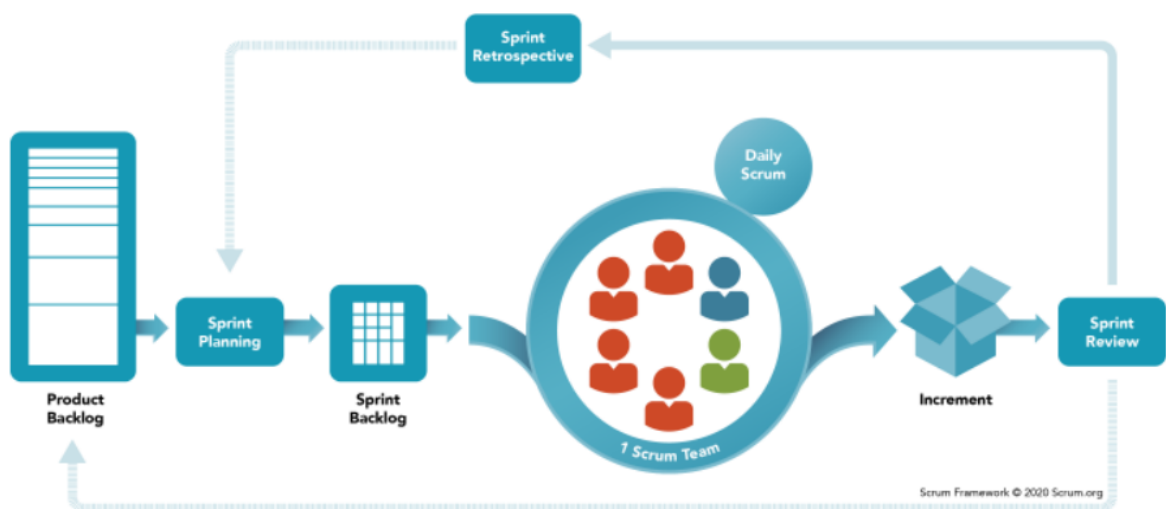
de avslører det ved deres handlinger (DeVries, 2022). Lean startup sier altså at man avdekker kundens faktiske behov gjennom testing og analyse av deres handlinger.

I læringsfasen vurderes det om utviklingen skal skje i samme retning eller om man skal justere retningen. Læringsfasen baserer seg på evalueringer av hypoteser og tilbakemeldinger fra målingsfasen. Dersom det er unødvendig å justere retning, vil MVP-et utvikles videre mot et sluttprodukt.

Scrum

Scrum er et rammeverk for produktutvikling. Metodikken går ut på at det er en scrumleder (*scrum master*) som styrer scrum-teamet i periodemessige sprinter der teamet og interessenter evaluerer resultatet og justerer kursen for neste sprint (Scrum, 2022). Til hver sprint samles krav og begrensninger både fra scrum-teamet og fra interessentene til en backlog. Scrum-tidslinjen kan i hovedsak deles inn i 4 deler: *Sprint planning*, *Daily scrum*, *Sprint review* og *Sprint retrospective*.

Figur 3.5 viser prosessen til Scrum med forklaring til de ulike stegene under.



Figur 3.5. Scrum.org (2022) *The Scrum Framework*.
Tilgjengelig fra:
<https://www.scrum.org/resources/what-is-scrum>
(Hentet: 11. mars 2022).

Sprint planning begynner sprinten ved å legge fram arbeidet som skal bli jobbet med i sprinten. Planen blir lagd i samarbeid med hele Scrum-teamet. Spørsmål som ofte besvares er:

- Hva kan bli gjort i denne sprinten?
- Hvordan skal arbeidet bli gjort?
- Hvorfor er sprinten verdifull/hva er målet?

Daily Scrum er et kort møte for utviklerne i Scrum-teamet som blir holdt hver dag gjennom sprinten. Utviklerne bruker møtene til å evaluere framgangen mot sprint-målet og evaluere framgangen mot sprint-backlog.

Sprint Review er en helhetlig vurdering av utfallet av sprinten og fastslår framtidige tilpasninger. Scrum-teamet presenterer resultatet av arbeidet for interessentene og diskuterer hva som har blitt oppnådd.

Sprint retrospective er en sluttevaluering hvor man går gjennom måter å forbedre kvalitet og effektivitet. Scrum-teamet evaluerer hvordan forrige sprinten gikk med tanke på teamets medlemmer, interaksjoner, prosesser og verktøy. Spørsmål som ofte besvares er:

- Hva gikk bra i sprinten?
- Hva kunne gått bedre?
- Hva skal gjøres for å forbedre neste sprint?

Scrum vs Lean startup

Scrum er et rammeverk som egner seg for større teams. For at Scrum kan gjennomføres etter sin hensikt krever det et utviklingsteam, en produkteier og en Scrumleder. Dette ligger ikke til rette i prosjektet, ettersom gruppen fungerer som et selvstendig produktteam. Det vil være utfordrende å gjennomføre *Daily Scrum* fordi gruppemedlemmene har valgfag som går parallelt med bachelorprosjektet, og ulike timeplaner. Lean startup, i likhet med Scrum, er en iterasjonsbasert utviklingsmetodikk. For å forsikre at riktig produkt utvikles blir et MVP utformet og testet for å evaluere om de riktige behovene dekkes. Metodikken baserer seg på hypotesedrevet utvikling som er en fordel når gruppen har manglende forkunnskaper i domenet problemet befinner seg.

Som følge av drøftingen ovenfor har gruppen bestemt å anvende utviklingsmetodikken Lean startup.

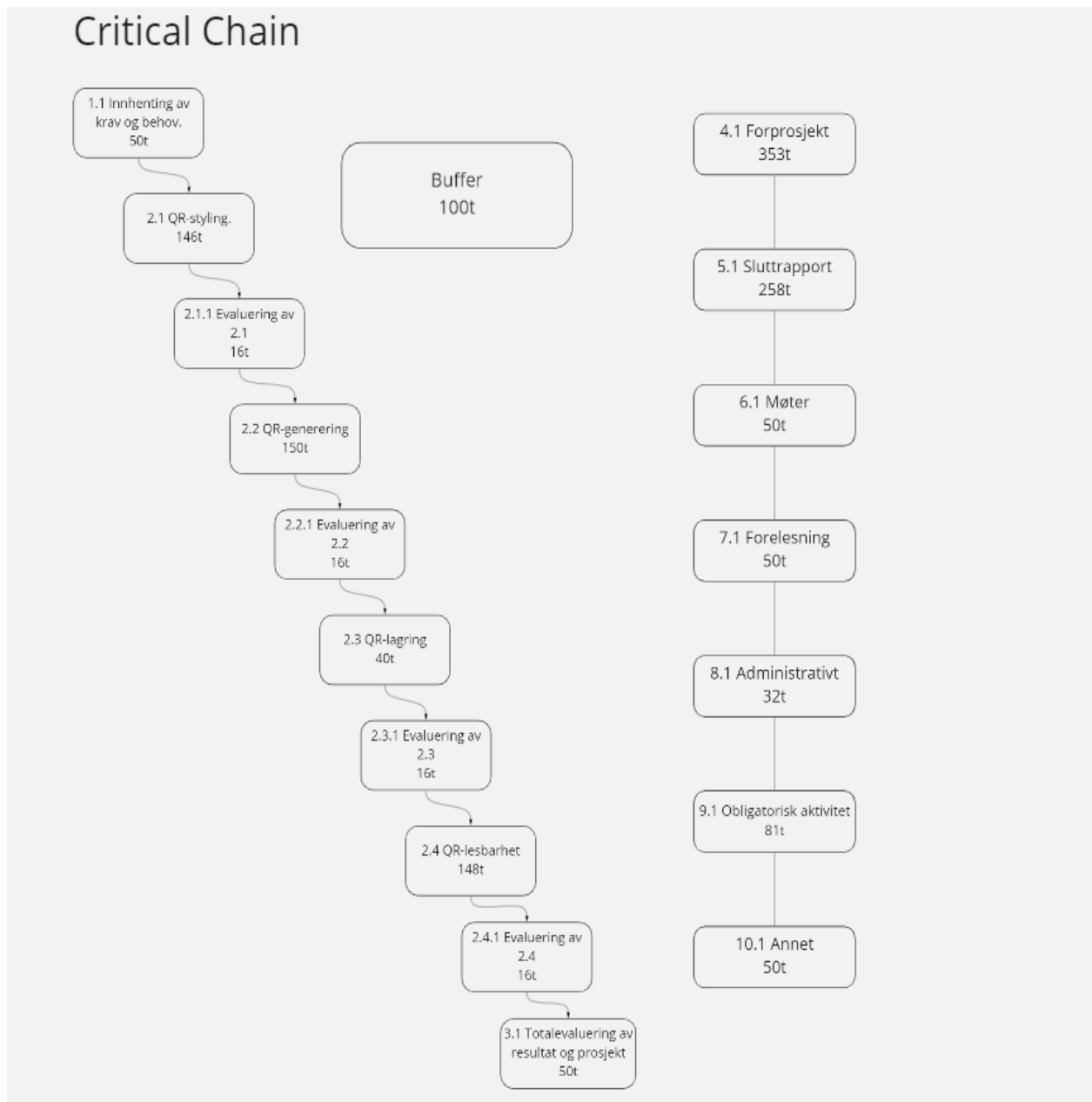
3.5.2 Prosjektplan

Gruppen har valgt å benytte Critical Chain for prosjektplanlegging. Critical Chain følger i større grad en logisk sti, i motsetning til Gantt-diagram. Ettersom prosjektet vil bli utført iterasjonsbasert og smidig, mener gruppen det er hensiktsmessig å ikke sette en spesifikk start- og sluttdato for hver prosess. Hver prosess vil istedenfor få tildelt et forventet antall timer. Det kan være vanskelig å fordele riktig antall timer til en prosess. Av den grunn inneholder Critical Chain en buffer hvor det kan trekkes og legges til timer hvis en prosess varer lenger eller kortere enn anslått.

I gruppens Critical Chain (se Figur 3.6) er det laget underpunkter på prosess 2.1 til 2.4. Disse er evalueringpunkter der forrige prosess skal evalueres. Dersom forrige prosess ikke er tilfredsstillende går gruppen tilbake til prosessen for å utbedre. Dette er lagt inn slik at det blir enklere å evaluere produktets hoveddeler underveis.

Punkt "9.0 Obligatorisk aktivitet" er en samlebetegnelse på alle obligatoriske aktiviteter til bachelorfaget, utenom rapportskriving. Dette vil for eksempel være analyse av tidligere bacheloroppgaver. Punktene "4.0 Forprosjekt" og "5.0 Sluttrapport" inkluderer også obligatorisk aktivitet rundt visjonsdokument, prosjekthåndbok og kravdokument.

Critical Chain er beskrevet mer detaljert i støttedokumentet "Prosjekthåndbok" kapittel 2.



Figur 3.6. Critical Chain diagram - gruppens prosjektplan

3.5.3 Risikovurdering

Risikoanalysen tar for seg ulike hendelser som kan inntreffe i løpet av perioden til bachelorprosjektet. Hver hendelse blir gitt en sannsynlighet og konsekvens. Sannsynligheten sier noe om sjansen for inntreffelse. Konsekvensen sier noe om hvor stor påvirkning hendelsen har. Risikoprodukt er et produkt av sannsynlighet og konsekvens. Produktet gir en indikasjon på hvor

alvorlig en hendelse er. I tillegg er det lagt inn tiltak som fungerer forebyggende mot en hendelse eller som en respons dersom hendelsen inntreffer.

Gruppen har identifisert hendelsene til prosjektet som er ansett som viktige. Tabellen under viser de fire hendelsene med høyest risikoprodukt. Se [vedlegg 9.1](#) for resterende hendelser. Sannsynlighet og konsekvens har en verdi mellom 1-5, dermed har risikoproduktet en maksimal verdi på 25.

	Hendelse /Risiko	Årsak	Sannsynlighet	Konsekvens	Risiko-produkt	Tiltak
1	Applikasjonen gir lav nytteverdi.	Ineffektiv og lite brukervennlig produkt.	Lav(2)	Høy(4)	8	Gjennomføre brukertesting.
2	Feilvurdering av oppgavens omfang.	Klarer ikke oppfylle mål og funksjonelle krav.	Middels(3)	Høy(4)	12	Går nøye gjennom oppgaven og planlegger godt.
3	For stor arbeidsmengde.	Arbeidsmengden er for stor ut ifra forventningene.	Lav(2)	Høy(4)	8	Lager gode planer og holder disse.
4	Tilegner ikke nok kunnskap om hva som påvirker lesbarheten til en QR-kode.	Har ikke klart å hente nok informasjon eller at området ikke er tilstrekkelig forsket på.	Middels(3)	Middels(3)	9	Alle på gruppen tilegner seg tilstrekkelig kunnskap om QR-koder. Utføre tilstrekkelig forskning.

Tabell 3.1. Hendelsene med høyest risikoprodukt

3.6 Evalueringsplan

Gruppen har som mål å få evaluert alt som blir utviklet under prosjektet og prosjektet i sin helhet. Det vil bli brukt forskjellige metoder for å evaluere de ulike komponentene ved bachelorprosjektet.

Underveis i utviklingen benyttes arbeidsmetodikken Lean startup. For hver iterasjon skal funksjonalitet bli testet og evaluert i form av brukertesting og intervjuer. Basert på tilbakemeldingene kan gruppen gjøre nødvendige endringer.

Etter at alle iterasjonene har blitt evaluert er det viktig å evaluere resultatet. Prototypen skal bli evaluert på forskjellige måter for å få tilbakemelding fra flere ulike perspektiv. En måte vil være å få tilbakemelding fra oppdragsgiver. Basert på tilbakemeldingene får gruppen vite om oppdragsgiver er tilfreds med resultatet. Den andre måten er brukertesting av systemet. Brukertesting skal skje på medstudenter, studenter utenfor studiet og familie. Ved å teste

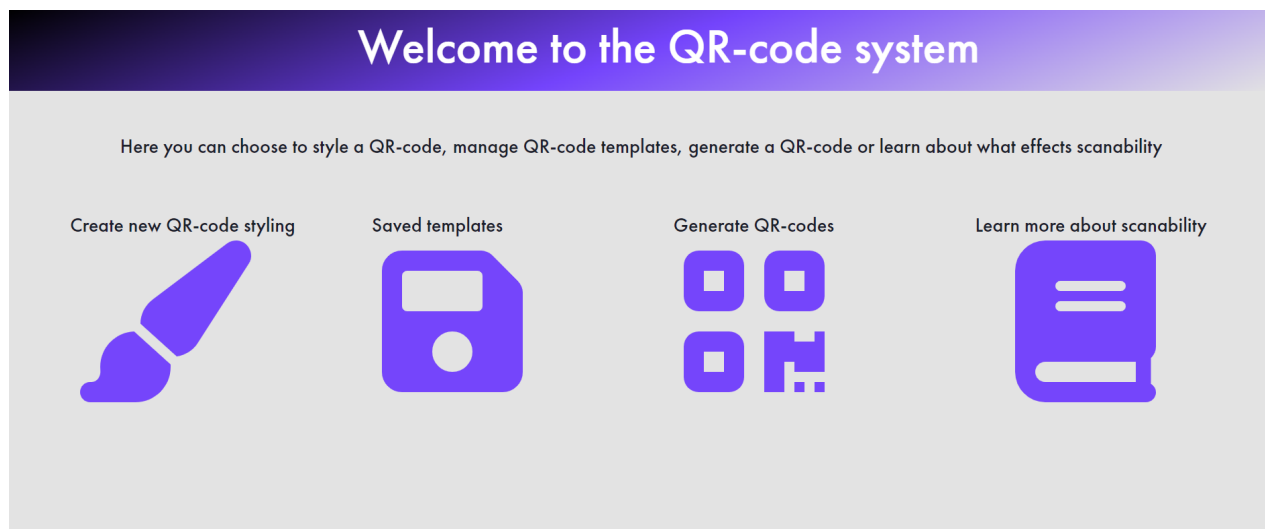
prototypen på brukere med ulike nivåer av teknisk kunnskap, vil gruppen få ulike perspektiv på brukeropplevelsen. Til slutt skal resultatet bli evaluert opp mot de funksjonelle kravene. De funksjonelle kravene er utformet utfra behovene og ønskene fra oppdragsgiver og interessenter. Hvis de fleste funksjonelle kravene er oppfylt, gir det en indikasjon på at produktet er utviklet i riktig retning.

Under evalueringen av hele prosjektet er det viktig å hente tilbakemeldinger fra gruppe-medlemmene. Måten gruppen skal gjøre det på er å få samtlige medlemmer til å individuelt reflektere over prosjektets gjennomførelse og gi en intern evaluering. Her er det viktig å se prosjektet som en helhet og å dele det opp i ulike deler. Det vil være hensiktsmessig å påpeke hva som fungerte bra og hva som fungerte mindre bra, samt hva man ville gjort annerledes dersom det skulle bli gjort på nytt. I tillegg kan det være nyttig å spørre om veileder sin mening rundt prosjektet.

4 LØSNING

I dette kapitlet blir prototypen beskrevet i detalj ved å gjøre rede for brukergrensesnittet og den tekniske løsningen. Systemet har tre hovedfunksjoner, hver av disse blir presentert under.

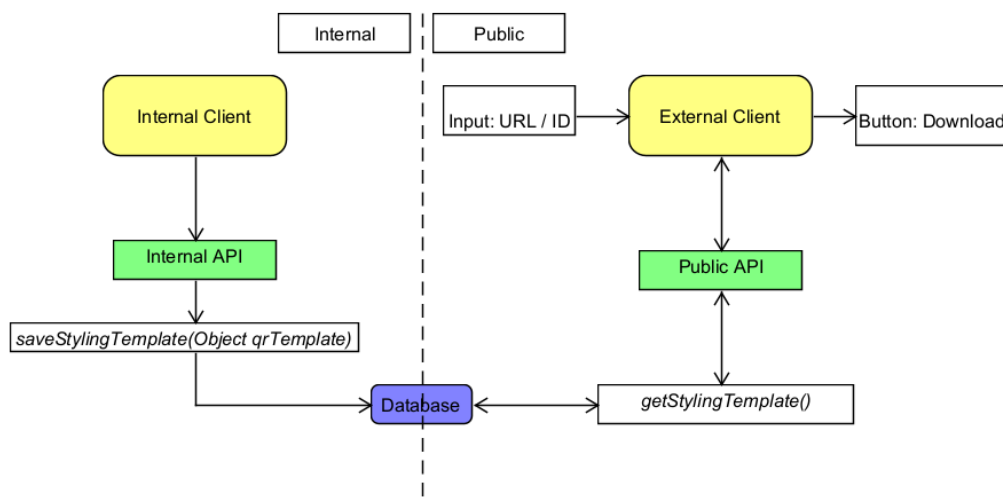
Brukerens første møte med prototypen er hjemmesiden hvor man kan navigere seg til ønsket side (se Figur 4.1). Ved å trykke på ett av ikonene blir brukeren sendt videre til en ny side.



Figur 4.1. Hjemmesiden til prototypen

4.1 Arkitektur

Den arkitektoniske løsningen fungerer på samme måte som beskrevet i [Kapittel 3.3 Valgt løsning](#), for systemets klient. Grunnet konsekvenser ved valg av løsning for styling av QR-koder er arkitekturen for generering gjennom eksterne klienter endret (se Figur 4.2).



Figur 4.2. Arkitektonisk løsning for generering av QR-koder gjennom ekstern klient

4.2 Styling av QR-koder

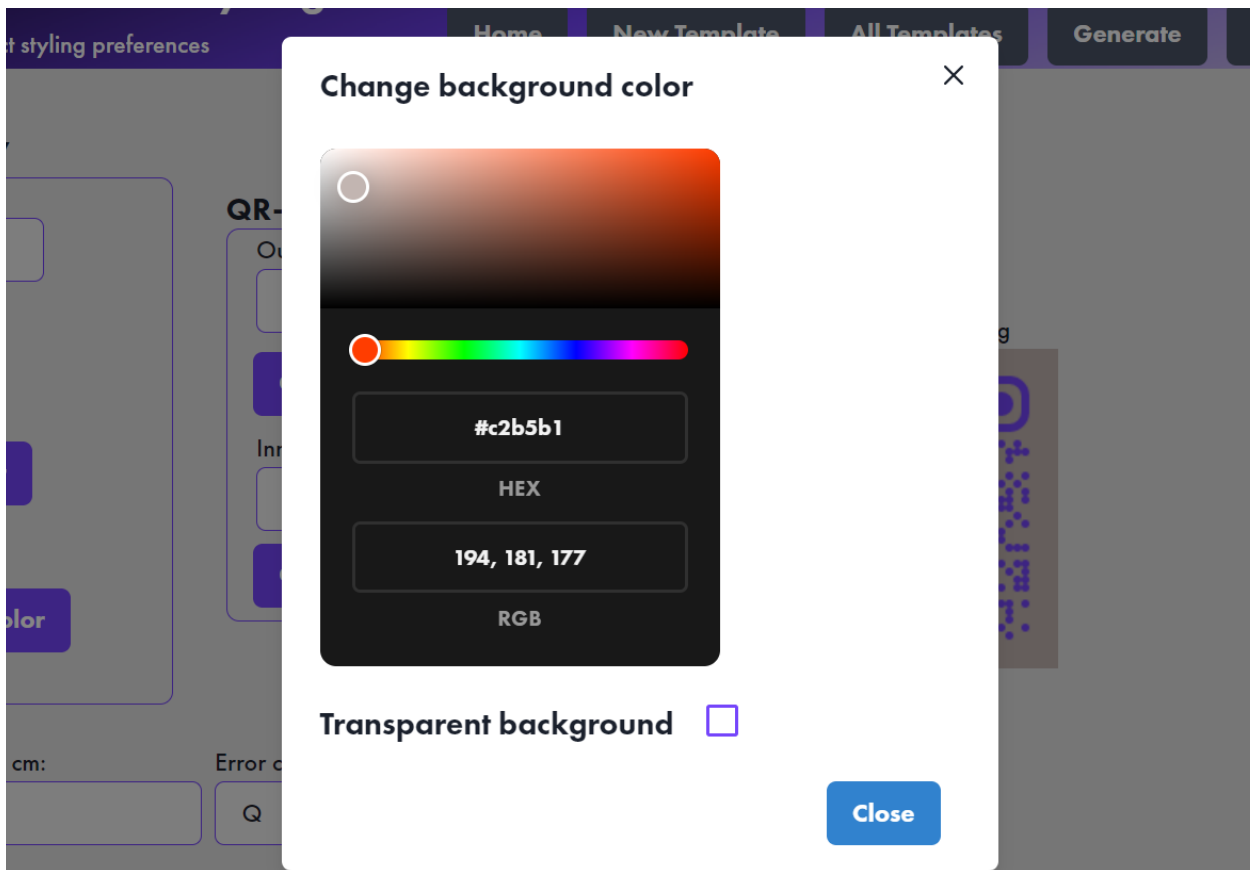
Brukerne er fornøyde med funksjonaliteten dagens løsning tilbyr. Favorit ønsket derfor at løsningen skulle ha tilnærmet like muligheter for styling. Løsningen benytter seg av biblioteket qr-code-styling på klient-siden. Biblioteket tilbyr et utvalg av funksjoner for styling av QR-koder.

Brukergransnittet består av en navigasjonslinje, en HTML-form med input-komponenter og en visning av valgt styling. Brukeren har muligheten til å redigere mønsteret på “kroppen” til QR-koden, hjørne-dotten og hjørne-rammen, samt å endre fargen på hver av disse delene (se Figur 4.3). Brukeren kan også sjekke om kontrasten på QR-koden er tilfredsstillende ved å trykke på knappen “Check contrast”.

Figur 4.3. Brukergrensesnittet for styling av QR-koder

I tillegg er Favrit sin logo lagt i midten av QR-koden med mulighet for å endre på størrelsen. Dersom det ønskes å generere en QR-kode uten logo kan brukeren sette "image size" til 0. Det er mulig å navngi malen slik at det kan tydeliggjøres hvilken mal som skal brukes i ulike tilfeller. Brukeren får også muligheten til å kunne endre margin på QR-koden og nivået av feilkorrigering (se Figur 4.3).

Ved endring av for- og bakgrunnsfarge vil brukeren få opp en modalboks (se Figur 4.4). Denne modalboksen inneholder en fargevelger hvor brukeren kan endre farge ved en hjelp av HSV, HEX eller RGB input. Brukeren kan huke av en checkbox for å velge gjennomsiktig bakgrunnsfarge. Fargevelgeren kommer fra et bibliotek som heter react-color-palette.



Figur 4.4. Fargevelger - Modal

Brukeren kan lagre stilingen som en mal ved å trykke på knappen "Save template". Ved lagring sender klienten en forespørsel til API-endepunktet `/styling/v1/save-template` (se Figur 4.5). Alle attributtene tilhørende malen blir lagret i databasen.

```
@PostMapping(value = ["/save-template"])
fun saveTemplate(@RequestBody qrTemplateDto: QrTemplateDto): QrTemplateDto {

    stylingService.addQrTemplate(qrTemplateDto)
    return qrTemplateDto

}
```

Figur 4.5. API-endepunkt for lagring av mal

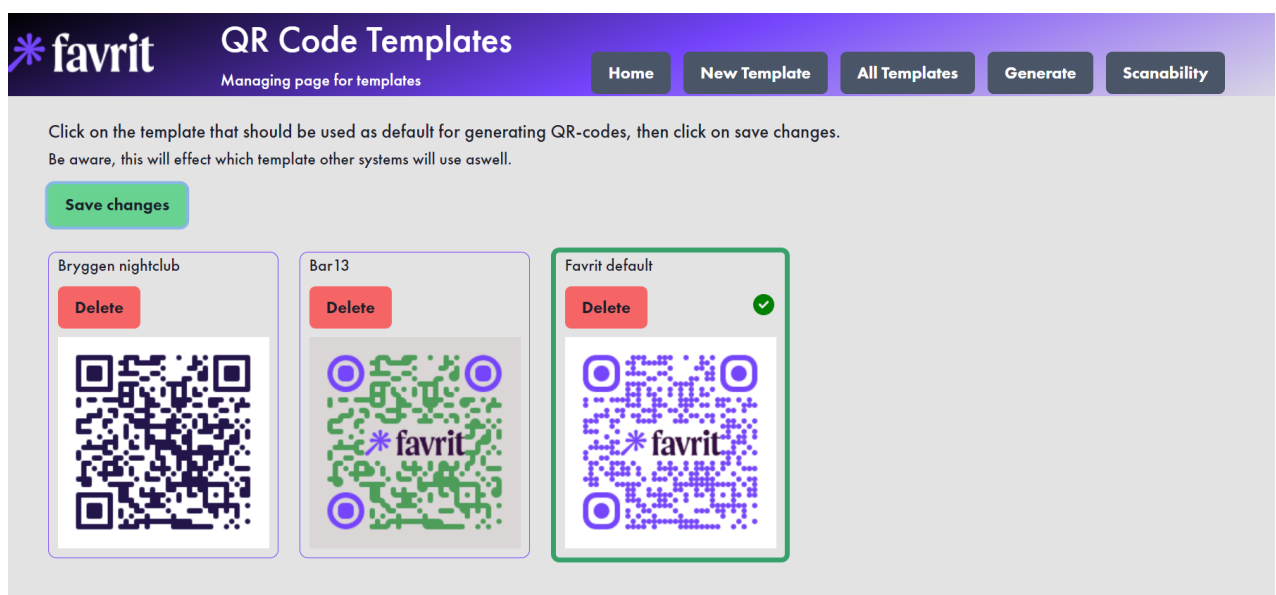
Endepunktet vil kalle på metoden `addQrTemplate()` med objektet `QrTemplateDto` som parameter. Service-laget vil ta ansvar for å mappe objektet til ønskelige datastruktur i databasen før den lagres (se Figur 4.6).

```
fun addQrTemplate(qrTemplateDto: QrTemplateDto) {  
    repository.save(Utils.mapDtoToQrTemplate(qrTemplateDto))  
}
```

Figur 4.6. Service-laget for lagring av mal

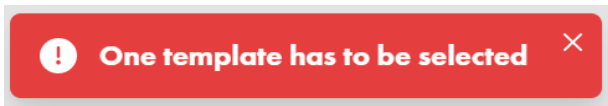
Endepunktet vil returnere HTTP statuskoden `200 OK` og det lagrede objektet, om forespørselen er vellykket.

Brukeren vil se en liste over alle lagrede maler ved å navigere seg til “All templates” fra startsidene eller navigasjonslinjen. Hensikten til denne siden er å kunne slette lagrede maler og kunne endre hvilken mal som skal brukes som standard for hele systemet (se Figur 4.7). Malen som velges her vil få følge for hvilken mal som brukes av andre klienter som benytter API-et.

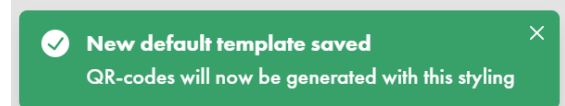


Figur 4.7. Brukergrensesnitt for håndtering av QR-kode maler

Systemet er avhengig av at det alltid er valgt en mal. Derfor er det blitt implementert tilbakemeldinger til brukeren dersom det ikke er valgt en mal og når endringer er blitt lagret (se Figur 4.8 og Figur 4.9). Tilbakemeldingene blir vist som en dialogboks (*toast*).



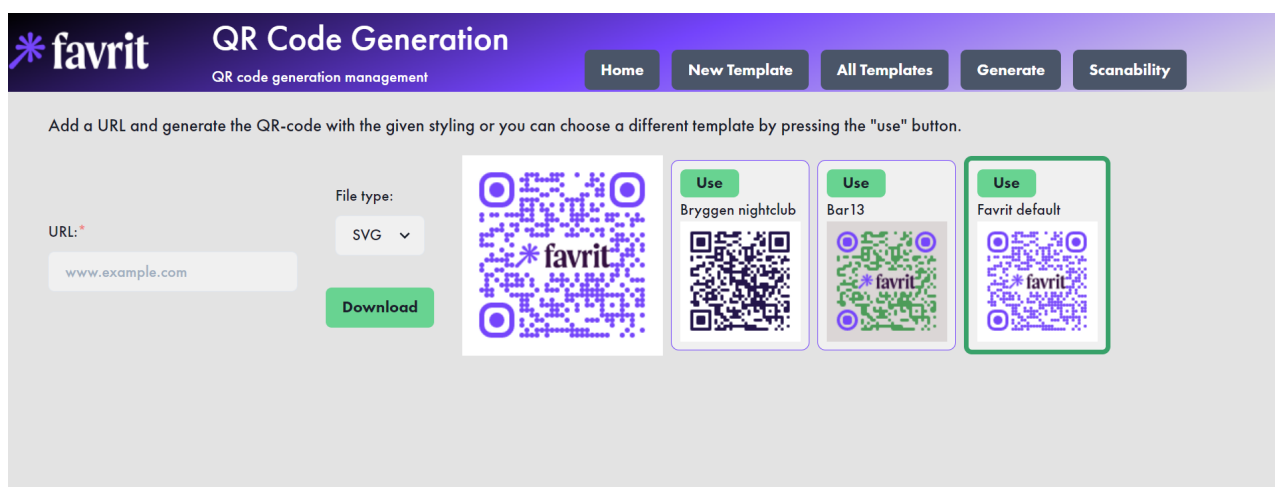
Figur 4.8. Tilbakemelding til bruker dersom ingen mal er valgt



Figur 4.9. Tilbakemelding til bruker dersom endringer er utført

4.3 Generering av QR-koder

Generering av QR-koder er lagt til rette for på to ulike måter, gjennom brukergrensesnittet og ved at eksterne klienter kan koble seg på API-et. Ved generering gjennom brukergrensesnittet må brukeren legge inn en URL (se Figur 4.10). QR-koden vil benytte malen som er valgt av brukerne. Klienten sender en GET-forespørsel til endepunktet `/templates/v1/selected-template` og får returnert malen som er valgt. Dersom det skulle være behov for å generere QR-koder uten den forhåndsbestemte malen, vil brukeren få muligheten til å bytte til en av de andre lagrede malene. Brukerne kan velge hvilket format QR-koden lastes ned som. Alternativene er: SVG, PDF, EPS, PNG, JPEG og WEBP. Brukeren vil kunne laste ned QR-koden lokalt og distribuere koden videre. QR-koden vil ikke bli lagret i systemet, men i stedet regenereres ved behov.



Figur 4.10. Brukergrensesnittet for å generere QR-kode med valgt mal

Eksterne klienter må ta i bruk biblioteket `qr-code-styling` og opprette et tilfellet av JavaScript klassen `QRCodeStyling` dersom de ønsker å benytte seg av API-et. Deretter må klienten hente stilingen til valgt mal fra API-et og sette inn attributtene den mottar. Klienten må også implementere funksjonen for nedlastning som kommer fra biblioteket.

4.4 Innsikt i lesbarhet

Det viktigste behovet som ikke dekkes av dagens løsning er tilbakemelding på lesbarhet. Etter en QR-kode er stilet vet ikke markedsføringsavdelingen om det nye utseende har for dårlig lesbarhet. Det er derfor ønskelig at systemet gir innsikt i hva som påvirker lesbarheten og en tilbakemelding på lesbarheten til gitt stiling. Markedsføringsavdelingen ønsket å få innsikt i hvordan kontrast og logo påvirker lesbarheten, samt hvordan ulike overflater og omgivelser påvirker. Av den grunn har gruppen utført et eksperiment.

I følge Jason Coleman er kontrast mellom forgrunns- og bakgrunnsfarge viktig for å sikre god lesbarhet (Coleman, 2011, s. 18). QR-koder med lav kontrast vil derfor være vanskelige å skanne. Eksperimentet tar utgangspunkt i denne teorien, samt at kontrast er en av egenskapene som er målbar. Hensikten med eksperimentet er å produsere data for deretter å analysere hva som påvirker lesbarheten til en QR-kode. Blir det produsert nok data utvikles en algoritme for å gi brukeren tilbakemelding på lesbarheten til en stilet mal. Det er formålstjenlig å finne: hvor grenseverdien for kontrast ligger, hvilke farger fungerer bra eller dårlig, om tydeligere *positioning detection markers* påvirker lesbarheten, hvordan ulike overflater og omgivelser kan påvirke lesbarheten, og hvilket operativsystem som tilbyr best skanning. Basert på hva som undersøkes, er det utviklet en hypotese, "styling av en QR-kode kan føre til at QR-koden blir ulesbar".

4.4.1 Utstyr

Det er viktig at utstyret som er brukt under eksperimentet kan representere utstyret til en vanlig gjest. Utstyret som er brukt til å skanne QR-kodene er iPhone 12 og Samsung Galaxy S21. Begge telefonene ble lansert for omkring halvannet år siden (Wikipedia, 2022a; Wikipedia, 2022b). Den digitale skjermen som er brukt til skanning er en Lenovo IdeaPad L340. QR-kodene er skrevet ut på Navigator Universal A4-papir (210x297 mm), uten hull, 80g/m². Det blir brukt et plastark som legges over A4-papiret for å etterligne laminat.

Alle QR-kodene har like egenskaper bortsett fra farge og logo. De inneholder en identisk streng, som kan representere en forventet meny-URL. QR-kodene består av originalt mønster med en *error correction* av nivå Q (25%). Alle er generert som filtype PNG med en fysisk størrelsen på 5x5 cm. QR-kodene som inneholder logo, bruker Favrit sin logo med en logo-størrelse på 0,5 i gruppens prototype.

4.4.2 Forsøksoppsett

Eksperimentet ble gjennomført ved å manuelt style, skanne og dokumentere resultatene. Alle QR-kodene har blitt skannet gjennom kamera-applikasjonene til telefonene. De har blitt skannet på tre forskjellige overflater i to ulike omgivelser. Overflatene som har blitt brukt er digital skjerm, papir og laminert papir. Omgivelsene har vært normal belysning og dempet belysning. Skanning av QR-koder i dempet belysning har blitt utført i et rom uten vinduer med så lav belysning som mulig. Dette ble gjort med hensikt om å skape omgivelser som kan tilsvare et serveringssted på kveldstid. Med digital skjerm har omgivelsene ikke blitt tatt hensyn til. Dette resulterte i at hver QR-kode har blitt skannet ti ganger.

Eksperimentet består av flere iterasjoner. Hver iterasjon blir utført på samme måte. Først blir hver QR-kode stylet, skannet på digital skjerm og limt inn i et dokument. Videre blir dokumentet skrevet ut på papir, og QR-kodene blir skannet på nytt med og uten laminat. Til slutt blir de skannet i et rom med dempet belysning. Resultatene ble dokumentert fortløpende. Resultatene som blir presentert videre i kapittelet er laveste grenseverdi som ga vellykket lesing med dårligste telefon i lyst rom. Det er disse resultatene som er av størst interesse.

Eksperimentet har valgt å benytte seg av RGB fargemodellen. I forkant av eksperimentets utførelse var det nødvendig å tilegne seg kunnskap om hvordan RGB modellen fungerer. Den består av tre komponenter der hver komponent er et tall fra 0-255. En RGB farge kan se slik ut, "116, 68, 252". Tallene sier noe om hvor mye rødt, grønt og blått det er i fargen. De tre tallene viser respektive nivået av rødt, grønt og blått. Den høyeste verdien en komponent kan ha er 255. Dersom alle tallene er 255, blir fargen hvit. Dersom alle tallene er 0, blir fargen svart.

I følge Lode Vandevenne sin artikkel om farger, vil høyere verdi på RGB komponentene gi lysere farge (Vandevenne, 2004). Eksperimentet baserer seg derfor på å finne grenseverdien for når en farge blir for lys med hensyn på kontrast. Med grenseverdi menes laveste differansen mellom

summert RGB forgrunnsfarge og summert RGB bakgrunnsfarge som ga vellykket skanning. For eksempel dersom bakgrunnsfargen er hvit (255, 255, 255) og forgrunnsfargen er lyserosa (255, 210, 255), blir grenseverdien $(255+255+255) - (255+210+255) = 45$. Ved funn av en grenseverdi, ble grenseverdi-fargen og fargen rett under grenseverdien skannet to ganger ekstra for å forsikre at grenseverdien er riktig.

4.4.3 Ensfarget QR-kode

Eksperimentets første iterasjon omhandlet å finne grenseverdien for kontrast til ulike farger på ensfargede QR-koder. Ved testing av grenseverdi ble alle tallene økt til 255, for deretter å senke ett av tallene til QR-koden ble lesbar (se Figur 4.11). Videre ble QR-koder med høyere og lavere verdi generert for å se om lesbarheten forandret seg på papir. Prosessen ble gjentatt for hver RGB komponent.



Figur 3. 255, 210, 255



Figur 2. 255, 190, 255



Figur 1. 255, 170, 255

Figur 4.11. Tre QR-koder med tilhørende RGB farge under QR-koden

Videre ble QR-koder med andre fargekombinasjoner og omtrent lik grenseverdi generert. Tabell 4.1 viser grenseverdien til de ulike fargene på papir med normal belysning. Se [vedlegg 9.6](#) for detaljerte resultater.

	Turkis	Rosa	Gul	Grønn/ turkis	Oransje	Grønn	Grå	Lilla
Grenseverdi	65	45	255	100	100+	100	100+	80

Tabell 4.1. Grenseverdi for ulike farger

4.4.4 QR-koder med tydeligere *positioning detection markers*

Det er foretrukket å finne ut om tydeligere *positioning detection markers* kan påvirke lesbarheten. Måten dette ble undersøkt på var ved å generere to sett med QR-koder. Begge settene hadde *positioning detection markers* med 50 lavere RGB verdi enn forgrunnsfargen (se Figur 4.12). Det ene settet inneholder QR-koder der forgrunnsfargen tidligere ikke var lesbar, og det andre settet der forgrunnsfargen tidligere var lesbar.



Figur 3. Forgrunn: 230, 247, 228.
Hjørner: 210, 237, 208



Figur 2. Forgrunn: 220, 242, 223.
Hjørner: 200, 232, 203



Figur 1. Forgrunn: 210, 237, 218.
Hjørner: 190, 227, 198

Figur 4.12. Tre QR-koder med tilhørende RGB farger under QR-koden.

Resultatene viser grenseverdien til forgrunnsfargen til de ulike fargene. Tabell 4.2 viser grenseverdien på papir med normal belysning. Gul ble ikke testet fordi den hadde tydelige problemer med lesbarheten, og det var ikke mulig å lage sterkere gul-kontrast. Se [vedlegg 9.6](#) for detaljerte resultater.

	Turkis	Rosa	Gul	Grønn/ turkis	Oransje	Grønn	Grå	Lilla
Grenseverdi	55	45	x	60	100+	80	100+	60

Tabell 4.2. Grenseverdi for ulike farger med 50 lavere summert RGB på *position detection markers*

4.4.5 QR-koder med logo

Favrit ønsker å bruke logoen sin i QR-kodene, men er usikre på om det påvirker lesbarheten. Det har derfor blitt generert et nytt sett med QR-koder som inneholder logo. Tabell 4.3 viser grenseverdien på papir med normal belysning. Se [vedlegg 9.6](#) for detaljerte resultater.

	Turkis	Rosa	Gul	Grønn	Oransje	Grå	Blå
Grenseverdi	85	75	Gikk ikke	140	120	120	80

Tabell 4.3. Grenseverdi for ulike farger med logo

4.4.6 Konklusjon

Etter 1400 skanninger er det samlet nok data til å konkludere. Det har blitt funnet grenseverdier for forskjellige farger med ulike betingelser til QR-kodene. Eksperimentet har i tillegg funnet ut hvilke overflater som fungerer bedre enn andre og om omgivelsene påvirker lesbarheten. Resultatene til skanning av gul QR-kode har blitt utelatt i konklusjonen. Prototypen informerer brukeren om at gul QR-kode har tydelig dårligere lesbarhet.

Grenseverdien til forgrunnsfargen varierer utfra ulike betingelser. Ser man resultatene i sammenheng, ligger grenseverdien til ensfarget QR-kode mellom 45-125. Grenseverdien til QR-koder med høyere kontrast på *positioning detection markers* ligger omkring 20 lavere enn hos ensfarget QR-kode. Det vil si at lesbarheten øker. QR-koder med logo viste seg generelt å ha dårligere lesbarhet. Grenseverdien her lå på 75-140.

Ulike overflater påvirker lesbarheten på forskjellige måter. Resultatene har vist at QR-koder har dårligere lesbarhet på digital skjerm enn på papir. Trenden tilsier at grenseverdien på digital skjerm ligger 30-70 høyere enn på papir. QR-koder på laminerte papir har tilsvarende lik lesbarhet som vanlig papir. Det er kun enkelttilfeller hvor lesbarheten på laminert papir har vært dårligere. Allikevel kan QR-koder på laminert papir være vanskeligere å skanne for brukeren. Det skyldes refleksjon i laminatet som gjør at QR-koden blir dekket over. Med en gang lysrefleksjonen ikke dekker over QR-koden, har den tilsvarende lesbarhet som på papir.

Omgivelsen QR-koden er tilstede i, kan påvirke lesbarheten. Trenden indikerer at QR-koder i omgivelser med normal belysning har bedre lesbarhet enn i dempet belysning. Lesbarheten har

ikke blitt påvirket av den dempede belysningen i omtrent 25% av tilfellene. Dette skyldes at kamera-applikasjonene har automatisk lyskorrigering slik at QR-koden har blitt lysere.

Gjennom hele eksperimentet har en av telefonene hatt flere vellykkede skanninger enn den andre. Samsung Galaxy S21 har klart å skanne alle QR-kodene som iPhone 12 har klart, men i tillegg har den skannet QR-koder som har hatt lavere grenseverdi på alle overflatene.

Grunnet resultatene i konklusjonen, kan gruppen fastslå at hypotesen stemmer.

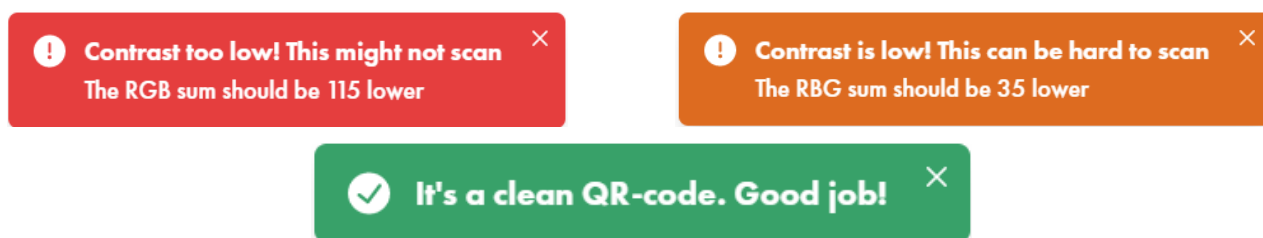
4.4.7 Algoritme

Basert på resultatene er det utviklet en algoritme som gir tilbakemelding på lesbarheten til gitt styling. Algoritmen er utviklet med hensyn på de dårligste vellykkede skanningene, slik at QR-kodene skal kunne skannes med flest mulig telefoner. Algoritmen sammenligner forgrunnsfarge og bakgrunnsfarge, og sjekker om kontrasten er høy nok. I tillegg tar den betraktning til kontrasten mellom *positioning detection markers* og forgrunnsfargen. Det er blitt lagt inn en sikkerhetsmargin, slik at algoritmen tar hensyn til mobiler med dårligere skanning.

Algoritmen baserer seg på RGB komponentene. Den summerer opp alle tallene til forgrunnsfargen og bakgrunnsfargen, for deretter å sjekke differansen. Gitt at *positioning detection markers* har stor nok kontrast til forgrunnsfargen, vil algoritmen tillate lavere kontrast mellom forgrunnsfargen og bakgrunnsfargen.

Ut fra differansen gir algoritmen tilbakemelding på om kontrasten er tilfredsstillende. Brukeren får tilbakemelding på lesbarheten ved å trykke på knappen "Check contrast" på stylingsiden. Den gir tre ulike tilbakemeldinger: defekt, advarsel og vellykket. Tilbakemeldingene blir vist som en dialogboks, og det står hvor langt unna fargen er for å bli tilfredsstilt (se Figur 4.13).

Alle QR-kodene som har blitt generert og godkjent av "Check contrast" knappen har hatt fremragende lesbarhet. QR-kodene som har fått advarsel-melding har klart å bli skannet. Dette var som forventet, siden algoritmen tar høyde for dårligere mobiler og omgivelser.



Figur 4.13. Tre ulike tilbakemeldinger basert på kontrastnivåene

4.4.8 Informasjonsside

Basert på resultatene fra eksperimentet og anskaffet kunnskap om QR-koder ble det utviklet et informasjonsskriv. Skrivet ble implementert som en informasjonsside i klienten. Informasjonssiden har som hensikt å skape et godt grunnlag for hva som er viktig å være bevisst på når en stiler en QR-kode. Informasjonssiden fungerer også som en oppsummering av kunnskapen gruppen har tilegnet seg når det kommer til hva som påvirker lesbarheten.

5 EVALUERING OG RESULTAT

Dette kapittelet tar for seg hvordan resultatene evalueres, hvilke metoder og framgangsmåter som er brukt og en konklusjon av evalueringen.

5.1 Evalueringsmetode

Her vil ulike evalueringsmetoder som er benyttet under utviklingen av systemet bli presentert. Metoder for å evaluere om det er riktig produkt som bygges og metoder for å evaluere om produktet bygges riktig.

5.1.1 Evaluering av produktutvikling

Hvordan kan det sikres at det bygges et produkt som faktisk løser problemet som skal løses? Arbeidsmetodikken Lean startup er blitt anvendt under produktets utvikling. Metodikken har prinsipper for hvordan man evaluerer produktets utvikling, nettopp for å sørge for at det er riktig produkt som bygges.

Framgangsmåten for evaluering går ut på å utforme antagelser og hypoteser for løsninger på et problem. I neste fase bygges det et MVP hvor målet er å implementere funksjonalitet som skal løse problemet. Videre vil potensielle brukere bli kalt inn til et møte hvor MVP-et testes sammen med brukeren. Basert på tilbakemeldinger og observasjoner under brukertesten vil antagelsene og hypotesene bli bekreftet eller avkreftet. Til slutt vil det bli tatt en avgjørelse om riktig produkt bygges eller om det er behov for justeringer. Dette vil foregå over flere iterasjoner. Denne metoden sørger for at man fortløpende evaluerer deler av systemet for å ende opp med et sluttprodukt som løser problemet.

Første iterasjon omhandlet funksjonalitet for styling av QR-koder. Etter MVP-et for styling av QR-koder var ferdigstilt, ble markedsføringsavdelingen kalt inn til brukertesting. Brukerne etterlyste funksjonalitet som endring av bakgrunnsfarge, gjennomsiktig bakgrunn, opplastning av logo og at systemet ga tilbakemelding når malen ble lagret. Basert på tilbakemeldingene var det ikke behov for å justere retning. Savnet funksjonalitet ble lagt til, bortsett fra opplastning av logo. Det viste seg å være mer tidkrevende og utfordrende enn forventet. Dette ble nedprioritert fordi logoen blir sjeldent endret. I stedet har gruppen hardkodet en logo fra Favrit.

Andre iterasjon tok for seg generering av QR-koder. Etter MVP-et for generering var utarbeidet ble potensielle brukere fra supportavdelingen kalt inn til brukertesting. Etter brukertesten hadde brukerne innspill til forbedringer. Det første innspillet var muligheten til å legge inn en fullstendig URL til QR-koden som skulle genereres og ikke bare en kunde-ID. Det kan være behov for å generere QR-koder som ikke kun peker til kundenes menyer. Et annet innspill handlet om enkelttilfeller hvor det vil være nødvendig å generere QR-koder med en annen styling enn standarden. Det var ønskelig å kunne velge blant andre maler under generering. Som følge av dette var det også ønskelig at malene kunne navngis. Dette kan tydeliggjøre hvilken mal som brukes for generering av QR-koder, for kunder som ønsker en annen styling. Alle innspillene ble tatt hensyn til i videreutviklingen av systemet.

Tredje iterasjon bestod av å skaffe innsikt i lesbarhet. Dette eksisterer ikke i dagens løsning. Behov ble avdekket gjennom brukerintervjuer med markedsføringsavdelingen. De ønsket å få innsikt i hvordan ulike overflater og omgivelser påvirker lesbarheten. De ønsket også å få innsikt i hvordan kontrast og logo påvirker. Algoritmen og informasjonssiden ble implementert mot slutten av prosjektet. Funksjonaliteten ble dermed kun evaluert i sluttevalueringen med oppdragsgiver.

Etter ferdigstilt prototype ble det gjennomført en sluttevaluering med oppdragsgiver. Hensikten med denne evalueringen var å avgjøre i hvilken grad prototypen løser Favrit sine behov. Sluttevalueringen ble utført ved at oppdragsgiver fikk en demonstrasjon av den ferdigstilte prototypen. Det ble diskutert gruppens tolkning av oppgaven, hvordan denne tolkningen har formet løsningen og hvordan prosjektet ble gjennomført. Resultatene av evalueringen blir presentert i [Kapittel 5.2.](#)

5.1.2 Evaluering av kvalitet

For å evaluere om produktet er bygd med god nok kvalitet er det blitt implementert enhetstester. I tillegg er det blitt utført API-testing, manuell testing og brukertesting.

For evaluering av forretningslogikken i backend er det implementert enhetstester. Disse har som hensikt å teste isolerte deler av koden, ofte i form av en klasse i objektorienterte programmeringsspråk. Dette er for å generere synlige observasjoner som kan verifiseres. Grunnet strukturen på data som lagres i databasen må objektene som transporteres mellom frontend og

backend konverteres. Det er skrevet enhetstester som verifiserer at denne konverteringen skjer som forventet.

Verktøyet Postman er blitt benyttet for utvikling og testing av API-ene. Det er blitt laget en samling (*collection*) av forespørsler til alle API-endepunktene. Ved å legge til i testene hva som er forventet resultat av forespørslene kan alle testene kjøres og verifiseres.

Ved hver ferdigstilte MVP har klienten blitt testet internt og av uavhengige personer. Dersom det har vært avvik fra forventet flyt eller funksjonalitet har dette blitt utbedret fortløpende før brukertesting med aktuelle brukere.

Medstudenter, andre studenter utenfor studiet og familie har testet prototypen hvor hensikten er å få tilbakemelding på brukervennligheten. Gruppen definerte instruksjoner som skulle utføres. Disse instruksjonene skulle dekke flyten fra start til slutt slik at gruppen kunne observere og verifisere hele utførelsen ([se vedlegg 9.2](#)). Etter de hadde fullført instruksjonene fylte de ut et evalueringsskjema ([se vedlegg 9.3](#)).

Testing er beskrevet mer detaljert i støttedokumentet "Systemdokumentasjon" kapittel 10.

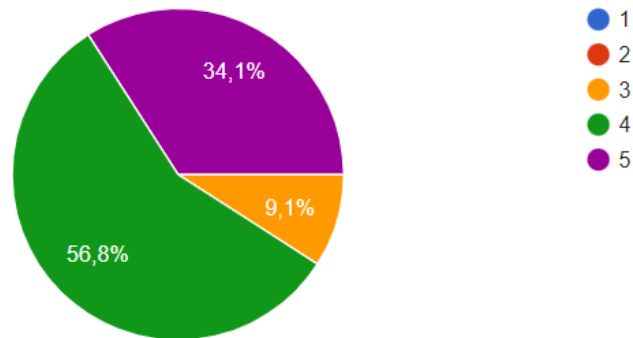
5.2 Evaluering av sluttprodukt

Oppdragsgiver var meget fornøyd med sluttproduktet. Løsningen tilbyr verdifull innsikt som Favrit ikke besitter i dag. Favrit har andre prosjekter under utvikling som tar i bruk QR-koder. På grunn av informasjonssiden i løsningen kan de benytte denne kunnskapen i andre prosjekter. Oppdragsgiver mener at det er potensiale for videreutvikling av prototypen som helhet. Funksjonaliteten som sjekker kontrast er noe som ingen andre løsninger tilbyr. Dette er noe oppdragsgiver vil videreføre uavhengig av om prototypen blir videreutviklet eller ikke. Oppdragsgiver påpekte også at de var svært fornøyd med tilnærmingen til oppgaven og hvordan gruppen har fungert som et selvstendig produktteam.

For evaluering av brukervennlighet er det blitt utført brukertesting på en variert målgruppe. Det ble utført totalt 44 tester. Resultatene av testene viste at systemet er brukervennlig (se Figur 5.1). Brukerne evaluerte systemet ved å gi en poengsum mellom 1-5 (1 er lavest og 5 er høyest) på hver av sidene i systemet. Alle resultatene fra brukertesting ligger i [vedlegg 9.4](#).

Hvordan var overordnet opplevelse av systemet? (fra 1-5)

44 svar



Figur 5.1. Resultat av brukertesting

Resultater fra brukertestene viste også at styllingsiden i systemet har forbedringspotensialet når det kommer til brukervennlighet. Det var lite intuitivt hvor en endrer mønsteret for hele QR-koden. Årsaken til dette kan skyldes at det er mange valgmuligheter. Gruppens tolkning av resultatet er at det ikke er bekymringsverdig ettersom det er første gang brukerne tester systemet. De har heller ikke forkunnskaper om systemets oppgaver i motsetning til de reelle brukerne.

5.3 Prosjektresultat

Prosjektet har resultert i en prototype av et QR-kodesystem, et eksperiment på lesbarheten til QR-koder og en rapport som dokumenterer prosessen. Resultatet har blitt formet basert på fortløpende innhenting av krav og behov fra interessenter. Basert på eksperimentet ble det utviklet en algoritme som gir tilbakemelding på lesbarheten til QR-kodene. Avslutningsvis er det formulert forslag for videre arbeid.

Rapporten har blitt utviklet kontinuerlig gjennom prosjektet. I starten presenteres litteratur og informasjon om QR-koder for å gi leser nødvendig innsikt i tematikken til prosjektet. Senere blir

det diskutert aktuelle løsninger. I den sammenheng blir det også presentert hvilken løsning gruppen valgte. Resultatet presenterer utfallet av prosjektet, som innebærer rapporten, eksperimentet og prototypen. Diskusjon tar for seg drøfting av resultatet. Kapitlet gir også leseren innblikk i hvordan prototypen ble utformet og konsekvensene av det. Til slutt gir gruppen en oversikt over deres syn på hva som kan arbeides med videre og hva som kan forbedres.

Prototypen gir omtrent de samme mulighetene som dagens løsning. En bruker får stilet en QR-kode med hensyn på de ulike komponentene til QR-koden. Videre kan brukeren lagre stylingmalen og navigere seg til listen over alle maler. Her kan det velges en standard mal for hele systemet, og det er mulighet til å slette maler. For generering av QR-koder må det legges inn en URL, og filformat kan velges.

Som følge av resultatene til eksperimentet har det blitt utviklet en informasjonsside og en algoritme. Informasjonssiden gir en oppsummering på hva som påvirker lesbarheten. Algoritmen er en løsning ingen QR-kode aktører tilbyr. Informasjonssiden og algoritmen skaper stor verdi for Favrit.

Gruppen har identifisert flere forslag for videreutvikling i prosjektet. Forslagene er et resultat av det gruppen ikke har hatt tid til å implementere og det som kan forbedre produktet. Det innebærer alt fra forbedring av lesbarhetsalgoritmen, til brukervennlighet. Forslagene gruppen identifiserte er: å utvide lesbarhetsalgoritmen, implementere opplastning av logo, implementere mulighet for dagens mønster på QR-koden og mulighet for endring av lagrede maler. Hyppig brukertesting resulterte i tilbakemeldinger på brukervennligheten. Mesteparten av tilbakemeldingene ble utbedret, de gjenstående er diskutert i videre arbeid i [Kapittel 7.2](#).

5.4 Prosjektgjennomføring

Gruppen valgte Lean startup som arbeidsmetodikk, med Critical Chain som prosjektplan. Disse valgene har gitt verdifulle resultater til prosjektet. Samarbeidet innad i gruppen, med veileder og oppdragsgiver har fungert på en god måte og gitt en god flyt på prosjektets gjennomføring. Prosjektet har vart over 5 måneder, med gradvis økende arbeidsmengde.

5.4.1 Arbeidsmetodikk - Lean startup

Gruppen har benyttet utviklingsmetodikken Lean startup som har fungert utmerket til dette prosjektet. Det har blitt utformet et MVP til hver iterasjon som deretter ble verifisert av relevante brukere før det ble videreutviklet. Iterasjonene har blitt ledet av to sentrale prinsipper innenfor Lean startup, hypotesedrevet utvikling og brukerens interesser. I starten av hver iterasjon har det blitt formulert hypoteser for funksjonalitet og design som fortløpende har blitt bekreftet eller avkreftet. Dette har fungert bra på grunn av den begrensede forkunnskapen til gruppen, og har formet løsningen etter brukerens behov. Brukerens interesser har vært hovedfokuset ved at tilnærmingen til prototypen har vært brukersentrert.

5.4.2 Prosjektplan - Critical Chain

For prosjektplanen valgte gruppen å benytte seg av Critical Chain. Gruppens valg av Critical Chain framfor Gantt-diagram var preget av usikkerheten rundt å sette start- og sluttdato for hver prosess. De fleste prosessene har holdt seg innenfor gitt antall timer. Det var derimot noen prosesser som gikk over antall timer og måtte ta timer fra bufferen, og noen som hadde timer til overs som fylte bufferen. På grunn av COVID-19 og annen sykdom ble prosjektet utsatt, som førte til en del timer å ta igjen de siste ukene av prosjektet. De fire prosessene som gikk mest over anslått tid var: *2.1 Qr styling*, *9.1 Obligatorisk aktivitet*, *2.4 Qr lesbarhet* og *5.1 Sluttrapport*. Disse prosessene tok respektivt 47 timer, 66 timer, 44 timer og 48 timer fra bufferen. De fire prosessene som fylte opp bufferen mest var: *10.1 Annet*, *8.1 Administrativt*, *2.3 lagring av Qr-koder* og *1.1 Innhenting av krav*. Disse prosessene fylte respektivt opp 38 timer, 32 timer, 25 timer og 25 timer til bufferen.

Fordi det ikke ble satt en start- og sluttdato på prosessene, men heller antall timer, unngikk gruppen at andre prosesser ble forskjøvet. Underveis i prosessen ble det bemerket at de avhengige prosessene i prosjektplanen ikke var like avhengige som først antatt. Etter *2.1 Qr styling* var utført kunne resterende prosesser bli utført parallelt. Allikevel mener gruppen at valget av Critical Chain var riktig.

5.4.3 Samarbeid

Gruppen har arbeidet meget godt, både innad i gruppen og kommunikasjonen med veileder og oppdragsgiver. Gruppen lagde tidlig i prosjektet en arbeidskontrakt for å skape gode rutiner og

hverdagsstruktur. Kontrakten ligger i Prosjekthåndboken. Den inneholdt konsekvenser dersom enkelte hendelser skulle oppstå. Dette har ført til at det har vært godt oppmøte på arbeidsplassen og svært lite forsentkomminger. Annet enn diverse sykdom underveis i prosjektet har samarbeidet fungert godt.

Gruppen har skapt et trygt miljø hvor alle har satt pris på tilbakemeldinger og evaluering av hverandres arbeid. Det ble gjennomført en intern evaluering av gruppemedlemmene i slutten av prosjektperioden. Under evalueringen ble det gitt en individuell helhetlig vurdering av hvordan gruppen har gjennomført prosjektet. Gruppen kom fram til at alle medlemmene har skapt verdi til prosjektet.

Jevnlige møter med oppdragsgiver og veileder har sørget for en trygg gjennomføring av prosjektet. Gruppen har ofte møtt veileder i gangene på skolen. Dette har gjort kommunikasjon, koordinering og tilbakemeldinger til en veldig enkel prosess som har fungert godt. Med oppdragsgiver og andre interessenter har kommunikasjon fungert godt over Slack.

6 DISKUSJON

Følgende delkapitler tar for seg tilnærmingen til hvordan prototypen ble utformet og konsekvensene av dette. Kapittelet drøfter også resultatene av prototypen.

6.1 Tilnærming

Gruppens tolkning av oppgavebeskrivelsen var at dette skulle være et tredelt system, hvor delene var styling av QR-koder, generering av QR-koder og innsikt til lesbarheten til en QR-kode.

Det ble deretter formulert mål for prosjektet. Målet var å “utvikle en prototype av et QR-kode-generator system som dekker de viktigste behovene for Favrit sin markedsføringsavdeling og support.”

Fordi alle delene av systemet er avhengig av en stilet QR-kode, var dette et naturlig startsted for utviklingen. Markedsføringsavdelingen ble innkalt til intervju for å avdekke behovene rundt funksjonalitet for styling av QR-koder. Ettersom det var mange krav til funksjonalitet ble det bestemt at løsningen måtte benytte seg av et bibliotek som dekker de viktigste funksjonene.

Etter styling-delen av systemet var ferdig evaluert og verifisert gikk gruppen over på generering av QR-koder. Her møtte gruppen på utfordringer ved ønsket arkitektur, dette blir drøftet i neste delkapittel. Siden en i gruppen hadde erfaring med generering av QR-koder, ble utforming av MVP-et basert på hypoteser. Brukertestening skulle så verifisere hypotesene.

Gruppens tilnærming til innsikt i lesbarhet tok utgangspunkt i å starte med det som er målbart. Kontrast mellom for- og bakgrunnsfarge kunne formuleres til tallverdier. Dermed ønsket gruppen å utvikle en algoritme som skulle avgjøre om kontrasten er tilfredsstillende. For å utvikle dette var det nødvendig med innsamling av data. Utover algoritmen var det ønskelig å lage en informasjonsside i systemet som skulle gi innsikt i hvilke elementer som påvirker lesbarheten. Denne siden skal også fungere som en oppskrift på hvordan en QR-kode med god lesbarhet burde utformes.

6.2 Konsekvenser av tilnærming

Gruppen valgte å benytte seg av et bibliotek for styling av QR-koder. En konsekvens av dette ble at den tenkte arkitektoniske løsningen for generering av QR-koder gjennom API-et ikke var gjennomførbart lengre. For å generere QR-koder er systemet avhengig av å bruke nedlastningsfunksjonen til biblioteket, som vil si at QR-kodene ikke kan genereres på serversiden. Løsningen ble tilpasset slik at public-API vil sende alle attributtene for stylingmalen som skal benyttes, til klienten. Dette fører til at klienten må implementere biblioteket qr-code-styling for å kunne ta imot disse. Klienten må i tillegg ta i bruk nedlastningsfunksjonen for å kunne generere QR-koden.

Derimot førte tilnærmingen til at prototypen dekker de viktigste behovene for styling og generering av QR-koder. Som følge av en iterasjonsbasert utviklingsyklus ble brukernes behov ivaretatt. Brukertesting i hver iterasjon resulterte i tilbakemeldinger som formet prototypen. For eksempel ble det avdekket behov for å navngi en mal og å velge en annen mal enn standardmalen ved generering av QR-koder.

Løsningen for innsikt i lesbarhet var tidkrevende. Gruppen måtte samle tilstrekkelig mengde data for å kunne utforme en algoritme som evaluerer kontrastnivået til en QR-kode. Dette ble gjort ved å gjennomføre et eksperiment. For å sikre et mangfold av data ble blant annet QR-kodene skannet med to ulike telefoner med ulike operativsystem. Dette skapte en avhengighet ved at alle grupped medlemmene måtte være fysisk tilstede for å utføre eksperimentet.

Som følge av eksperimentet har det blitt utviklet en algoritme som evaluerer kontrastnivået til QR-koden. Tilbakemelding på lesbarhet er en funksjonalitet som ikke eksisterer i dagens løsning. Oppdragsgiver påpekte derfor at algoritmen er noe som vil bli tatt med videre uavhengig om prototypen blir videreutviklet som helhet eller ikke.

6.3 Drøfting av resultat

Basert på oppdragsgivers forventninger har prosjektet blitt utført på en god måte og oppnådd målet. I prosjektbeskrivelsen ([se vedlegg 9.5](#)) ble det satt retningslinjer for: valg av teknologi, beskrivelse av prosjektets omfang, foreløpige funksjonelle krav, sentrale problemer med dagens løsning og at gruppen skal fungere som et selvstendig produktteam. På sluttevaluering med oppdragsgiver ble det formidlet at Aslak Hollund, som er oppdragsgiver og ekstern veileder

(sluttevaluering, 19. mai 2022), var imponert og meget fornøyd med resultatet, spesielt funksjonaliteten for å sjekke kontrastnivå. Dette bekrefter at resultatet oppnådde målet og forventningene.

Styrken som gir mest verdi er funksjonaliteten for å sjekke kontrastnivå. Denne funksjonaliteten finnes ikke i dagens løsning og gir Favrit verdifull innsikt de ikke har hatt tidligere. En annen styrke er at Favrit får generert QR-koder med unik styling på en enkel måte. Fortløpende tilbakemeldinger fra brukertesting og interessenter har resultert i forbedret opplevelse av systemet.

Det er enkelte svakheter med løsningen, men det meste er svakheter som kan utbedres eller fikses dersom man setter av tid til å implementere eller utvikle prototypen videre. Det gruppen anser som de største svakhetene er at: genereringen av QR-koder ikke kan gjøres gjennom et API, det ikke er mulig å redigere en allerede lagret mal, og at Favrit ikke har mulighet til å style QR-kodene med dagens design. Dette er ikke svakheter som er avgjørende for prosjektets resultater, men elementer som kunne blitt implementert om det var mer tid.

6.4 Refleksjon

Kunne gruppen gjort prosjektet på nytt ville de fleste valgene forbli de samme. Det ble gjort tilpasninger underveis i eksperimentet. Tilpasningene gjelder framgangsmåten for å finne grenseverdiene til fargenyansene. Underveis i eksperimentet ble det oppdaget en mer strukturert måte å definere grenseverdiene. Ved å sette en av RGB komponentene til 255 og senke de to andre med likt intervall var dette en mer systematisk tilnærming. Om gruppen skulle utført eksperimentet igjen ville dette blitt gjort fra begynnelsen av. Resultatene fra eksperimentet er fremdeles troverdige, men det kunne blitt oppnådd et enda mer nøyaktig resultat.

Gruppen ville undersøkt om hodeløs nettleser (*headless browser*) kunne vært et godt alternativ for problematikken ved generering av QR-koder gjennom et API. Hodeløs nettleser er en nettleser uten brukergrensesnitt (Chikovani, 2022). Ved å benytte dette konseptet kunne QR-kodene potensielt blitt generert ved hjelp av et skript.

7 KONKLUSJON OG VIDERE ARBEID

Følgende delkapitler tar for seg konklusjon av prosjektet og forslag til videre arbeid.

7.1 Konklusjon

Gruppen mener prosjektet har blitt løst på en tilfredsstillende måte. Nye behov er blitt innhentet fortløpende fra interessenter slik at prototypen er blitt tilpasset disse. Favrit har fått en prototype til et eget system for styling og generering av QR-koder. I tillegg gir systemet tilbakemelding på lesbarheten til en QR-kode etter styling. Prosjektet har bestått av å lage en prototype for et QR-kodesystem, med hensyn på tre definerte hovedmål. Disse målene er: gi mulighet for styling av QR-koder, gi mulighet for generering av QR-koder og gi innsikt i hva som påvirker lesbarheten til en QR-kode.

Prototypen oppfyller alle hovedmålene og ivaretar mulighetene for unikt design. Videre tillater den brukeren å generere en ny QR-kode gjennom brukergrensesnitt med gitt styling. Løsningen tilrettelegger for et alternativ til generering gjennom API, ved at andre klienter kan benytte seg av funksjonalitet som å hente en stilet mal og generere den. I tillegg gir løsningen innsikt i hva som påvirker lesbarheten til QR-koden.

Sammenligning av krav fra visjonsdokument opp mot resultatet viser at to krav ikke ble oppfylt. Funksjonalitet for opplastning av logo er ikke implementert. Endring av Favrit sin logo på QR-koden er sjeldent et behov. Dermed ble andre funksjonaliteter prioritert. En midlertidig løsning ble derfor å hardkode Favrit sin logo på alle QR-kodene. Generering av QR-koder gjennom et API ble heller ikke oppfylt. Grunnet til det var at biblioteket ikke støttet generering av QR-koder på serversiden, men gruppen kom med en alternativ løsning.

Prototypen er utviklet basert på problemstillingen og med hensyn på kravene. Ved å sammenligne resultatet og kravene opp mot problemstillingen, *“Hvordan kan Favrit få eierskap over sine QR-koder, og samtidig sikre unikt design og god lesbarhet?”*, anser gruppen prosjektet som meget godt gjennomført. Løsningen er en webapplikasjon i Favrit sitt eierskap, som sikrer et unikt design av QR-koder hvor alle elementer kan endres. Her kan brukeren lagre maler som kan bli brukt til generering av nye QR-koder. QR-kodene er sikret god lesbarhet ved hjelp av en informasjonsside og en algoritme som gir tilbakemelding på kontrast og logo. Det har blitt lagt til rette for link-redirect, ved at det kan legges inn en URL som kan peke til en link-redirect tjeneste.

7.2 Videre arbeid

Basert på konklusjonen mener gruppen prosjektet danner et godt grunnlag for videre utvikling av prototypen. Forslag for videreutvikling baserer seg på hva gruppen ville bygget videre og tilbakemeldinger fra interessenter og brukertesting.

Videreutvikling av lesbarhetsalgoritmen

Det mulig å utvide algoritmen til å gi tilbakemelding på ulike mønster, rammen rundt QR-koden og hvordan feilkorrigeringen påvirker lesbarheten.

Gruppen ville utvidet algoritmen ved å gjennomføre tilstrekkelig testing på elementene som er nevnt ovenfor. Etter tilstrekkelig testing kan algoritmen tilpasses til å ta hensyn til nye verdier.

Logo på QR-koden

Ettersom det sjeldent er behov for å endre logo ble dette nedprioritert. Favrit sin logo ble hardkodet i QR-kodene.

En løsning for dette kan være å benytte seg av en skybasert tjeneste som håndterer lagring av bilder.

Implementering av Favrit sitt mønster

Biblioteket som ble anvendt for styling av QR-kodene var begrenset til seks ulike mønster. Disse mønstrene inkluderer ikke mønsteret Favrit bruker på QR-kodene sine i dag.

Dette kan løses ved å utvide biblioteket som klienten benytter seg av. Det vil være nødvendig å skrive nye funksjoner som manipulerer datamodulene til ønsket mønster.

Endre på allerede lagrede maler

Brukeren har ikke mulighet til å gjøre endringer på en lagret mal. Et forslag kunne dermed vært å implementere funksjonalitet for dette. En "edit"-knapp kunne blitt lagt til på hver stylingmal i listen over alle malene. Ved å trykke på knappen kan brukeren få mulighet til å redigere den eksisterende stylingmalen. For å lagre endringene vil det være nødvendig å implementere et nytt endepunkt i API-et.

Utbedring av brukervennlighet

Resultatene fra brukertesting viste at stylingsiden kunne vært mer brukervennlig. Dette er ikke bekymringsverdig, ettersom det er mye informasjon i brukergrensesnittet som kan være overveldende første gang en tar i bruk systemet. I tillegg hadde ikke testbrukerne forkunnskaper om systemets oppgaver.

Et forslag kunne vært å lage en stegvis flyt for styling av QR-koder. For eksempel kan brukeren navngi og endre mønster på QR-koden i steg 1, velge for- og bakgrunnsfarge på QR-koden i steg 2, style hjørnene på QR-koden i steg 3. I siste steget kan brukeren redigere gjenværende elementer, sjekke lesbarheten og lagre QR-koden.

8 REFERANSER

Referanselisten er skrevet i Harvard referansestil.

Angular (2022) *What is Angular?*. Tilgjengelig fra: <https://angular.io/guide/what-is-angular> (Hentet 11. mars 2022).

Ardito, A. og Coppola, R. og Malnati, G. og Torchiano, M. (2020) *Effectiveness of Kotlin vs. Java in android app development tasks*. Tilgjengelig fra: <https://www.sciencedirect.com/science/article/pii/S0950584920301439> (Hentet: 07. mai 2022).

Chikovani, N. (2022) *Web Scraping with a Headless Browser: A Puppeteer Tutorial*. Tilgjengelig fra: <https://www.toptal.com/puppeteer/headless-browser-puppeteer-tutorial> (Hentet: 21. mai 2022).

Coleman, J. (2011) QR Codes: What Are They and Why Should You Care?, *Academia*, 1, s. 16-23.

DeVries, T. (2022) *The Lean Startup: How Today's Entrepreneurs Use Continuous Improvement to Create Radically Successful Businesses*. Tilgjengelig fra: <https://tylerdevries.com/book-summaries/the-lean-startup/> (Hentet: 10. mars 2022).

Dhaduk, H. (2022) Angular vs React 2022 : Which JS Framework your Project Requires?, *Web App Development*, 28. februar. Tilgjengelig fra: <https://www.simform.com/blog/angular-vs-react/> (Hentet: 11. mars 2022).

Digital didaktikk (2022) *QR- koder*. Tilgjengelig fra: <http://digitaldidaktikk.no/refleksjon/detalj/qr-koder> (Hentet 11. mars 2022).

Favrit (2022) *Welcome to Favrit*. Tilgjengelig fra: <https://careers.favrit.com/> (Hentet: 22. februar 2022).

GeeksforGeeks (2022) *Difference between TypeScript and JavaScript*. Tilgjengelig fra: <https://www.geeksforgeeks.org/difference-between-typescript-and-javascript/> (Hentet 11. mars 2022).

HVL (2022a) *Expo Bergen*. Tilgjengelig fra: <https://www.hvl.no/kalender/expo-2022/> (Hentet 3.mai 2022).

HVL (2022b) *Professor Jerry Chun-Wei Lin*. Tilgjengelig fra: <https://www.hvl.no/en/employee/?user=Chun-Wei.Lin> (Hentet 22. februar 2022).

IBM (2021) *REST APIs*. Tilgjengelig fra: <https://www.ibm.com/cloud/learn/rest-apis> (Hentet 14.mai 2022).

Leach, L.P (1999) *Critical Chain Project Management Improves Project Performance*. Sage Journals. s. 39-45.

PwC (2018) *4 tips for å lykkes med Lean Startup*. Tilgjengelig fra: <https://www.pwc.no/no/pwc-aktuelt/fire-tips-til-lykkes-med-lean.html> (Hentet: 10. mars 2022).

PwC (2021) *Lean startup, design thinking, lean. Hva er det egentlig?*. Tilgjengelig fra: <https://www.pwc.no/no/pwc-aktuelt/lean-startup-desig-thinking-hva-er-det.html> (Hentet: 10. mars 2022).

Qr Code Generator (2022) *The anatomy of a QR Code*. Tilgjengelig fra: <https://www.qr-code-generator.com/qr-code-marketing/qr-codes-basics/> (Hentet: 10. januar 2022).

React (2022) *Tutorial: Intro to React*. Tilgjengelig fra: <https://reactjs.org/tutorial/tutorial.html> (Hentet: 11. mars 2022).

Scrum (2022) *What is Scrum*. Tilgjengelig fra: <https://www.scrum.org/resources/what-is-scrum> (Hentet 11.mars 2022).

Smith, T. (2017) *The Most Common Problems With Java Ecosystem*. Tilgjengelig fra: <https://dzone.com/articles/most-common-problems-with-java-ecosystem> (Hentet: 21.mai 2022).

Spring (2022) *Building an Application with Spring Boot*. Tilgjengelig fra: <https://spring.io/guides/gs/spring-boot/> (Hentet 18. mars 2022).

Vandevenne, L. (2004) *Lode's Computer Graphics Tutorial Light and Color*. Tilgjengelig fra: <https://lodev.org/cgtutor/color.html> (hentet 20. mai 2022).

Wikipedia (2022a) *iPhone 12*. Tilgjengelig fra: https://en.wikipedia.org/wiki/IPhone_12 (Hentet: 14. mai 2022).

Wikipedia (2022b) *Samsung Galaxy S21*. Tilgjengelig fra: https://en.wikipedia.org/wiki/Samsung_Galaxy_S21 (Hentet: 14. mai 2022).

9 VEDLEGG

9.1 Risikoanalyse

	Hendelse /Risiko	Årsak	Sannsynlighet	Konsekvens	Risiko- produkt	Tiltak
1	Applikasjonen blir ikke tatt i bruk.	Ineffektiv og lite brukervennlig produkt.	Lav(2)	Høy(4)	8	Gjennomføre brukertesting.
2	Dårlige sprints/møter med oppdragsgiver og veileder.	Kaller ikke inn / dårlig forberedt til møter med oppdragsgiver og veileder.	Svært lav(1)	Middels(3)	3	Ha oppløring om møtefrekvens og innhold på møtene.
3	Endring av prosjektbeskrivelse / mål.	Oppdragsgiver eller gruppen kommer med kontramelding om oppgavens omfang.	Svært lav(1)	Svært høy(5)	5	Ha nok møter med oppdragsgiver så eventuelle endringer kommer tidlig.
4	Personalproblemer/ konflikt.	Det oppstår konflikter mellom gruppemedlemmene.	Svært lav(1)	Høy(4)	4	Ha en åpen dialog og bringe inn en tredjepart om det ikke kan løses internt.
5	Feilvurdering av oppgavens omfang.	Klarer ikke oppfylle mål og funksjonelle krav.	Middels(3)	Høy(4)	12	Gruppen går nøye igjennom hva oppgaven er og lager gode planer.
6	Gruppen har mistolket Favrit sine interesser.	For dårlig dialog med oppdragsgiver.	Lav(2)	Middels(3)	6	Ha en god dialog med oppdragsgiver.
7	For stor arbeidsmengde.	Arbeidsmengden er for stor ut ifra forventningene.	Lav(2)	Høy(4)	8	Lager gode planer og holder disse.
8	Klarer ikke å tilegne nok kunnskap om hva som påvirker lesbarheten til en QR-kode.	Har ikke klart å hente nok informasjon eller at området ikke er tilstrekkelig forsket på.	Middels(3)	Middels(3)	9	Lese nøye opp på QR-koder og passe på at alle har nok informasjon. Utføre tilstrekkelig forskning.
9	Prosjektet blir stoppet/forsinket på grunn av konflikt i Europa.	Konflikten mellom Ukraina og Russland utvider seg inkluderer NATO-landene.	Lav(2)	Middels(3)	6	Tilrettelegge for en eventuell utsettelse av oppgaven når det blir relevant.
10	Skolen stenger ned igjen.	Pandemien utvikler seg slik at skolen stenger ned.	Middels(3)	Svært lav(1)	3	Legge til rette for hjemmekontor og finne andre steder å jobbe sammen.

Sannsynlighet	Svært Høy (5)	5	10	15	20	25
	Høy (4)	4	8	12	16	20
	Middels (3)	3	6	9	12	15
	Lav (2)	2	4	6	8	10
	Svært Lav (1)	1	2	3	4	5
		Svært Lav (1)	Lav (2)	Middels (3)	Høy (4)	Svært Høy (5)
	Konsekvens					

9.2 Brukertestning instruksjer

1. Style en QR-kode med mørkeblå forgrunnsfarge.
2. Sjekk om kontrasten mellom fargene er tilfredsstillende.
3. Endre mønsteret på QR-koden til "dots"
4. Endre mønsteret på hjørnene etter ønske.
5. Lagre stylingen.
6. Naviger til siden som inneholder en oversikt over alle styling-malene.
7. Velg en mal som skal brukes for generering.
8. Naviger til siden for generering.
9. Generer en QR-kode med filtypen PNG.

9.3 Brukertestning spørreundersøkelse

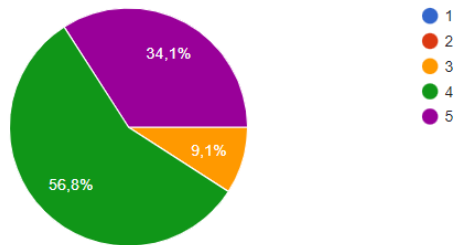
<https://forms.gle/J6VYKLeHziP7Cxvn8>

9.4 Resultat av brukertesting

Hvordan var overordnet opplevelse av systemet? (fra 1-5)

 Kopier

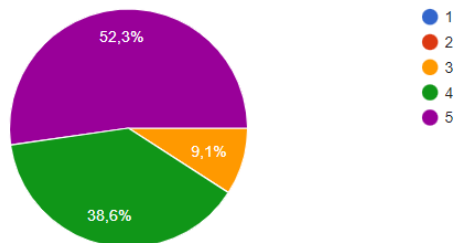
44 svar



Hvor enkelt var det å endre på fargene til QR-koden?

 Kopier

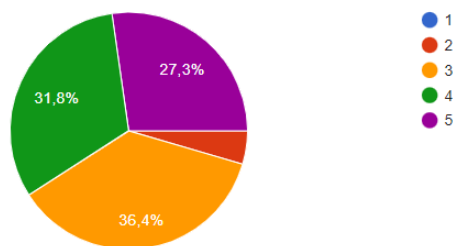
44 svar



Hvor enkelt var det å forstå hvor man endrer på mønsteret?

 Kopier

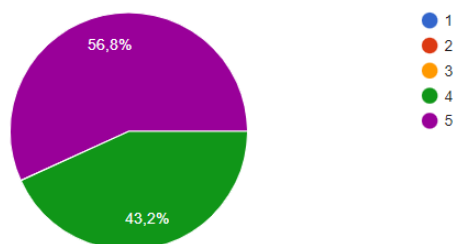
44 svar



Hvor enkelt var det å laste ned QR-koden?

 Kopier

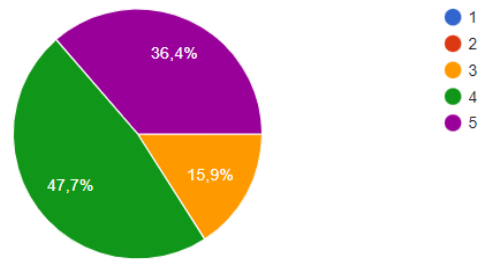
44 svar



Hvor enkelt var det å følge instruksene?

44 svar

Kopier



9.5 Prosjektbeskrivelse

Prosjektforslag til Bachelorprosjekt

ved Data- og Informasjonsteknologi-studiene for våren 2022.

QR Code Generator (favrit)

Favrit er et restaurantteknologifirma som lager point-of-sales systemer for utelivsbransjen. Favrit er et vekst-firma som i skrivende stund nærmer seg 100 ansatte. Hovedkontoret til favrit er i Oslo og firmaet har ansatte i flere andre norske byer, samt Sverige, USA og Storbritannia.

<https://favrit.com/>

Oppgave

Bakgrunn for prosjektet:

En stor del av favrits brukere kommer inn i løsningen ved hjelp av QR-koder som skannes med mobilen. Favrit bruker i dag en tredjeparts-løsning for å lage QR-koder, samt å håndtere trafikken som kommer inn via disse kodene, men ønsker å flytte dette til selveide systemer.

Tredjepartsløsningen som brukes i dag har to hovedfunksjoner.

1. Den genererer QR-koder, og tilbyr et grensesnitt for å tilpasse disse med forskjellige farger, former, detaljer og bilder.
2. Den har en link-redirect tjeneste som gjør at en QR-kode kan peke til en statisk url, men vi kan endre hva denne urlen peker videre på.

Vi har opplevd tre sentrale problemer med tredjepartsløsningen som motiverer oss til å prioritere å bytte den ut.

1. Vår oppetid har en ekstern avhengighet. Dette er den aller viktigste grunnen til at vi ønsker kontroll over denne delen av systemet vårt. Når tredjepartsløsningen er nede fungerer ikke våre QR-koder.
2. Tredjepartsbiblioteket har mulighet til å generere QR-koder via API, men disse QR-kodene blir statiske og ikke koblet opp til link-redirect løsningen.
3. Løsningen gir mulighet til å tilpasse designet på QR-kodene uten å gi noe feedback på lesbarheten på resultatet av valgene som gjøres i designet. Kvaliteten på QR-koden er en viktig del av den opplevde kvaliteten til produktet vi tilbyr til våre kunder, og det er derfor viktig for oss at vi opprettholder god lesbarhet på QR-kodene. Samtidig så er QR-kodene en av våre viktigste branding-flater, så vi ønsker å ta i bruk mulighetene som eksisterer til å tilpasse look-og-feel av QR-kodene.

Beskrivelse av hva prosjektet skal løse:

For dette prosjektet velger vi å fokusere på den første hovedfunksjonen, nettopp QR-kode genereringen.

Prosjektet skal kunne lage QR-koder som fungerer som gode aksesspunkter til tjenesten vår, samt tjener våre markedsføringsbehov. Vi ønsker også muligheten til å evaluere kvalitet/lesbarhet til genererte QR-koder.

Selv om prosjektet ikke skal løse problemstillingene relatert til link-redirect, så bør man planlegge for at det skal utvides til å dekke dette behovet ved en senere anledning.

Beskrivelse av teknologier / metoder og annet som er tenkt brukt i prosjektet:

Vi har sett for oss React, typescript frontend og en kotlin, spring-boot backend. Dette kan eventuelt diskuteres.

Beskrivelse av arbeidsoppgaver for studentene i prosjektet:

Studentene vil utgjøre et uavhengig produkt-team som, med støtte fra favoritt skal sørge for egen fremdrift i prosjektet. Studentene vil måtte koordinere med stakeholders i organisasjonen, samt innhente krav fortløpende. Informasjonen som eksisterer i dette dokumentet vil fungere som en base, men produktet skal til en viss grad også formes av gruppen. Som et konkret eksempel kommer studentene til å måtte gå i dialog med markedsføringsavdelingen for å finne deres behov for tilpasninger av utformingen av QR-kodene.

Foreløpig funksjonelle krav:

Favrit er en bedrift i konstant endring, og det er rimelig å tenke dette prosjektet også må være tilpasningsdyktig. Når det er sagt så kommer her en liste med krav som vi føler det er rimelig å forholde seg til i prosjektet. Som sagt må studentene som jobber med dette prosjektet ha et aktivt forhold til kravene, og tilpasse prosjektet de behovene de ser i organisasjonen.

- Løsningen må kunne lage QR-koder som encoder strenger (urler).
- Det må være mulighet til å lage QR-koder både igjennom et grensesnitt og igjennom et API.
- Det bør være et skille mellom public-API (som kan brukes i organisasjonen) og internal-API (som bare brukes av teamets egne løsninger).
- Senere skal linking til lokasjoner i favrits systemer være løst av link-redirect tjenesten, men for at produktet skal være en fungerende prototype bør systemet kunne ta imot lokasjons-Ider og lage QR-koder som linker direkte til lokasjon. Om lokasjonen alt har en QR-kode i systemet skal eksisterende kode returneres.
- Systemet bør kunne lage QR-koder som kan brandes til våre behov. Hvor lett det skal være å lage nye design må diskuteres med markedsføring.
- Systemet bør også kunne gi en vurdering av det valgte designet og si noe om lesbarheten til resulterende QR-kode.

Annet:

Om scopet på oppgaven skulle vise seg å være for lite for perioden vil det være mulig for gruppen å arbeide med link-redirect problematikken som er beskrevet i bakgrunnen for oppgaven.

Arbeidsspråket i favrit er engelsk, men de fleste stakeholders snakker Norsk. Så i møter vil det mest sannsynlig gå fint å snakke Norsk, men i offentlige slack-kanaler og lignende bør man holde seg til Engelsk.

Vi ønsker at teamet i stor grad er uavhengig, og de vil få autoriteten nødvendig til å løse oppgaven slik de ønsker. Favrit tilbyr business caset, og så vil det være gruppens oppgave å løse det. For å støtte teamet i dette kommer studentene til å ha tilgang på ressurser som vi har tilgjengelig i firmaet i form av fagkompetanse og tilgang på stakeholders. Vi ønsker å ha oppfølgingsmøter hver andre uke, og i startfasen vil vi også bidra med støtte til arkitektur og teknologivalg. Selv om teamet eier løsningen under utviklingen, er det selvsagt viktig at resultatet blir noe vi kan bruke og videreutvikle videre.

9.6 Resultater fra eksperiment

3	Forgrunnsfarge	Digital skjerm		Papir - Naturlig lys		Papir - dempet belysning		Laminat - naturlig lys		Laminat - dempet belysning	
		iPhone	Android	iPhone	Android	iPhone	Android	iPhone	Android	iPhone	Android
5	Mindre grønt (rosa)										
6	255, 210, 255	Nei	Nei	Ja - zoom ut	Ja	Nei	Nei	Ja - zoom ut	Ja	Nei	Nei
7	255, 205, 255	Nei	Nei	Ja - zoom ut	Ja	Nei	Ja	Ja - zoom ut	Ja	Nei	Ja
8	255, 190, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
9	255, 175, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
10	255, 165, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
11	255, 155, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
12											
13	Mindre rødt (turkis)										
14	215, 255, 255	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
15	210, 255, 255	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
16	205, 255, 255	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
17	200, 255, 255	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
18	190, 255, 255	Nei	Nei	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
19	180, 255, 255	Nei	Nei	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
20	170, 255, 255	Nei	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
21	160, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
22	150, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
23	140, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
24	130, 255, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
25											
26	Mindre blå (gul)										
27	255, 255, 200	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
28	255, 255, 170	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
29	255, 255, 100	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
30	255, 255, 0	Nei	Ja	Ja - zoom ut	Ja	Nei	Ja	Ja - zoom ut	Ja	Nei	Ja
32	Forskjellige farger > 100										
33	183, 255, 227	Nei	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
34	255, 212, 198	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
35	210, 237, 218	Nei	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
36	210, 222, 232	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
37	224, 205, 236	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
38	198, 255, 232	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
39	255, 222, 208	Nei	Ja	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
40	220, 242, 223	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
41	220, 232, 232	Nei	Ja	Ikke sjekket - feil	Ikke sjekket - feil	x	x	Ikke sjekket - feil	Ikke sjekket - feil	x	x
42	234, 205, 246	Nei	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
43	213, 255, 237	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
44	255, 232, 218	Nei	Ja	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
45	230, 247, 228	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
46	225, 242, 237	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
47	239, 215, 251	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
48	223, 255, 247	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
49	255, 242, 228	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
50	236, 255, 234	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
51	232, 250, 243	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
52	245, 225, 255	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei

Vedlegget viser resultatene fra skanning av ensfargede QR-koder

3	Forgrunnsfarge	Hjørnefarge	Digital skjerm		Papir - Naturlig lys		Papir - dempet belysning		Laminat - naturlig lys		Laminat - dempet belysning	
			iPhone	Android	iPhone	Android	iPhone	Android	iPhone	Android	iPhone	Android
4												
5	Mindre grønt (rosa)											
6	255, 225, 255	255, 175, 255	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
7	255, 220, 255	255, 170, 255	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
8	255, 215, 255	255, 165, 255	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
9	255, 210, 255	255, 160, 255	Nei	Nei	Ja	Ja	Nei	Ja	Nei	Ja	Nei	Ja
10	255, 205, 255	255, 155, 255	Nei	Nei	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
11	255, 200, 255	255, 150, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
12	255, 190, 255	255, 140, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
13	255, 175, 255	255, 125, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
14	255, 165, 255	255, 115, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
15												
16	Mindre rødt (turkis)											
17	230, 255, 255	180, 255, 255	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
18	225, 255, 255	175, 255, 255	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
19	220, 255, 255	170, 255, 255	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
20	215, 255, 255	165, 255, 255	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
21	210, 255, 255	160, 255, 255	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
22	205, 255, 255	155, 255, 255	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
23	200, 255, 255	150, 255, 255	Nei	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
24	190, 255, 255	140, 255, 255	Nei	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
25	180, 255, 255	130, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
26	175, 255, 255	125, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
27	170, 255, 255	120, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
28	160, 255, 255	110, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
29	150, 255, 255	100, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
30	140, 255, 255	90, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
31	130, 255, 255	80, 255, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
32												
33	Mindre blå (gul)											
34	255, 255, 200	x	x	x	x	x	x	x	x	x	x	x
35	255, 255, 100	x	x	x	x	x	x	x	x	x	x	x
36	255, 255, 0	x	x	x	x	x	x	x	x	x	x	x
38	Forskjellige farger >= 100											
39	183, 255, 227	163, 245, 207	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
40	255, 212, 198	245, 192, 178	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
41	210, 237, 218	190, 227, 198	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
42	210, 222, 232	200, 212, 212	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
43	224, 205, 236	214, 185, 216	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
44	198, 255, 232	178, 245, 212	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
45	255, 222, 208	245, 202, 188	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
46	220, 242, 223	200, 232, 203	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
47	220, 232, 232	210, 212, 212	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
48	234, 205, 246	219, 185, 231	Nei	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
49	213, 255, 237	193, 240, 222	Nei	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
50	255, 232, 218	245, 212, 198	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
51	230, 247, 228	210, 237, 208	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
52	225, 242, 237	209, 230, 215	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
53	239, 215, 251	229, 195, 231	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
54	223, 255, 247	203, 245, 227	Nei	Ja	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
55	255, 242, 228	245, 222, 208	Nei	Ja	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
56	236, 255, 234	217, 240, 218	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
57	232, 250, 243	222, 230, 223	Nei	Ja	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
58	245, 225, 255	225, 215, 235	Nei	Ja	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei

Vedlegget viser resultatene fra skanning av QR-koder med 50 lavere summert RBG på *position detection markers*.

3	Forgrunnsfarge	Digital skjerm		Papir - Naturlig lys		Papir - dempet belysning		Laminat - naturlig lys		Laminat - dempet belysning	
		iPhone	Android	iPhone	Android	iPhone	Android	iPhone	Android	iPhone	Android
4											
5	Mindre rødt (turkis)										
6	0, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
7	50, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
8	100, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
9	110, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
10	120, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
11	130, 255, 255	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
12	140, 255, 255	Nei	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
13	150, 255, 255	Nei	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
14	160, 255, 255	Nei	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
15	170, 255, 255	Nei	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Nei	Ja
16	180, 255, 255	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
17	190, 255, 255	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
18											
19											
20	Mindre grønt (rosa)										
21	255, 140, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
22	255, 150, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
23	255, 160, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
24	255, 170, 255	Nei	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
25	255, 180, 255	Nei	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
26	255, 190, 255	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
27	255, 200, 255	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
28											
29											
30	Mindre blå (gul)										
31	255, 255, 0	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
32	255, 255, 50	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Ja
33	255, 255, 100	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
34	255, 255, 125	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
35	255, 255, 150	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
38	Forskjellige farger										
39	255, 165, 165	Ja	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
40	255, 175, 175	Ja	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
41	255, 185, 185	Ja	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
42	255, 195, 195	Nei	Nei	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
43	255, 205, 205	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
44	255, 215, 215	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
45	255, 225, 225	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
46	255, 235, 235	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
47	135, 255, 135	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
48	145, 255, 145	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
49	155, 255, 155	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
50	165, 255, 165	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
51	175, 255, 175	Nei	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
52	185, 255, 185	Nei	Ja	Ja	Ja	Nei	Ja	Nei	Ja	Nei	Ja
53	195, 255, 195	Nei	Nei	Nei	Ja	Nei	Ja	Nei	Ja	Nei	Nei
54	205, 255, 205	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
55	215, 255, 215	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
56	225, 255, 225	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
57	235, 255, 235	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
58	185, 185, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
59	195, 195, 255	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
60	205, 205, 255	Nei	Nei	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
61	215, 215, 255	Nei	Nei	Ja	Ja	Nei	Nei	Ja	Ja	Nei	Nei
62	225, 225, 255	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
63	235, 235, 255	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei
64	195, 195, 195	Ja	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
65	205, 205, 205	Ja	Ja	Ja	Ja	Nei	Ja	Ja	Ja	Nei	Ja
66	215, 215, 215	Nei	Ja	Ja	Ja	Nei	Nei	Ja	Ja	Nei	Nei
67	220, 220, 220	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
68	225, 225, 225	Nei	Nei	Nei	Ja	Nei	Nei	Nei	Ja	Nei	Nei
69	230, 230, 230	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei	Nei

Vedlegget viser resultatene fra skanning av ensfargede QR-koder med logo

