



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Chorium - En applikasjon for enklere
organisering av kor

Chorium - An Application for Easier
Organization of Choirs

Andreas Conradi Nitschke

Bjørn Egil Øygarden

Bachelor i ingeniørfag, data

Institutt for datateknologi, elektroteknologi og realfag

Veileder: Atle Birger Geitung

13.05.2024

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> Chorium - En applikasjon for enklere organisering av kor	<i>Dato:</i> 13.05.2023
<i>Forfatter(e):</i> Andreas Conradi Nitschke og Bjørn Egil Øygarden	<i>Antall sider u/vedlegg:</i> 43
	<i>Antall sider vedlegg:</i> 8
<i>Studieretning:</i> Dataingeniør	<i>Antall disketter/CD-er:</i> Ingen
<i>Kontaktperson ved studieretning:</i> Atle Birger Geitung	<i>Gradering:</i> Ingen
<i>Merknader:</i>	

<i>Oppdragsgiver:</i> Egendefinert	<i>Oppdragsgivers referanse:</i> Ingen
<i>Oppdragsgivers kontaktperson:</i> Andreas Conradi Nitschke Bjørn Egil Øygarden	<i>Telefon:</i> +47 97417676 +47 41091197

Sammendrag:

Målet med dette bachelorprosjektet er å utvikle en applikasjon som skal hjelpe kor med organisering av lydfiler, noter, arrangementer og kommunikasjon innad i koret.

Stikkord:

Flutter	Mobilapplikasjon	Øving og organisering
---------	------------------	-----------------------

FORORD

Denne rapporten tar for seg planlegging, utviklingen og evalueringen av bachelorprosjektet “Chorium - En applikasjon for enklere organisering av kor”, gjennomført våren 2024 ved Høgskulen på Vestlandet, studieretning Dataingeniør. Prosjektet er et egendefinert initiativ som har som mål å utvikle en mobilapplikasjon til bruk for kororganisasjoner. Applikasjonen er tenkt som en helhetlig løsning for de nødvendige tjenester et kor kan trenge for organisering, øving og kommunikasjon. Prosjektet er gjennomført av Andreas Conradi Nitschke og Bjørn Egil Øygarden.

Vi ønsker å takke vår veileder, Atle Birger Geitung, som foreslo å gjennomføre denne applikasjonen som et bachelorprosjekt. Vi vil også takke Studinekoret Kalliope og Rebellion som var med på å teste applikasjonen og gi verdifulle tilbakemeldinger under prosessen.



INNHALDSFORTEGNELSE

FORORD	III
ORDLISTE	VI
1. INNLEDNING	1
1.1 KONTEKST	1
1.2 MOTIVASJON.....	1
1.3 PROSJEKTEIER.....	2
1.4 PROBLEMBESKRIVELSE OG MÅL	2
1.5 OPPBYGGING AV RAPPORTEN.....	4
2. PROSJEKTBEKRIVELSE	6
2.1 PRAKTISK BAKGRUNN	6
2.1.1 Tidligere arbeid.....	6
2.1.2 Initielle krav.....	6
2.2 AVGRENSNINGER	10
2.3 RESSURSER	10
3. DESIGN AV PROSJEKTET	11
3.1 FORSLAG TIL LØSNING	11
3.1.1 Native.....	11
3.1.2 Web-basert.....	11
3.1.3 Hybridløsning	12
3.2 VALGT LØSNING	12
3.3 VALG AV VERKTØY	13
3.4 PROSJEKTMETODIKK	14
3.4.1 Utviklingsmetodikk.....	14
3.4.2 Prosjektplan	15
3.5 EVALUERINGSPLAN	16
4. DETALJERT LØSNING	18
4.1 SYSTEMOVERSIKT.....	18
4.2 BRUKERGRENSESNIITT	20
4.2.2 Designprosess	21
4.2.3 Brukerflyt	21
4.2.4 Iterasjon 1	22
4.2.5 Iterasjon 2	24
4.3 SYSTEMARKITEKTUR	25
4.4 DEPLOYMENT	26
5. RESULTATER	29
5.1 EVALUERINGSMETODE	29
5.1.1 Brukertestning.....	29
5.1.2 Design av brukerundersøkelse	29
5.2 EVALUERINGSRESULTAT	30
5.2.1 Analyse av resultater fra brukerundersøkelse	30
5.2.2 Iterasjon 2	32
5.3 PROSJEKTGJENNOMFØRING	32
6. DISKUSJON	34



6.1 ANALYSE AV PROSJEKTRISULTATER	34
6.2 VURDERING AV TILNÆRMINGER OG VALG	36
6.3 HELHETLIG SYSTEMPERSPEKTIV	37
6.4 OPPSUMMERING	37
7. KONKLUSJON OG VIDERE ARBEID	38
7.1 KONKLUSJON	38
7.2 VIDERE ARBEID	38
<i>Integrering av Selvregulert Læring</i>	39
8. REFERANSER	41
8.1 ILLUSTRASJONER	43
9. VEDLEGG	44
VEDLEGG A: BRUKERUNDERSØKELSE OPPSETT	44
VEDLEGG B: RESULTATER FRA BRUKERUNDERSØKELSE ITERASJON 1	45

Ordliste

Begrep	Forklaring
BLAPP	Tidligere prosjekt av Andreas. https://play.google.com/store/apps/details?id=no.blak.blapp
Design Science Research (DSR)	Utviklingsmetodikk med fokus på design og utvikling av innovative løsninger. Basert på brukeropplevelse og jevnlig evaluering.
Klient	Klienten, eller front-end, er delen av en applikasjon som brukeren direkte interagerer med. Inneholder et brukergrensesnitt.
Back-end	Back-end refererer til serverdelen av en nettside eller applikasjon som håndterer databehandling, lagring og server-logikk. Den fungerer som 'motoren' som kjører applikasjonen og håndterer brukerforespørsler fra front-end.
iOS	Operativsystem for mobile enheter produsert av Apple
Appstore	Plattform for distribusjon av applikasjoner for Apple-enheter.
Android	Android er et operativsystem for mobile enheter utviklet av Google, og det brukes av en rekke mobilprodusenter verden over. Det er designet for smarttelefoner og nettbrett og er det mest brukte mobile operativsystemet globalt.
Google Play	Plattform for distribusjon av applikasjoner til Android-enheter.
Web	Global plattform for dokumenter og applikasjoner, brukes via nettleser.
Rammeverk	En samling forhåndsskrevne koder, verktøy og konvensjoner som brukes til å utvikle applikasjoner.
Flutter	Et rammeverk utviklet av Google for å effektivt lage applikasjoner for flere plattformer fra samme kildekode.
Firebase	En plattform utviklet av Google som tilbyr et bredt spekter av back-end-tjenester for applikasjoner.
Crashlytics	Verktøy som brukes for å spore, logge og analysere applikasjonskrasj.
Google Cloud Platform	Samling av skybaserte tjenester som tilbyr hosting og verktøy for databehandling, datalagring, dataanalyse, m.m. Gir bedrifter muligheten til å kjøre applikasjoner og lagre data på Googles infrastruktur.
Git	Git er et distribuert versjonskontrollsystem som lar utviklere effektivt spore og håndtere endringer i kodebasen, samarbeide på prosjekter og vedlikeholde forskjellige versjoner av filer og applikasjoner



Utviklingsmiljø	En samling verktøy og ressurser brukt av programvareutviklere for å skrive, teste og feilsøke programvare.
Emulator	En emulator er et program som etterligner et annet system, slik som en Android-emulator som lar utviklere teste og kjøre mobilapplikasjoner på en datamaskin som om det var en fysisk mobil enhet.
Visual Studio Code	Lettvekts, fleksibel teksteditor med et stort utvalg utvidelser som muliggjør utvikling av mange typer programvare.
Android Studio	Integrert utviklingsmiljø nødvendig for utvikling av Android-applikasjoner. Tilbyr emulatorer og andre verktøy.
Xcode	Integrert utviklingsmiljø av Apple som er nødvendig for utvikling av iOS-applikasjoner.
Figma	Nettbasert designverktøy som brukes for å skape brukergrensesnitt og prototyper for web- og mobilapplikasjoner.
KanBan	En smidig prosjektstyringsmetode som fokuserer på å forbedre effektiviteten ved å visualisere arbeidsflyten, begrense arbeid under behandling og maksimere gjennomstrømningen av oppgaver i et produksjonssystem
System Usability Scale (SUS)	Enkelt, ti-punkts spørreskjema som brukes til å evaluere brukervennligheten av et system eller en applikasjon.
Likert-skala	En spørreskjema-skala som brukes til å måle holdninger eller meninger. Den består av en serie påstander som respondenter vurderer på en skala, ofte fra "sterkt uenig" til "sterkt enig"
Apple Testflight	Apple sitt tilbud for testing av applikasjoner til brukere utenfor organisasjonen.
Unified platform	En integrert løsning som kombinerer flere verktøy, systemer, eller tjenester innenfor én enkelt plattform.

1. INNLEDNING

Dette kapitlet tar for seg bakgrunnen for prosjektet. Her vil man få en god ide om konteksten bak prosjektet og hvilke problemer som skal løses.

1.1 Kontekst

I vårt raskt utviklende samfunn spiller musikk- og kulturaktiviteter, slik som kor, en viktig rolle som samlingspunkter for fellesskap, kreativitet og kulturell utveksling. En studie publisert i *Current Biology* avdekker at musikk besitter universelle egenskaper som binder mennesker sammen på tvers av kulturelle grenser, noe som understreker dens essensielle rolle i å fremme sosial samhörighet og kulturelt uttrykk i ulike samfunn. (A. Mehr et al., 2018). Denne universelle tiltrekningskraften understreker betydningen av musikk for å fremme fellesskap, samhörighet og legge til rette for kulturelle utvekslinger.

I dagens teknologiske landskap er det betydelige muligheter for innovasjon innen organisering, administrasjon og øvingskultur for koraktiviteter. Imidlertid har mange kor, basert på egne og bekjentes erfaringer, møtt utfordringer med effektiv kommunikasjon og tilgang til essensielle ressurser som noter og øvingsfiler. Disse utfordringene fremhever behovet for bedre integrerte løsninger.

Flere organisasjoner er tvunget til å benytte seg av et mangfold av plattformer for å imøtekomme behovene som gjelder organisering, planlegging og distribusjon av øvingsmateriale. Dessverre er ikke disse løsningene alltid tilpasset de spesifikke behovene et kor har, noe som kan resultere i forvirring rundt hvor informasjon og nødvendige filer befinner seg. Dette kan i sin tur føre til redusert øvingseffektivitet hvis medlemmene sliter med å finne frem til de nødvendige ressursene.

1.2 Motivasjon

Motivasjonen bak oppstarten av dette prosjektet springer ut fra direkte forespørsler og en anerkjennelse av de eksisterende utfordringene innen kororganisasjoner. Disse utfordringene omfatter blant annet ineffektive kommunikasjonskanaler, vanskeligheter med å tilgjengeliggjøre og administrere noter og lydfiler, samt utfordringene med å holde medlemmer engasjert og informert. Slike problemer har betydelige implikasjoner for både det musikalske utbyttet og det sosiale fellesskapet i korene.

Prosjektet er drevet av et behov for å forenkle og forbedre de administrative prosessene og øke tilgjengeligheten av øvingsmateriale gjennom en helhetlig digital plattform

spesifikt designet for korvirksomhet. Ved å tilby en sentralisert løsning, der alle nødvendige funksjoner for korstyring er samlet i én applikasjon, siktes det mot å styrke samarbeidet innad i korene og fremme en mer effektiv øvingskultur.

Det endelige målet med prosjektet er å utvikle en applikasjon som imøtekommer de samlede behovene til kororganisasjoner på en måte som er intuitiv og brukervennlig. For å sikre at løsningen møter brukernes behov og forventninger, vil det utføres brukertester i flere iterasjoner gjennom utviklingsfasen. Tilbakemeldinger hentet fra disse testene vil spille en kritisk rolle i den videre utviklingen av applikasjonen, med et særlig fokus på brukervennlighet, funksjonalitet og tilpasningsevne til korenes varierte behov.

Denne tilnærmingen viser en inngående forståelse av de utfordringene kor møter, og en dedikasjon til å utvikle løsninger som ikke bare adresserer disse problemene, men også forbedrer koropplevelsen for deltakerne. Ved å tilpasse teknologiske løsninger til de unike behovene innen korvirksomhet, tar prosjektet viktige skritt mot en mer integrert og ressurseffektiv praksis for koraktiviteter.

1.3 Prosjekteier

Prosjektet springer ut fra et personlig prosjekt av Andreas, som ville utvide arbeidet med en applikasjon han tidligere hadde utviklet, til et fullverdig bachelorprosjekt. Andreas, med sin tidligere investering i applikasjonen, inntar naturlig rollen som prosjekteier. Gjennom sitt engasjement i studentkor under studietiden, oppdaget Andreas at organiseringen innad i koret ikke var optimal. Dette førte til utviklingen av en dedikert applikasjon ment å løse organisatoriske utfordringer for det spesifikke koret han var en del av. Bachelorprosjektet har som mål å ta utgangspunkt i denne originale ideen og utvide den med mer funksjonalitet, slik at den kan tjene et bredere spekter av kor og musikkorganisasjoner. Andreas planlegger å fortsette arbeidet med å utvikle applikasjonen også etter at bachelorprosjektet er fullført, noe som gjør han spesielt investert i prosjektets suksess.

1.4 Problembeskrivelse og mål

Basert på erfaringer står mange kor overfor betydelige utfordringer med effektiviteten i øvingsprosesser, hvor tilgang til organiserte øvingsfiler og kommunikasjon ofte er spredt over ulike plattformer. Dette kan føre til forvirring, økt tidsbruk på administrasjon fremfor musikk, og potensielt lavere deltakelse og motivasjon blant medlemmene. Disse utfordringene danner grunnlaget for utviklingen av en dedikert applikasjon designet for å møte korvirksomheters spesifikke behov. For å undersøke potensialet og effekten av en slik løsning, har vi formulert følgende forskningsspørsmål:

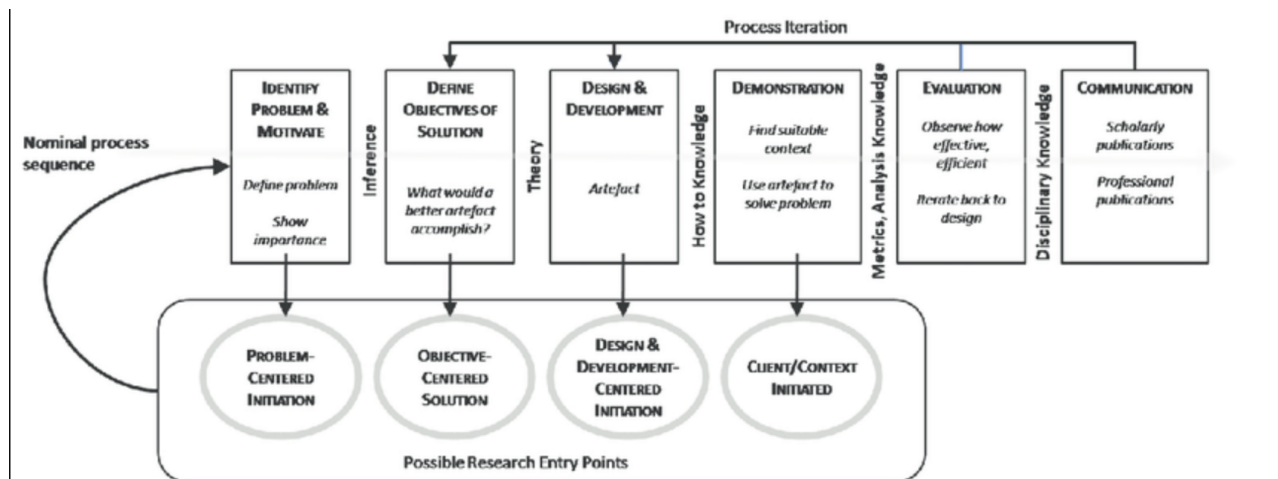
1. Kan en dedikert applikasjon øke effektiviteten for øving og organisering av kor sammenlignet med nåværende metoder?
2. I hvilken grad øker applikasjonen tilgjengeligheten av øvingsmateriale?
3. Hvordan påvirker applikasjonen kormedlemmenes motivasjon for individuell øving?
4. Hvordan bidrar applikasjonen til bedre organisering og planlegging av øvinger og andre arrangementer?
5. På hvilken måte forbedrer applikasjonen kommunikasjon mellom ledere og kormedlemmer?

Mål for prosjektet

- Kortsiktig mål:
 - Utvikle en brukervennlig applikasjon som sentraliserer øvingsfiler, kommunikasjon, og kalenderfunksjoner for et kor.
 - Dette initiativet vil direkte adressere de identifiserte utfordringene og bidra til en dypere forståelse om hvordan teknologi kan forbedre korvirksomheten.
- Langsiktige mål
 - Skape en standardisert plattform som kan adopteres av kor, for å forbedre organisatorisk effektivitet og styrke kormedlemmenes engasjement og tilfredshet.
 - Gjennom å besvare de ovennevnte forskningsspørsmålene, vil prosjektet bidra med viktig innsikt i hvordan en slik plattform kan designes, implementeres, og optimaliseres for å møte korenes behov.

Design Science Research (DSR)

For å nå disse målene, benytter vi Design Science Research ([Brocke et al., 2020](#)) som vår metodologiske tilnærming. DSR er valgt på grunn av dens fokus på å utvikle og evaluere innovative løsninger som effektivt løser identifiserte problemer. Dette gir oss en strukturert ramme for å kombinere teoretisk innsikt med praktisk anvendelse gjennom iterativ utvikling og evaluering.



Figur 1: Design Science Research Methodology (Peffers, 2008)

1.5 Oppbygging av rapporten

Denne rapporten er bygd opp på følgende måte:

Kapittel 1 - Innledning:

- Kapittel 1 gir en introduksjon til prosjektet og problemstillingen, samt kontekst bak prosjektet og hvorfor det gjennomføres.

Kapittel 2 - Prosjektbeskrivelse:

- Kapittel 2 beskriver prosjektet og hvordan en løsning på problemet kan se ut.

Kapittel 3 - Design av prosjektet:

- Kapittel 3 går i dybden på hvordan prosjektet skal løses, og planen for hvordan utviklingen skal gjennomføres og til slutt evalueres.

Kapittel 4 - Detaljert løsning

- Kapittel 4 tar for seg å beskrive den gjennomførte løsningen i detalj.

Kapittel 5 - Resultater

- Dette kapitlet går over evalueringsmetodikk og påfølgende resultater av prosjektet.

Kapittel 6 - Diskusjon

- I dette kapitlet vil de bli diskutert styrker og svakheter med prosjektets resultat.

Kapittel 7 - Konklusjon og videre arbeid

- Kapittel 7 består av konklusjonen av gjennomført arbeid, og man vil også få en oversikt over hvordan prosjektet kan videreføres.

Kapittel 8 - Referanser

- Dette kapitlet gir en oversikt over kilder brukt i rapporten.

Kapittel 9 - Vedlegg

- Kapittel 9 er en samling av vedlegg referert til i rapporten

2. PROSJEKTBESKRIVELSE

Dette kapitlet tar for seg prosjektets grunnlag, samt ser på diverse tekniske krav og praktiske overveielser.

2.1 Praktisk bakgrunn

2.1.1 Tidligere arbeid

Prosjektet bygger på et tidligere utviklingsarbeid utført av et av gruppemedlemmene, hvor en applikasjon ble laget for et kor han deltar i. (["BLAPP," 2022](#)) Denne applikasjonen fokuserte hovedsakelig på øvingsrelaterte funksjoner, som å gjøre det mulig for brukerne å lytte til lydfiler samtidig som de ser på noter. Dette var en funksjonalitet som tidligere løsninger, som Google Drive og Dropbox, ikke kunne tilby. Mottakelsen var svært positiv, og flere andre kor uttrykte raskt interesse for å benytte et slikt verktøy.

Den opprinnelige applikasjonen ble utviklet med begrenset erfaring, noe som medførte flere utfordringer. Brukergrensesnittet hadde mindre fokus på brukervennlighet, ettersom utvikleren Andreas, som også skulle administrere koret fremover, ikke hadde behov for et avansert grensesnitt. Sikkerhet ble også ansett som mindre kritisk da applikasjonen var ment for en lukket gruppe på omtrent 30 medlemmer som kjente hverandre godt.

På grunn av disse begrensningene, inkludert det faktum at nesten all funksjonalitet var klientsidebasert, var applikasjonen sårbar for sikkerhetsproblemer og mindre effektiv i større skala. Tilbakemeldinger fra brukerne ble kontinuerlig samlet inn, noe som ga verdifull innsikt, men det var tydelig at en ny, mer robust løsning var nødvendig for å møte behovene til et bredere publikum.

I tråd med prinsippene i Design Science Research ([Brocke et al., 2020](#)) ble tilbakemeldinger fra brukerne kontinuerlig samlet inn og analysert. Denne iterative prosessen ga verdifull innsikt i nødvendige forbedringer og nye funksjoner som kunne integreres i en ny applikasjon. Resultatene fra disse tilbakemeldingene viste et behov for en mer robust løsning som kunne møte kravene til et bredere publikum, med økt fokus på brukervennlighet og sikkerhet.

2.1.2 Initielle krav

Prosjektet startet med et klart sett av funksjonelle krav, med mål om å integrere flere funksjoner som for øyeblikket er spredt over forskjellige tjenester. Målet med applikasjonen er å skape en helhetlig plattform som forenkler organiseringen og øvingen

for kor, ved å samle alle nødvendige verktøy i én applikasjon. Her er de spesifikke kravene og ideene som ble stilt til prosjektet:

1. Opplasting og lagring av noter

- **Beskrivelse:** Applikasjonen skal tillate brukere å laste opp og lagre noter i et format som er lett tilgjengelig for alle medlemmer.
- **Begrunnelse:** Denne funksjonaliteten forenkler tilgangen til noter under øvinger og opptredener.

2. Opplasting av sanger

- **Beskrivelse:** Brukere skal kunne laste opp lydfiler av sanger.
- **Begrunnelse:** Dette gir medlemmer enkel tilgang til øvingsmateriale direkte i applikasjonen.

3. Spille av lydfiler sammen med notene:

- **Beskrivelse:** Lydfiler skal kunne spilles av mens noter vises på skjermen.
- **Begrunnelse:** Dette gjør det mulig for brukere å følge notene mens de lytter til musikken, noe som forbedrer øvingsprosessen.

4. Skrive notater på noter:

- **Beskrivelse:** Brukere skal kunne skrive personlige notater på notene. Dirigenten skal også kunne skrive generelle kommentarer til alle brukere
- **Begrunnelse:** Dette gir en personlig tilpassing og instruksjoner som kan forbedre individuell øving

5. Spore progresjon på sanger:

- **Beskrivelse:** Applikasjonen skal ha funksjonalitet for å spore individuell progresjon på ulike sanger
- **Begrunnelse:** Dette hjelper med å overvåke og forbedre øvingsfremgangen

6. Lage spillelister og setlister

- **Beskrivelse:** Brukere skal kunne lage spillelister og setlister for konserter og øvinger
- **Begrunnelse:** Dette forenkler organiseringen av repertoar og øvinger.

7. Opprette arrangementer i kalenderen

- **Beskrivelse:** En integrert kalenderfunksjon for å opprette og administrere arrangementer

- **Begrunnelse:** Dette letter planleggingen og koordineringen av koraktiviteter

8. Administratorroller med større tilgang og kontroll

- **Beskrivelse:** Det skal være ulike brukernivåer med spesifikke rettigheter.
- **Begrunnelse:** Dette sikrer bedre kontroll over øvingsmaterialet som er tilgjengelig for koret, og hvem som har tilgang til å styre dette.

9. Publisere kunngjøringer

- **Beskrivelse:** Applikasjonen skal tillate administratorer å publisere kunngjøringer til alle medlemmer
- **Begrunnelse:** Dette sikrer effektiv kommunikasjon om viktige oppdateringer eller meldinger

10. Delta på arrangementer

- **Beskrivelse:** Medlemmer skal kunne indikere sin deltakelse på arrangementer og legge til kommentarer om deres deltakelse.
- **Begrunnelse:** Dette gjør det enklere å planlegge og organisere arrangementer basert på medlemmers tilgjengelighet.

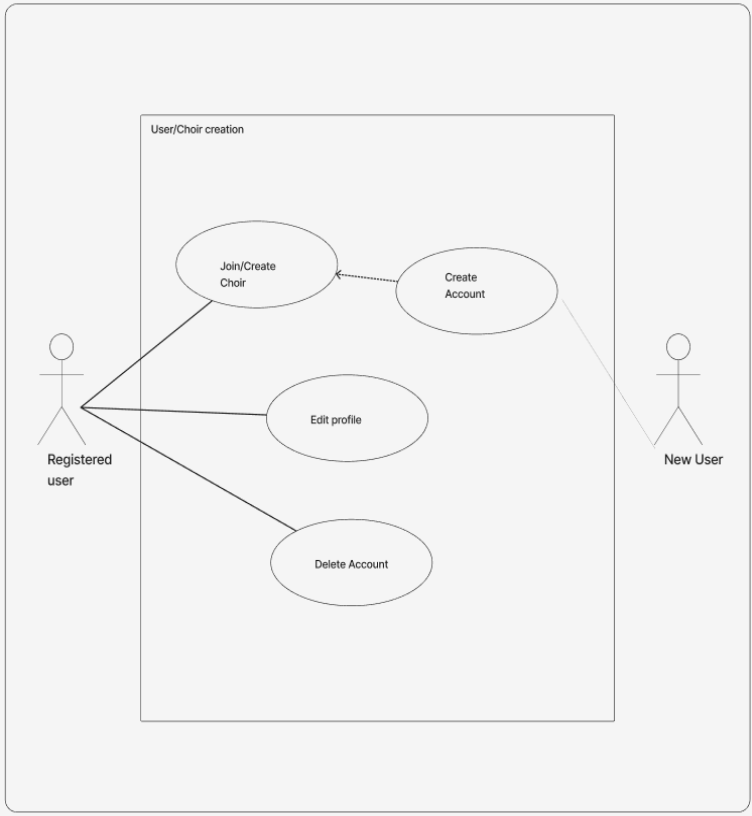
11. Eksportere oppmøtedata

- **Beskrivelse:** Applikasjonen skal ha funksjonalitet for å eksportere oppmøtedata for senere rapportering
- **Begrunnelse:** Dette forenkler administrasjonens evne til å analysere deltakelse over tid, samt rapportering til Norges musikkforbund eller liknende.

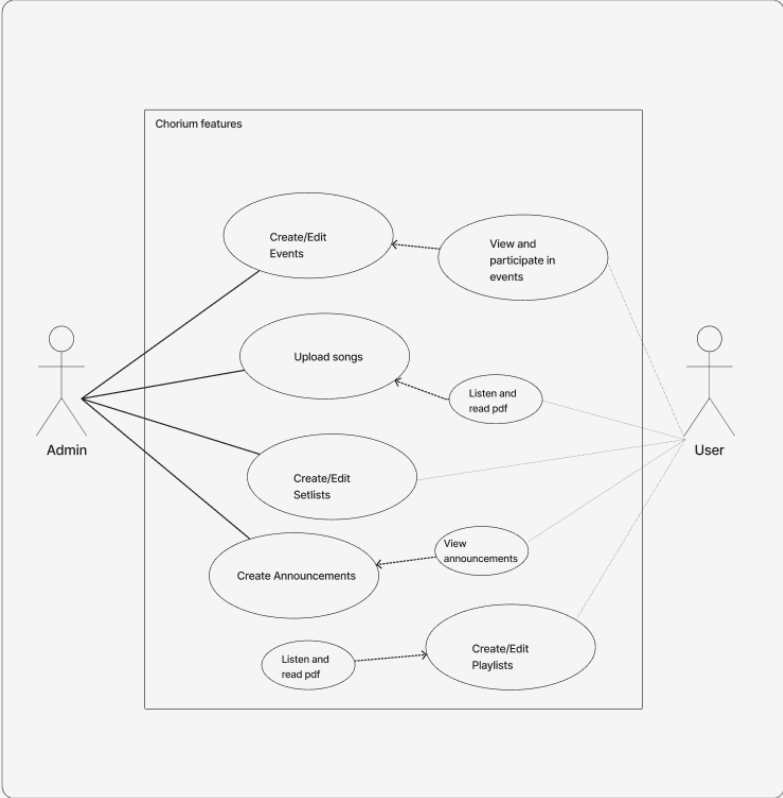
12. Bytte mellom aktive kor

- **Beskrivelse:** Brukere som er medlem av flere kor skal enkelt kunne bytte mellom disse i applikasjonen
- **Begrunnelse:** Dette gjør det lett for brukere å være med i flere kor uten å måtte registrere flere brukere.

Disse brukstilfellediagrammene viser nøkkelfunksjonene til applikasjonen.



Figur 2: Brukstilfellediagram



Figur 3: Brukstilfellediagram

2.2 Avgrensninger

I utviklingen av denne applikasjonen, vil prioriteringen ligge på å implementere de essensielle funksjonene først, på grunn av begrenset tid. Det er rom for å inkludere ytterligere funksjonalitet hvis tidsrammen tillater det. En ytterligere utfordring er utformingen av et tiltalende brukergrensesnitt. Teamet har begrenset erfaring med designprinsipper, selv om det er en grunnleggende forståelse for viktige aspekter som universell utforming. Skapingen av et brukervennlig grensesnitt er kritisk, men på grunn av teamets begrensede erfaring med design, kan det estetiske aspektet bli sekundært. Dette innebærer ikke at teamet ikke vil strebe etter å sikre at applikasjonen ikke bare er funksjonell, men også visuelt akseptabel.

2.3 Ressurser

ChatGPT er en avansert LLM (Large Language Model) utviklet av OpenAI (OpenAI, 2024). Verktøyet er trent opp på tekst hentet fra internett, og kan forstå og generere menneskelig tekst. Dette kan være veldig nyttig i utviklingsprosjekter, både for å diskutere diverse problemstillinger og designvalg, men det åpner også for nye muligheter gjennom verktøy som GitHub Copilot, et verktøy for kodegenerering, som studier viser kan gjøre utviklere 55% mer effektive (Kalliamvakou, 2022). Forståelsen av slike verktøy, og evnen til å effektivt anvende slike teknologier i utviklingsprosessen, er essensielt om man ønsker å henge med i fremtidens digitale landskap (Kalliamvakou, 2024).

3. DESIGN AV PROSJEKTET

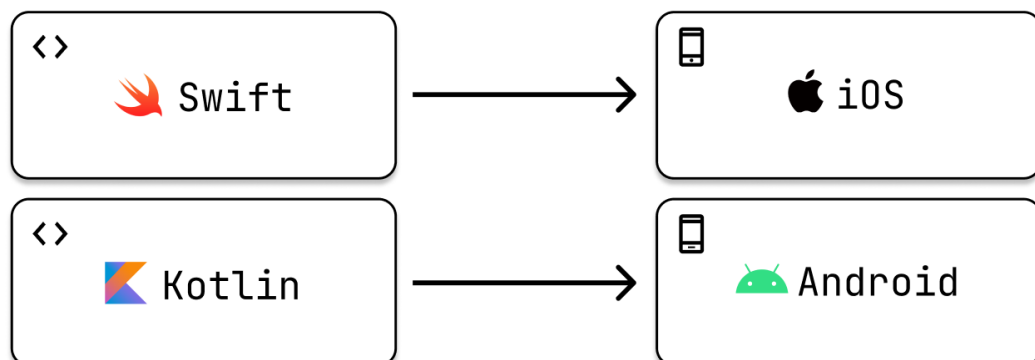
I dette kapittelet vil leseren få en klar forståelse av hvilke teknologiske og metodiske tilnærminger som er valgt for prosjektet, hvorfor disse valgene ble tatt, og hvordan de bidrar til å forme veien videre for prosjektet. Dette inkluderer teknologier og programmeringsspråk, utviklingsrammeverk, databaseløsninger og andre verktøy, samt metoder for evaluering av det ferdige produktet.

3.1 Forslag til løsning

Når det gjelder utvikling av applikasjoner for mobile enheter er det et par forskjellige tilnærminger man kan gå for, hver med sine fordeler og begrensninger, avhengig av prosjektets rammer.

3.1.1 Native

Native utvikling er den første tilnærmingen, hvor applikasjoner skreddersys for iOS og Android operativsystemene ved å benytte deres respektive programmeringsspråk, Swift for iOS og Java eller Kotlin for Android (IBM, u.å.). Dette fører til at utviklere må bygge separate applikasjoner for hver plattform, eller eventuelt utvikle for en, og prøve å konvertere over til den andre. Dette kan være en mer krevende prosess, men man belønnes med applikasjoner som er optimalisert for hver plattform, noe som generelt fører til bedre ytelse og mer sømløs brukeropplevelse (Gamble, 2022).

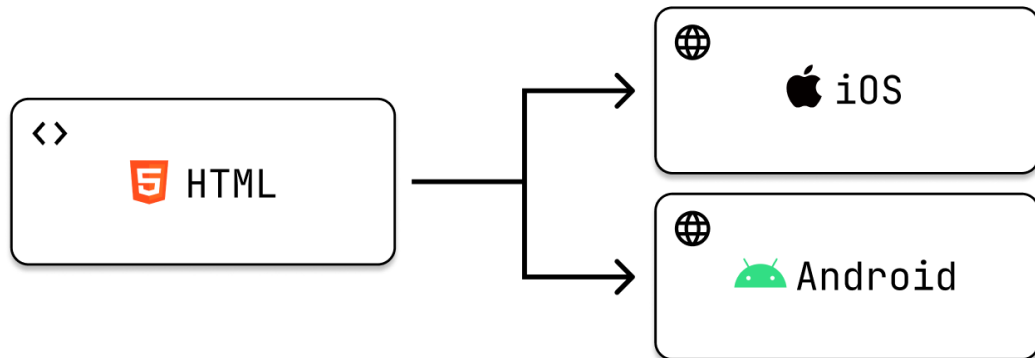


Figur 2: Native utvikling

3.1.2 Web-basert

Web-basert utvikling tilbyr en annen tilnærming, hvor applikasjonen utvikles som en web-applikasjon tilgjengelig via nettleser. Dette unngår behovet for plattformspesifikk utvikling, men setter heller fokuset på responsivt design for å sikre kompatibilitet på tvers av enheter med forskjellige skjermstørrelser. Denne tilnærmingen kan være ideell for

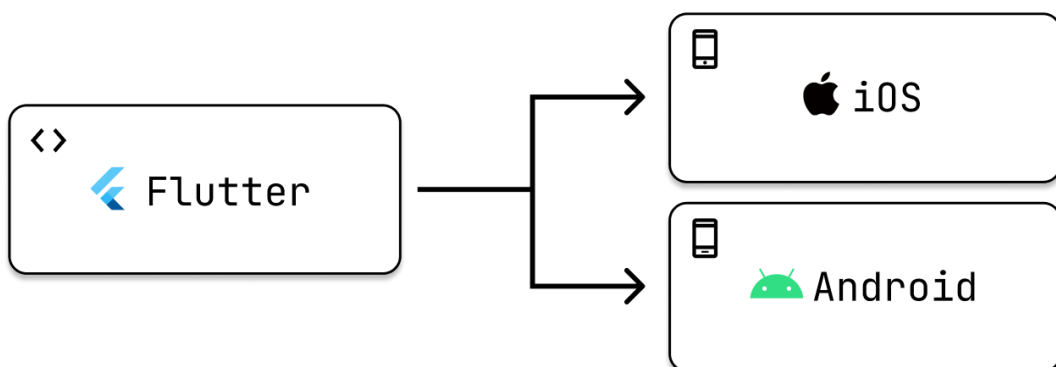
prosjekter med begrensede ressurser eller for utviklere med eksisterende erfaring i webutvikling, da den utnytter kjente teknologier som HTML, CSS og JavaScript (Griffith, u.å.). En negativ side med slike applikasjoner er at man må være koblet til internett for å bruke dem.



Figur 3: Web-basert utvikling

3.1.3 Hybridløsning

En mellomliggende løsning er en hybrid tilnærming, hvor man ved hjelp av bestemte rammeverk kan utvikle en enkelt kodebase som deretter tilpasses til å fungere på flere plattformer. Denne løsningen tar litt positive aspekter fra de to andre tilnærmingene ved å gjøre utviklingsprosessen effektiv, men samtidig tilby brukeropplevelser som ligner mer på native applikasjoner. Rammeverk som Flutter og React Native tilbyr fleksibilitet til å velge et programmeringsspråk basert på utviklerens erfaringer og kompetanse (Dutta, u.å.).



Figur 4: Hybrid utvikling

3.2 Valgt Løsning

For dette prosjektet valgte vi en hybrid tilnærming ved å bruke Flutter-rammeverket. Denne tilnærmingen tillater oss å utvikle en kryssplattform-applikasjon fra én felles

kodebase, noe som betydelig øker effektiviteten i utviklingsprosessen. Dette valget var gunstig gitt prosjektets tidsbegrensninger og nødvendighet for å støtte iOS- og Android-enheter, med potensial for utvidelse til web. En viktig fordel med en hybrid løsning fremfor en webbasert løsning er at man ikke ofrer betydelig med ytelse. Dart-språket, som brukes med Flutter-rammeverket, kompilerer applikasjonen til native kode, noe som sikrer responsiv brukeropplevelse (Obum, 2022). Videre forenkler en enhetlig kodebase prosessen med kontinuerlig utrulling av oppdateringer over flere plattformer.

3.3 Valg av verktøy

Flutter:

Valget av teknologiske verktøy falt naturligvis, grunnet tidligere erfaring og utvikling, på kombinasjonen av Flutter og Firebase. Flutter er et rammeverk utviklet av Google, og tillater utvikling av både iOS- og Android-applikasjoner fra en og samme kodebase som er skrevet i programmeringsspråket Dart, med potensial for utvidelse til web (Obum, 2022). Dette fører til en betydelig effektivitet i utviklingsprosessen, samtidig som det sikrer konsistens på tvers av plattformer.

Firestore:

For back-end falt valget på Firestore, også fra Google, som er en skybasert tjeneste som tilbyr løsninger for databehandling, autentisering, integrasjon av push-varsler, og en rekke andre tjenester som gjør applikasjonsutviklingen lettere. Firestore tilbyr også verdifulle innsikter gjennom sin detaljerte statistikk, som er veldig nyttig for å måle brukerengasjement (Stevenson, 2018).

Git:

For å gjøre samarbeid og versjonshåndtering enkelt, brukes Git i utviklingsprosessen. Git gjør det enkelt å samarbeide effektivt innad i teamet, da det gir ekstra kontroll og sporbarhet av kodebasen, samtidig som det støtter en iterativ arbeidsmetodikk med regelmessige oppdateringer og forbedringer av applikasjonen (Spinellis, 2012). For ytterligere funksjonalitet og oppbevaring av koden i skylagring brukes GitHub. GitHub tilrettelegger for enda bedre samarbeid gjennom funksjoner som "Pull Requests", som gjør det mulig for gruppe medlemmene å diskutere endringer og gjennomgå kode før det rulles ut (Spinellis, 2012)

Utviklingsmiljø:

Utviklingsmiljøet vårt støtter seg på både Android Studio, Xcode og Visual Studio Code. Disse utviklingsmiljøene tilbyr god støtte for Flutter-utvikling, særlig Android Studio, inkludert bruk av emulatorer som man kan kjøre applikasjonen på underveis i utvikling,

noe som gjør feilsøking mye mer effektivt, samt muliggjør testing på mange forskjellige enheter og plattformer uten behov for fysisk maskinvare. Xcode på en maskin som kjører macOS er nødvendig for å bygge til iOS enheter (Flutter, u.å.).

Figma:

Figma, et web-basert designverktøy, ble benyttet for å designe og prototype wireframes for brukergrensesnittet. Dette verktøyet gjør det mulig å arrangere ulike elementer og komponenter for å utforme et sammenhengende brukergrensesnitt. Videre tillater Figma å integrere enkel navigering mellom forskjellige sider i applikasjonen, noe som kan aktiveres gjennom interaksjoner med forskjellige elementer (Perera, 2023). Dette bidrar til å simulere hvordan den endelige applikasjonen vil oppføre seg når brukeren interagerer med den, som gjør det enklere å designe brukergrensesnitt.

Trello:

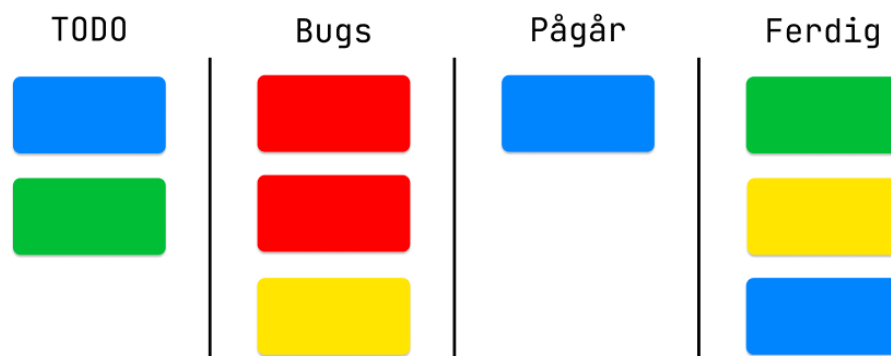
Trello er et samarbeidsverktøy som lar deg organisere arbeidsoppgaver med en KanBan tavle (Trello, u.å.). På denne tavlen kan man opprette "Klistrelapper", og plassere de under egendefinerte kategorier, som for eksempel "TODO", "Pågår" og "Ferdig"

3.4 Prosjektmetodikk

I dette delkapittelet vil vi se på metodikk som skal anvendes for prosjektet. Dette innebærer utviklingsmetodikk, prosjektplaner og evalueringsmetodikk.

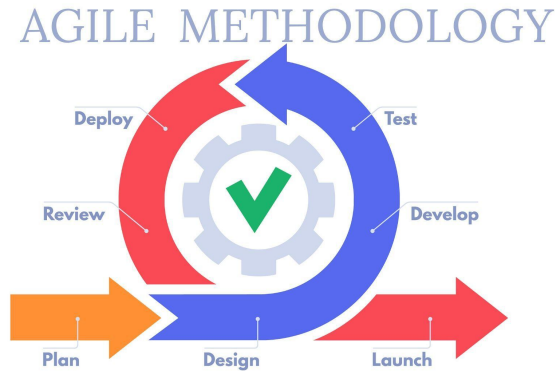
3.4.1 Utviklingsmetodikk

I dette prosjektet brukes en fleksibel smidig utviklingsmetodikk, sentrert rundt et KanBan-brett for oversikt og fordeling av arbeidsoppgaver. Ved å organisere problemer og ønsket funksjonalitet på et KanBan-brett, bidrar det til en god og effektiv arbeidsflyt som tillater kontinuerlige justeringer og prioriteringer basert på prosjektets behov og diverse utfordringer som kan oppstå (Rehkopf, u.å.).



Figur 5: Eksempel på KanBan brett

Utviklingsmetodikken er iterativ, noe som betyr at applikasjonen utvikles og gis ut i flere omganger. Dette tilrettelegger for kontinuerlig evalueringer og forbedringer av funksjoner basert på tilbakemeldinger fra brukere. Hver iterasjon er en mulighet for å justere og optimalisere produktet, øke funksjonalitet og forbedre brukervennligheten.



Figur 6: Smidig metodikk (Davies, u.å.)

Med hensyn til prosjektets tidsbegrensninger, planlegges det for to utgivelser av applikasjonen til brukerne. Den første utgivelsen, en tidlig versjon av produktet, lanseres tidlig i utviklingsfasen. Dette valget muliggjør innsamling av verdifulle tilbakemeldinger fra brukertestere, som blir avgjørende for videre utvikling og forbedring.

3.4.2 Prosjektplan

En forenklet plan over prosjektets oppgaver er utarbeidet og presentert gjennom et Gantt-diagram, som er inkludert som et vedlegg. Dette diagrammet gir en oversikt over forventede tidsrammer for gjennomføringen av de forskjellige oppgavene og deres varighet. Det er viktig å merke seg at nøyaktigheten av tidsestimatene for enkelte oppgaver kan være utfordrende å fastslå, grunnet mangel på tidligere erfaringer og det faktum at mange oppgaver utføres parallelt. Dette gjelder spesielt for utviklingsarbeidet, hvor utvikling av diverse funksjoner og design ofte skjer samtidig.

Task ID	Task Name	Start Week	End Week	Uke 3	Uke 4	Uke 5	Uke 6	Uke 7	Uke 8	Uke 9	Uke 10	Uke 11	Uke 12	Uke 13	Uke 14	Uke 15	Uke 16	Uke 17	Uke 18	Uke 19	Uke 20	Uke 21	Uke 22	Uke 23	Uke 24
01	Dokumentasjon	Uke 3	Uke 24																						
01.01	Prosjekthåndbok	Uke 3	Uke 24																						
01.02	Visjonsdokument	Uke 3	Uke 10																						
01.03	Kravdokument	Uke 6	Uke 14																						
01.04	Midtveisrapport	Uke 6	Uke 10																						
01.04.01	Midtveispresentasjon	Uke 10	Uke 11																						
01.05	Endelig rapport	Uke 11	Uke 20																						
01.06	EXPO Plakat	Uke 20	Uke 22																						
01.07	Presentasjon	Uke 21	Uke 23																						
02	Utvikling	Uke 3	Uke 19																						
02.01	Gjøre klar iterasjon 1 for brukertesting	Uke 4	Uke 6																						
02.02	Brukeresting runde 1	Uke 7	Uke 10																						
02.02.01	Brukerundersøkelser	Uke 11	Uke 11																						
02.03	Iterasjon 2	Uke 9	Uke 15																						
02.03.01	Web-app	Uke 9	Uke 15																						
02.03.01	Fixe bugs	Uke 9	Uke 15																						
02.03.01	GUI redesign	Uke 9	Uke 15																						
02.04	Brukeresting runde 2	Uke 15	Uke 18																						
02.04.01	Brukerundersøkelser	Uke 18	Uke 19																						

Figur 7: Fremdriftsplan som GANTT-diagram (Vedlegg 1: Prosjekthåndbok)

I løpet av et prosjekt er det flere risikoer som kan ødelegge for den initielle planen. Disse risikoene er viktig å identifisere tidlig slik at man kan håndtere de før det går for langt og eventuelt skaper store problemer for prosjektet.

Dårlig samarbeid

Et dysfunksjonelt samarbeid kan fort sette en stopper for prosjektets fremgang. Risikoen for at samarbeidet kan bli dårlig er ikke null. Selv om risikoen for at samarbeidet skal bryte sammen ikke er eliminert, er teammedlemmene kjent med hverandre og forventer ikke at dette vil utgjøre en utfordring, forutsatt at effektiv kommunikasjon og regelmessige møter opprettholdes. Gitt at dette er et egendefinert prosjekt uten en fast arbeidsplass eller fastsatte arbeidstider, blir god kommunikasjon enda mer kritisk for å motvirke de utfordringene det medfører.

Skjevfordelt arbeidsmengde

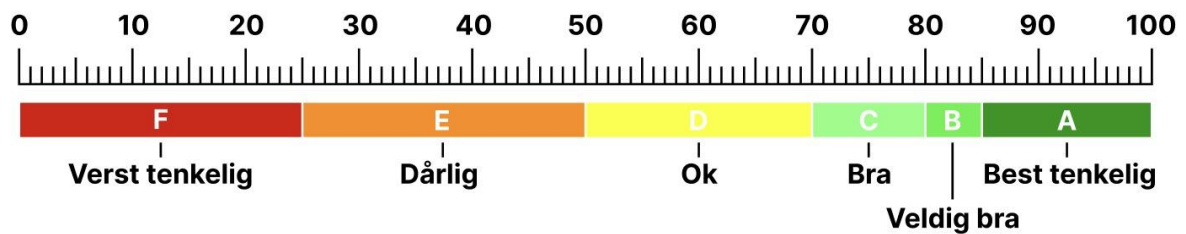
Det er en betydelig fare for at arbeidsfordelingen kan bli ujevnt fordelt. Dette skyldes at prosjektet springer ut av tidligere arbeid fra ett av gruppemedlemmene, som naturligvis gjør denne personen mer kjent med problemstillingen og tidligere anvendte løsninger for lignende funksjonalitet. Det er derfor essensielt å bruke tid på opplæring i begynnelsen av prosjektet. Dette vil bidra til å jevne ut kunnskapsnivået blant deltakerne. Videre er det viktig å etablere god kommunikasjon hvor alle spørsmål kan stilles fritt. Det er også viktig å utnytte de ulike styrkene til gruppemedlemmene, og fordele arbeidsmengden i tråd med dette.

3.5 Evalueringsplan

Prosjektresultatet evalueres ved flere aspekter. Ett aspekt er evaluering av selve koden. Koden kan evalueres ved å sette opp funksjonelle tester som sjekker at funksjoner gjør som de skal. Man kan også evaluere kode ved å gjøre tester av kodekvalitet og sjekke koden opp mot rammeverk for beste praksiser. Dette innebærer lesbarhet, kvalitet på kommentarer og vedlikeholdbarhet av koden. Man ønsker også å evaluere kodens sikkerhet og sjekke at applikasjonen er i tråd med sikkerhetsstandarder for å unngå angrep mot for eksempel server og database (Codacy, 2023). En annen viktig måte man evaluerer resultatet på er å sjekke om applikasjonen møter forhåndsdefinerte funksjonelle og ikke-funksjonelle krav.

Det viktigste aspektet ved evalueringen er å analysere tilbakemeldinger fra brukere i målgruppen. Her vil man bruke systematiske brukerundersøkelser sendt ut til brukertestere, og deretter kalle inn noen brukere til en kvalitativ intervjurunde. Her vil man finne ut hvordan produktet måles opp mot målgruppens tidligere løsninger.

Gruppen ønsker å ha to runder med brukertesting for å kunne sette de to opp mot hverandre og evaluere om senere iterasjoner er en forbedring fra tidligere iterasjoner. Brukeropplevelse er viktig å evaluere, og dette kan, i tillegg til tilbakemeldinger fra brukere, måles opp mot krav om universell utforming. For å måle brukbarhet skal undersøkelsene følge System Usability Scale, som er et format brukt til evaluering av brukervennlighet (Chinn, 2022), i tillegg til noen spørsmål angående øving og organisering av koret. System Usability Scale lar brukeren rangere påstander på en Likert-skala, som er en femdelt skala hvor brukeren rangerer en påstand fra 1 til 5 (Malt og Grønmo, 2020), og deretter kalkuleres det en poengsum fra 0-100.



Figur 8: SUS poengsum-skala

4. DETALJERT LØSNING

Kapittel 4 vil ta for seg detaljerte beskrivelser av hvordan applikasjonen er bygd opp. Dette inkluderer systemarkitektur, brukergrensesnitt og generell dataflyt i systemet. Etter dette kapittelet vil leseren få en god forståelse for hvordan systemet er bygd opp og valgene bak løsningene.

4.1 Systemoversikt

Applikasjonen er strukturert rundt flere kjernekomponenter som til sammen tilbyr en omfattende løsning for koradministrasjon og øvingsassistanse. Systemets arkitektur fremmer en integrert brukeropplevelse gjennom en modulbasert tilnærming.

AudioHandler

Kjernefunksjonaliteten i applikasjonen er AudioHandler, en modul designet for kontinuerlig og ubrutt lydavspilling. Denne modulen er integrert slik at brukerne kan lytte til lydfiler selv mens de navigerer forskjellige seksjoner av applikasjonen. Dette sikrer at brukerne kan engasjere seg i musikkstykker uten avbrytelser, noe som er essensielt for øving og læring.

Spillelister og Setlister

Administrative brukere, som korledere, har muligheten til å opprette settlister for øvelser, mens individuelle brukere kan opprette personlige spillelister. Disse funksjonene er avgjørende for å tilpasse brukeropplevelsen og støtte både kollektive og individuelle behov.

SessionProvider

SessionProvider håndterer autentisering og sesjonsdata, sikrer at brukere får tilgang til relevant innhold basert på deres koraffiliasjon. Dette muliggjør en personlig tilpasset applikasjon hvor brukere enkelt kan veksle mellom ulike kor de er medlem av.

Kalenderintegrasjon

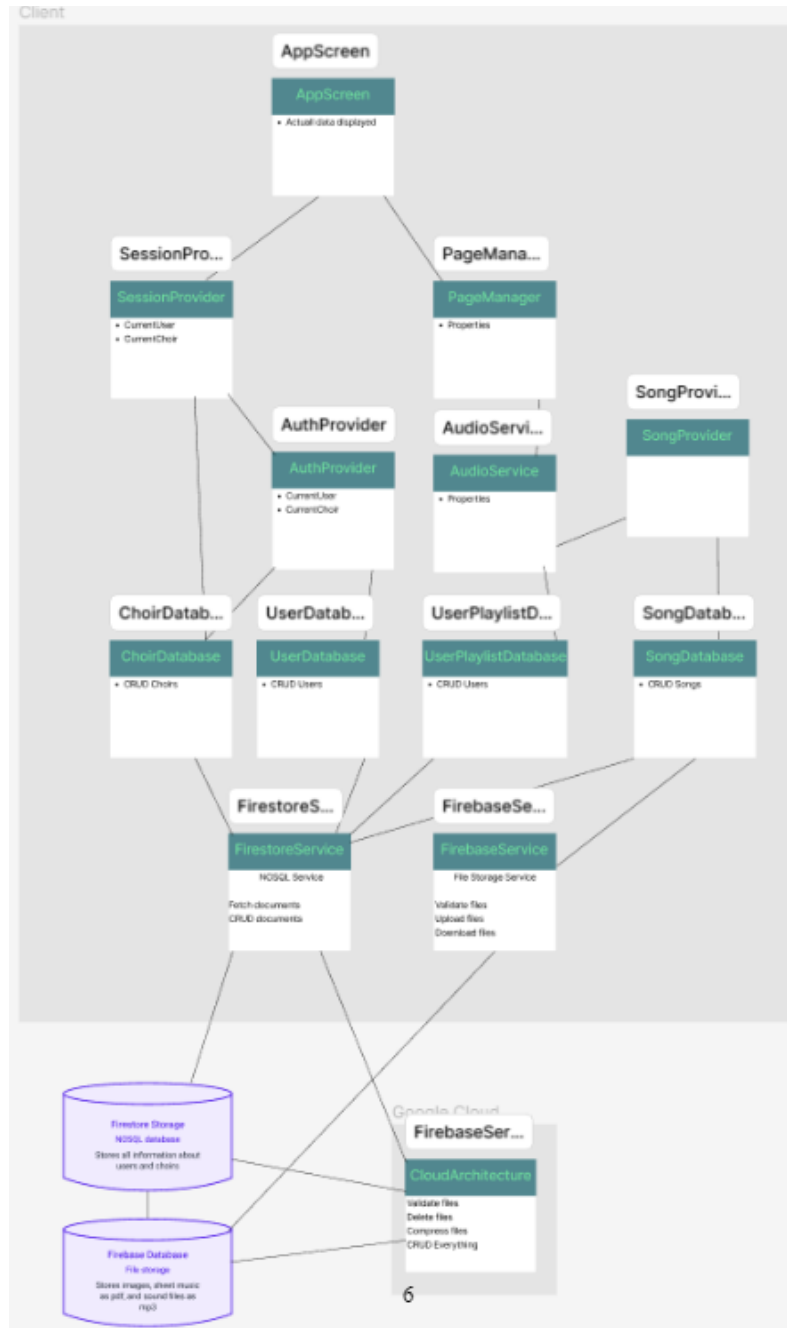
Kalendermodulen benytter Syncfusion sitt bibliotek for å tilby en omfattende og fleksibel kalenderløsning ([“Syncfusion Flutter Calendar,” n.d.](#)). Integrering med Google Maps API ([“Google maps API,” n.d.](#)) og Firebase forbedrer funksjonaliteten ved å tillate detaljert eventhåndtering med tid, sted og notifikasjoner.

Notifikasjoner

For kunngjøringer benyttes Firebase Cloud Messaging (FCM) som støtter både event-baserte og planlagte varslinger. FCMs integrasjon sikrer effektiv distribusjon av viktig

informasjon til kor medlemmer, noe som er vitalt for organisasjonsaspekter av korvirksomheten.

Figur 11 viser overordnet arkitektur for å gi et bilde av alt som skjer fra når dataen blir hentet fra databasen til det blir vist på skjermen.



Figur 9: Overordnet arkitektur fra database til skjerm

4.2 Brukergrensesnitt

For en mobilbasert applikasjon er det viktig med et brukergrensesnitt som er enkelt å forstå for nye brukere og som styrer brukerflyten på en naturlig måte. Flutter rammeverket gjør det enkelt å strukturere moduler ved bruk av forskjellige “Widgets” for forskjellige funksjoner, for eksempel “Padding” som legger til tomrom rundt et objekt, eller “Column” som strukturerer elementer i en kolonne (Mistry, 2023). Ved å strukturere disse på forskjellige måter kan man oppnå et resultat nært det man ønsker.

4.2.1 Viktige designprinsipper

Når man utvikler et brukergrensesnitt for mobile applikasjoner er det flere aspekter å vurdere for å sikre en god brukeropplevelse.

Responsivitet:

Det er viktig at applikasjonen fungerer sømløst på tvers av et bredt spekter av enheter med varierende skjermstørrelser. For å oppnå dette bør man benytte seg av relative måleenheter for elementene i grensesnittet. Dette sikrer at brukergrensesnittet tilpasser seg dynamisk og beholder funksjonalitet og estetikk uavhengig av enheten (Banús, 2023).

Intuitivitet:

Et brukergrensesnitt burde være intuitivt slik at nye brukere raskt kan forstå hvordan de navigerer og interagerer med applikasjonen uten mye forvirring eller behov for opplæring.

Konsistens:

Konsistens i designet er ønskelig for å skape en sømløs brukeropplevelse. Dette betyr at visuelle elementer, typografi, fargevalg, og navigasjon skal være enhetlig gjennom hele applikasjonen. Konsistens hjelper med å styrke merkevare og gjør det enklere for brukerne å forstå og huske hvordan appen fungerer på tvers av moduler.

Estetikk:

Et godt visuelt uttrykk i applikasjonen spiller en merkverdig stor rolle i hvordan den oppfattes av brukerne. Et tiltalende design kan ikke bare øke engasjement, men også bidra til positive konnotasjoner og økt tillit til applikasjonen. Det er derfor viktig å investere tid i å utvikle et visuelt design som appellerer til målgruppen, men som ikke ofrer funksjonalitet (Blavo, 2020).

Universell utforming:

Sist men ikke minst er det viktig å huske på universell utforming når man designer brukergrensesnitt. For at alle skal kunne bruke applikasjonen, er det viktig å tilrettelegge for mennesker med diverse funksjonshemninger (UUTilsynet, u.å.). Noen metoder for å gjøre applikasjoner mer tilgjengelig er for eksempel store kontraster mellom tekst og bakgrunn, og passe på at elementer er merket riktig så eventuelle talefunksjoner på enheter kan lese opp hva de forskjellige elementene på skjermen er.

4.2.2 Designprosess

For design og prototyping av brukergrensesnitt og brukerflyt brukte vi Figma. Figma er et verktøy som lar deg designe brukergrensesnitt og enkelt sette opp en wireframe prototype av applikasjonen ved å legge til navigasjon fra forskjellige elementer på skjermen for å simulere trykk(Perera, 2023).

4.2.3 Brukerflyt

Applikasjonen er delt opp i tre hovedmoduler, “Musikk”, “Dashboard” og “Kunngjøringer”. Disse kan man navigere mellom ved hjelp av en navigasjonslinje på bunn av siden.

Musikk:

Denne modulen er hjertet av det musikalske innholdet i koret, og er hvor brukere finner alle sanger, spillelister og setlister for konserter. Brukerne kan enkelt trykke seg inn på individuelle sanger for å vise notene og spille av sangen, noe som er nyttig for øving før konserter. Ved avspilling vil man få opp en liten avspiller som vises mens man kan navigere rundt på andre sider i bakgrunnen. For å øve på teksten kan man stille på vanskelighetsgrad, som vil fjerne deler av teksten for å teste hukommelsen. Om man føler man kan sangen, kan man markere den som “Kan”, eller så kan man markere den “Må kunne”, eller “Kan ikke” for å tydeliggjøre hva man må øve mer på.

Dashboard:

Fungerer som hovedside, og tilbyr en oversikt over kommende arrangementer, liste over kormedlemmer, og setlister for kommende konserter. Fra denne modulen kan brukere navigere til kalender for utvidet visning av arrangementer og muligheten til å opprette nye arrangementer, en funksjon som er sentral for planlegging og organisering.

Kunngjøringer:

Viser en strøm av kunngjøringer og beskjeder publisert av kormedlemmer. Denne modulen er designet for å lette kommunikasjonen innad i koret og tillater brukerne til å publisere viktig informasjon effektivt. Denne strømmen viser kunngjøringer sortert fra nylig på toppen, til eldre nedover.

Tilleggsfunksjoner:

En hamburgermeny i øvre venstre hjørne gir tilgang til flere sider som brukerprofil, innstillinger, og admin-side. Denne menyen tilbyr også muligheten for å bytte aktivt kor og logge ut, noe som er integrert for å støtte brukere som kanskje er involvert i flere kor.

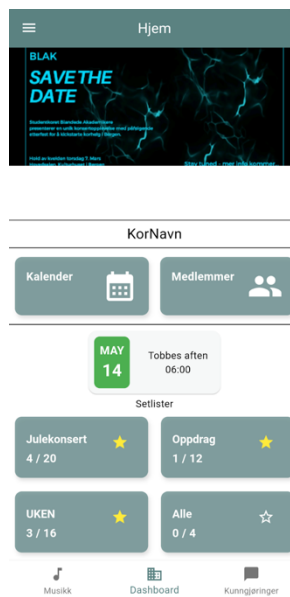
4.2.4 Iterasjon 1

Første iterasjon av Chorium var preget av størst fokus på grunnleggende funksjonalitet, med brukergrensesnitt som en baktanke. Det var ønskelig å ha flere temaer tilgjengelig for applikasjonen, dette innebærer bakgrunnsfarger og primærfarger. Tanken bak dette er å gi brukerne mulighet til å bruke farger som mer ligner på fargene som koret bruker i deres eget design, og også muligheten til å velge mellom lys og mørkt design.

Her kan vi se hvordan de tre hovedmodulene var designet i iterasjon 1, musikk-modulen (Figur 13), dashboard-modulen (Figur 13), og kunngjøring-modulen (Figur 14):



Figur 12: Musikk-modul iterasjon 1

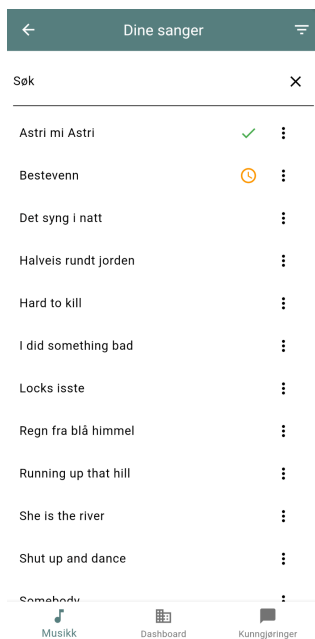


Figur 10: Dashboard-modul iterasjon 1

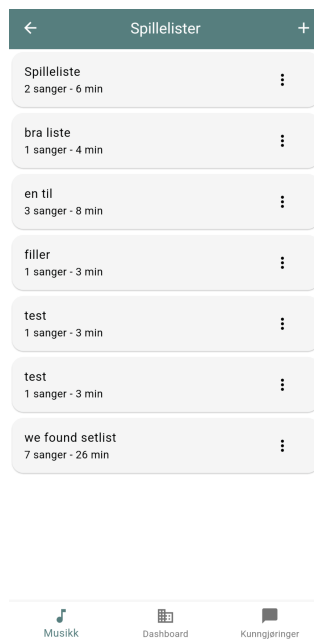


Figur 14: Kunngjøring-modul iterasjon 1

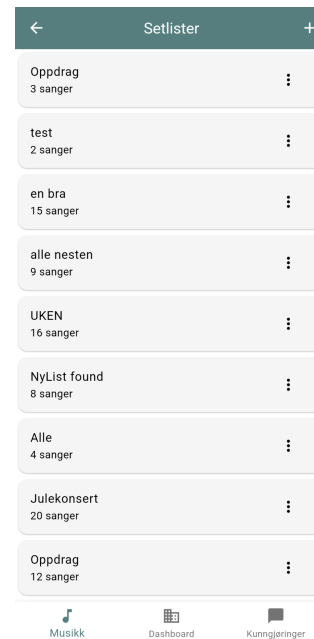
Herfra kan man gå videre til andre sider som gir mer funksjonalitet, for eksempel "Sanger" (Figur 15), "Spillelister" (Figur 16) og "Setlister" (Figur 17) som lister opp respektive innhold:



Figur 11: Sang-side iterasjon 1

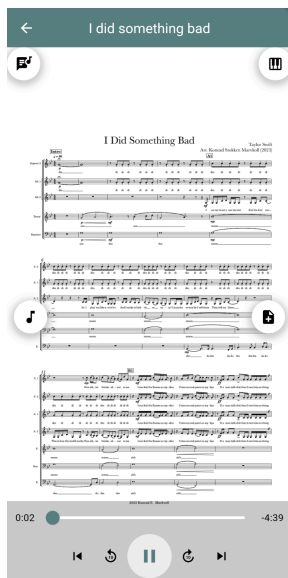


Figur 12: Spilleliste-side iterasjon 1

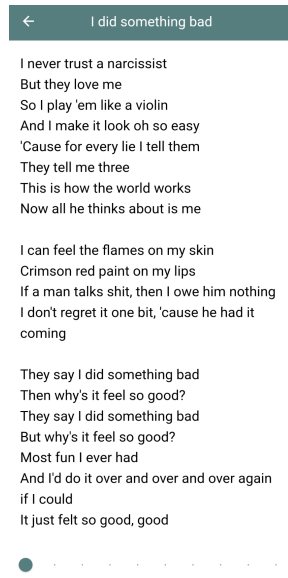


Figur 13: Setlist-side iterasjon 1

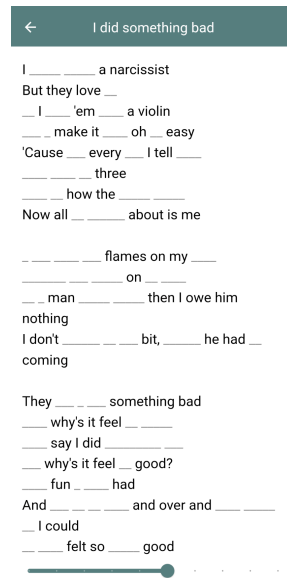
Den viktigste delen av applikasjonen er funksjonaliteten som effektiviserer øving. Her kan man åpne sanger og lese noter samtidig som man kan spille av musikken. Her kan man også få tilgang på tekst, notater, både personlige og fra dirigent, samt et enkelt keyboard.



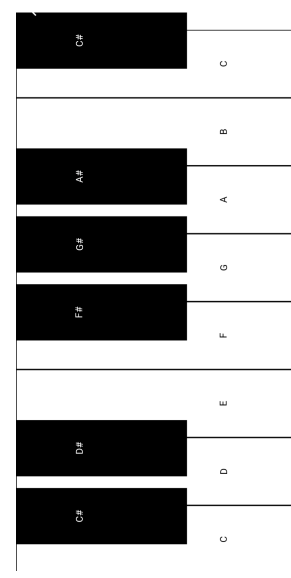
Figur 18: PDF-side



Figur 19: Tekst-side

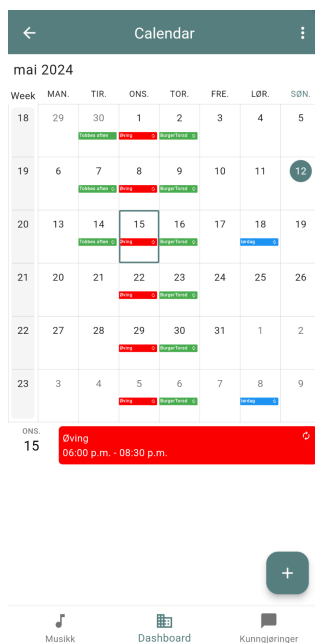


Figur 20: Tekst-side med justert vanskelighetsgrad

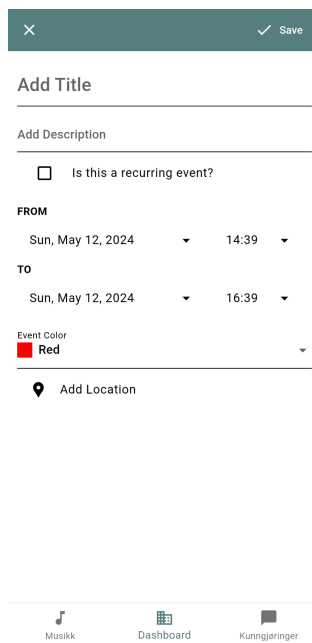


Figur 21: Piano-side

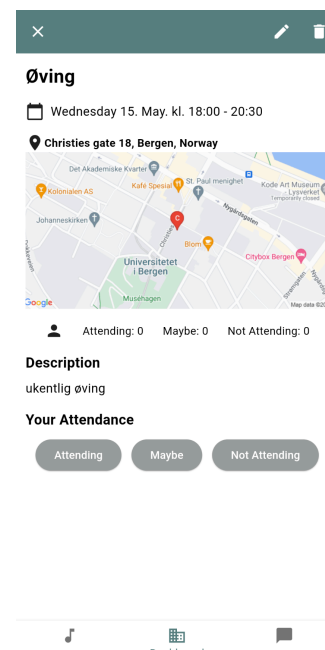
Og via dashboard får man tilgang på kalender og arrangementer:



Figur 22: Kalender-side



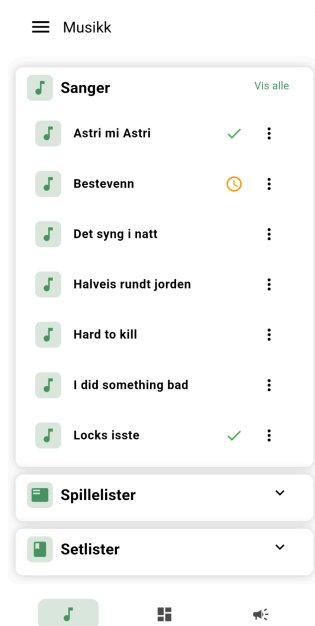
Figur 14: Opprett arrangement



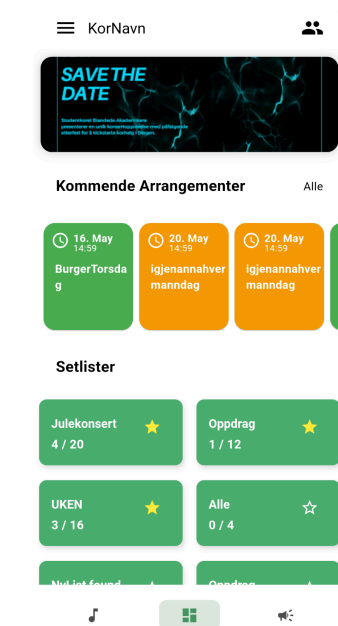
Figur 24: Arrangement-info

4.2.5 Iterasjon 2

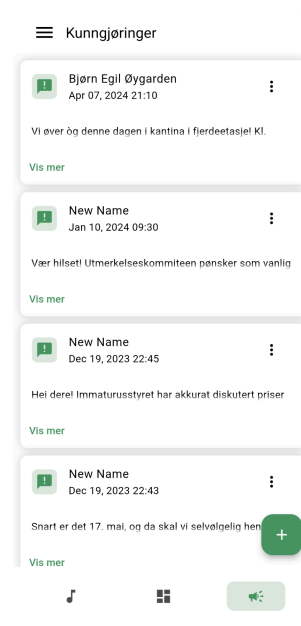
For iterasjon 2 ønsket vi å forbedre det visuelle uttrykket og samtidig sikre et enhetlig design på tvers av alle sidene. Målet var å skape et mer sammenhengende visuelt språk, som forhåpentligvis forbedrer den generelle brukeropplevelsen. Ved å implementere et likt visuelt språk på tvers av sider, sikrer man at brukerne opplever en intuitiv og konsistent navigasjon.



Figur 25: Musikk-modul iterasjon 2



Figur 15: Dashboard-modul iterasjon 2

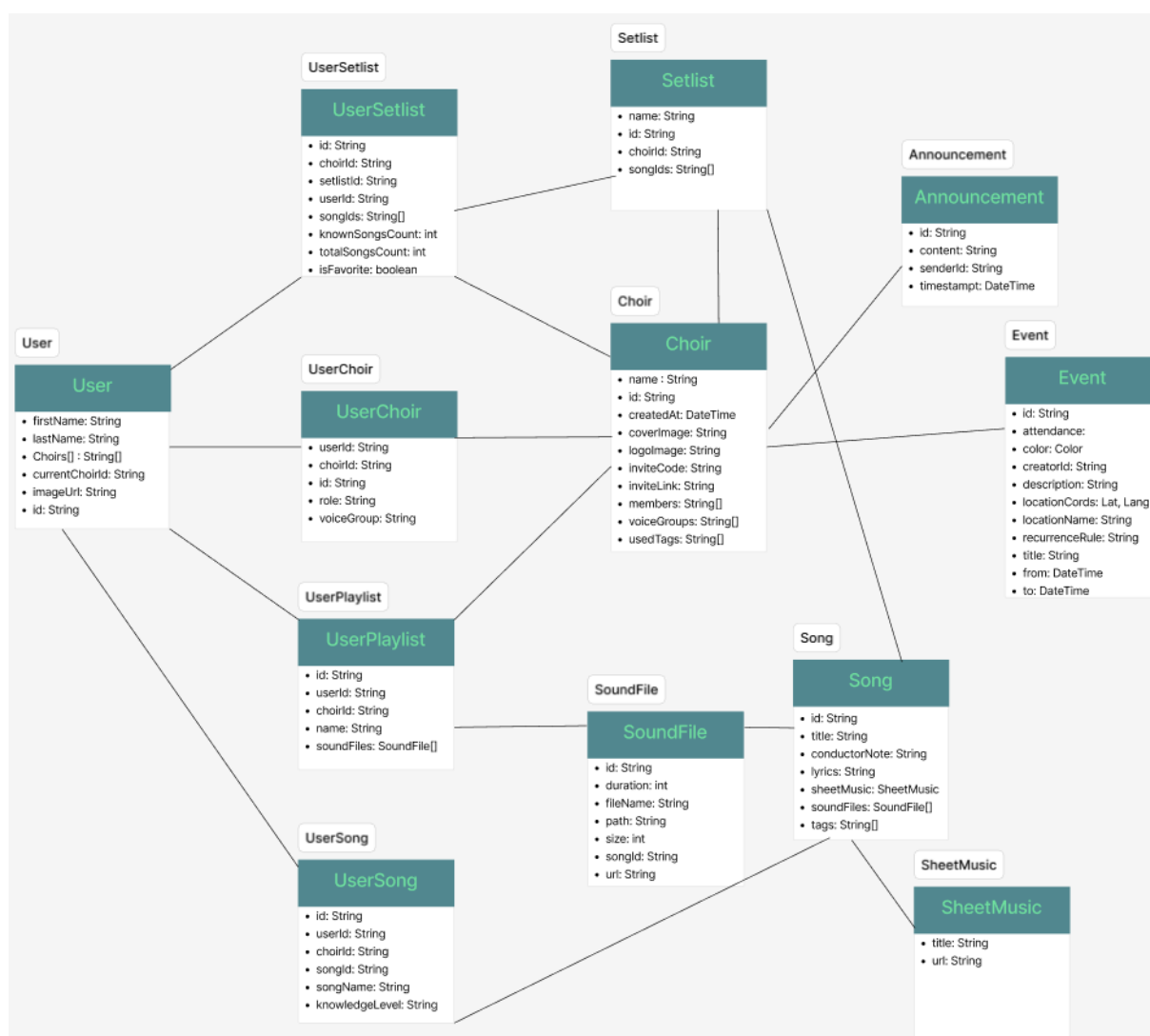


Figur 27: Kunngjøring-modul iterasjon 2

4.3 Systemarkitektur

For en grundig forståelse av systemets arkitektur og de interne relasjonene mellom klassene, kan et klassesdiagram være nyttig. Dette diagrammet illustrerer systemets struktur og samspillet mellom ulike komponenter, noe som er avgjørende for å forstå dataflyt og modulær interaksjon i applikasjonen.

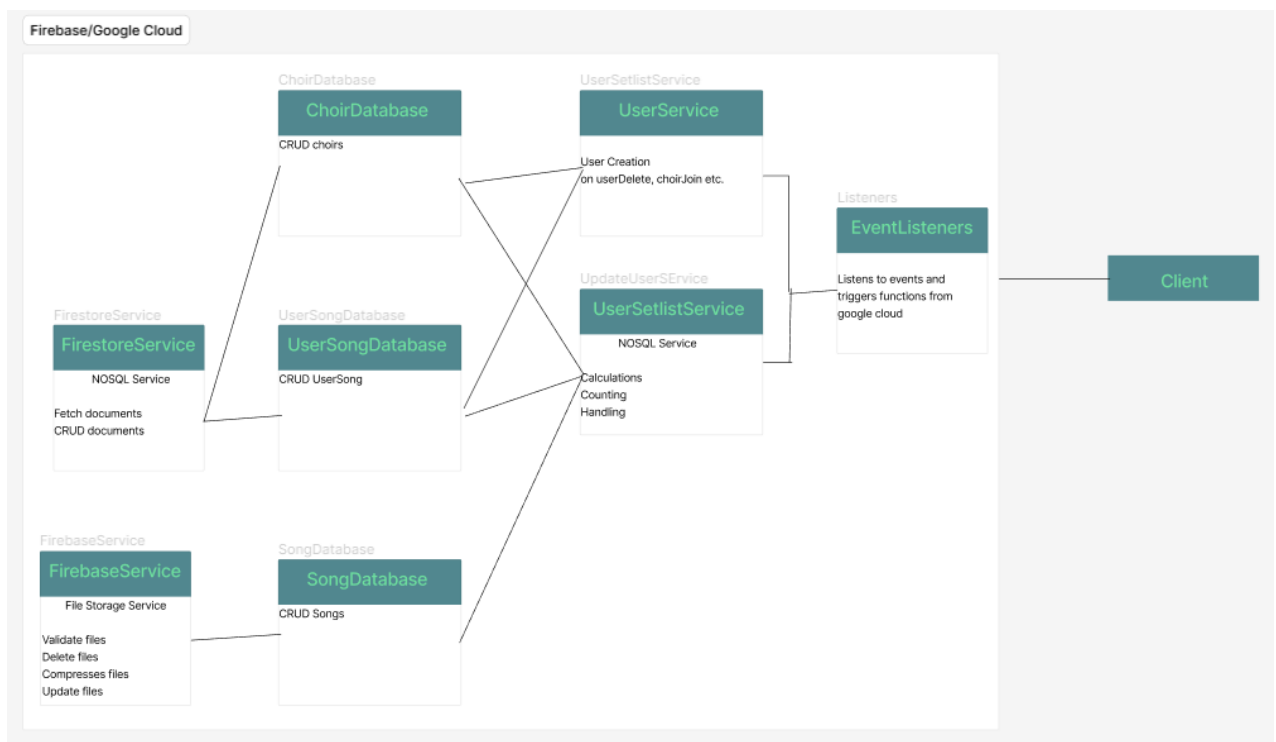
Selv om Firebase ikke tilbyr en tradisjonell relasjonsdatabase, kompenserer vi for dette gjennom bruk av relasjonsklasser som UserPlaylist, UserChoir, og UserSong. Disse klassene muliggjør representasjon av komplekse relasjoner mellom dataelementer ved å benytte unike ID-er for hvert dokument. Dette tilrettelegger for en dynamisk brukeropplevelse der fremgang og personalisering kan spores og tilpasses individuelt.



Figur 16: Databasearkitekturen i Firebase

Operasjoner som krever høy integritet, som opprettelse, sletting og oppdatering av dokumenter, må håndteres med særlig omsorg. For eksempel, når et nytt medlem blir en del av koret, etableres det automatisk relasjoner til hver sang og setliste. Tilsvarende, ved sletting av et kor, skal tilhørende lydfiler, sangdokumenter og deres relasjoner fjernes for å opprettholde datakonsistens. Medlemmene skal også oppdateres slik at de ikke lenger er tilknyttet det slettede koret.

Integrasjonen mellom Firebase og Google Cloud spiller en kritisk rolle i å sikre at disse operasjonene utføres sikkert. For å styrke sikkerheten ytterligere, benyttes Firebase Security Rules. (["Firebase Security Rules," 2024](#)) Disse reglene muliggjør detaljert konfigurering av tilgangskontroll og validering på servernivå, som sikrer at kun autoriserte operasjoner utføres på kritiske data. Ved å definere spesifikke regler for hvem som kan lese, skrive, oppdatere eller slette informasjon i databasen, forsterker vi systemets robusthet mot uautorisert tilgang og datalekkasjer.



Figur 17: Forenklet backend arkitektur som kjører i Google Cloud

4.4 Deployment

Distribusjon:

Applikasjonen distribueres gjennom anerkjente plattformer som Appstore og Google Play Store. Disse plattformene tilbyr robuste metoder for å sikre at applikasjonen gjennomgår

nøye testing før lansering. Etter intern testing, benyttes fasilitetene til Apple TestFlight og Google Play intern testing for å distribuere applikasjonen til en utvalgt gruppe av testbrukere. Dette tillater samling av verdifull brukertilbakemelding og identifisering av eventuelle problemer før den offisielle lanseringen.

Review Prosessen:

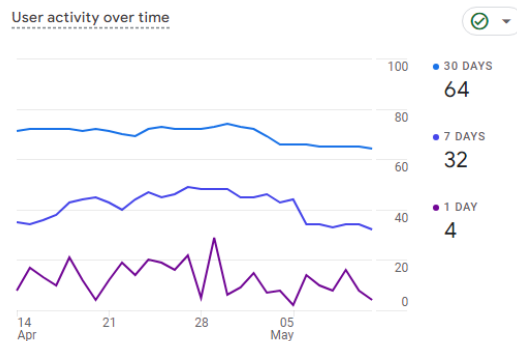
Etter testperioden blir applikasjonen sendt til review hos Apple og Google. Denne prosessen inkluderer en evaluering av om applikasjonen overholder de relevante standardene for funksjonalitet, brukervennlighet og sikkerhet, samt kontroller for å forsikre at applikasjonen ikke inneholder urettmessig innhold. ([“Review guidance for your first app launch,” 2024](#)) Godkjenningsprosessen er nødvendig for hver ny versjon av appen, og sikrer at alle oppdateringer opprettholder de høye standardene satt av distribusjonsplattformene.

Backend Oppdateringer:

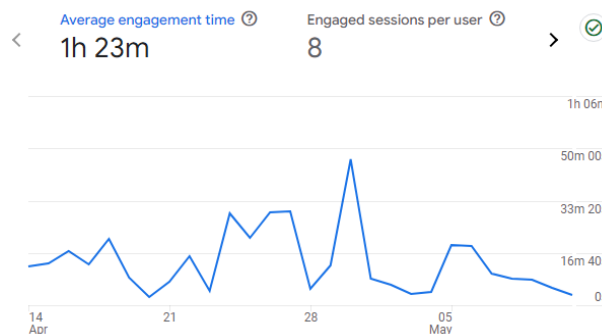
I motsetning til distribusjonen av klientapplikasjonen, som krever grundige godkjenningsprosesser via App Store og Google Play Store, tillater hosting på Google Cloud Platform for en mer fleksibel oppdateringsprosess av backend-systemet. Dette gir utviklerteamet mulighet til å implementere oppdateringer og forbedringer i sanntid. Selv om denne tilnærmingen øker agiliteten i utviklingsprosessen, er det også en forsterket risiko for feil hvis ikke omfattende testing og kvalitetssikring er gjennomført. Derfor implementeres strenge testprosedyrer og overvåkning for å sikre at enhver oppdatering opprettholder systemets integritet og driftsstabilitet.

Overvåkning og Vedlikehold:

For effektiv overvåkning av applikasjonens ytelse og stabilitet, benytter vi Firebase sin suite av verktøy, inkludert Firebase Crashlytics og Analytics. ([“Firebase Crashlytics,” 2024](#); [“Google Analytics,” 2024](#)) Disse verktøyene gir detaljert innsikt i applikasjonens operasjonelle helse, identifiserer og dokumenterer feil, og gir analyser av brukeratferd. Denne informasjonen er kritisk for kontinuerlig forbedring av applikasjonen, og hjelper utviklingsteamet med å raskt adressere og korrigere feil, samt å forstå og tilpasse brukernes behov og oppførsel.



Figur 30: Google Analytics viser brukeraktivitet over tid



Figur 31: Brukerengasjement



Figur 182: Firebase Crashlytics gir viktig innsikt i brukers opplevelser i appen

Issues	Versions	Events ↓	Users
Repetitive crashes FlutterCrashlytics.recordFlutterFatalError io.flutter.plugins.firebase.crashlytics.FlutterError - A RenderFlex overflowed by 117 pixels on the right... Error thrown during la...	1.0.0 - 1.0.0	74	2
Early crashes Repetitive crashes RenderBox.size io.flutter.plugins.firebase.crashlytics.FlutterError - RenderBox was not laid out: _RenderListTile#c22df layoutBoundary=up1... ↳ 4 variants	1.0.0 - 1.0.0	10	1
Repetitive crashes _SoundFileSelectorState.build.<fn> io.flutter.plugins.firebase.crashlytics.FlutterError - setState() called after dispose(): _SoundFileSelectorState#efec6(lifecycle ...	1.0.0 - 1.0.0	6	4

Figur 33: Vi får også detaljert informasjon og stacktrace på hvert enkelt problem så vi vet hvor problemet oppstår

5. RESULTATER

I løpet av dette kapittelet vil vi detaljert se på hvilke metoder som er benyttet under evalueringen av prosjektets kvalitet og relevans. Disse metodene vil gi et svar på om prosjektet oppfyller tidligere definerte målsettinger, og er grunnlag for diskusjon om hvorvidt prosjektet var suksess. I hovedsak vil prosjektet bli evaluert basert på verdifulle tilbakemeldinger fra brukerundersøkelser.

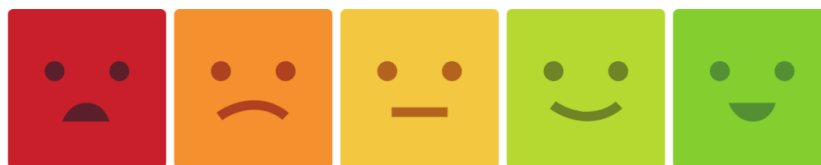
5.1 Evalueringsmetode

5.1.1 Brukertestning

Chorium er en applikasjon designet spesifikt for organisering av koraktiviteter, det var derfor avgjørende for prosjektet å opprettholde en nær tilknytning til kormiljøet, slik at applikasjonen kunne bli testet og evaluert i reelle situasjoner. Grunnet Andreas sin tilknytning til kormiljøet, var det lett å etablere samarbeid med kororganisasjoner som var villige til å bruke Chorium i deres daglige praksis. Dette samarbeidet ga verdifull innsikt i hvordan det var å bruke applikasjonen i praksis, både med hensyn til brukeropplevelsen og hvordan den påvirker korets øvingskultur.

5.1.2 Design av brukerundersøkelse

For å evaluere brukeropplevelsen og applikasjonens effekt på øvingskulturen, ble det laget en todelt brukerundersøkelse som ble sendt ut til testbrukerne. Den første delen av undersøkelsen brukte System Usability Scale (SUS), en anerkjent metode for å måle brukervennligheten til systemer. Denne delen består av ti spørsmål formulert på en måte som reflekterer både positive og negative aspekter ved brukeropplevelsen, og er arrangert med annenhver positiv og negativ påstand (Chinn, 2022). For hvert spørsmål svarer brukeren på en skala fra 1 til 5, hvor 1 betyr helt uenig og 5 betyr helt enig, hvor enig de er i en påstand (Malt og Grønmo, 2020).



Figur 19: Likert-skala illustrasjon (ClearSurvey, u.å.)

Poengsummen for en SUS undersøkelse beregnes ved å regne positive og negative spørsmål hver for seg. For positive påstander vil man ta brukerens svar på skalaen minus 1, og deretter legge sammen resultatet for alle påstandene. For negative påstander vil

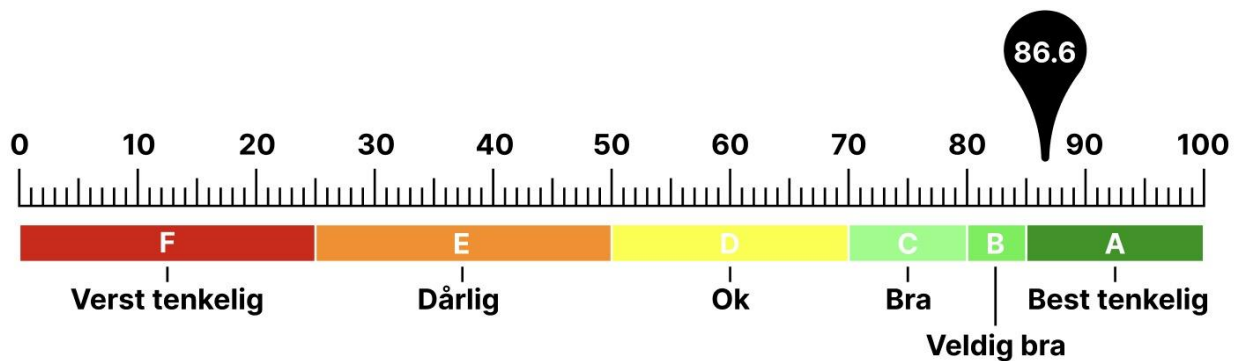
man legge sammen brukerens svar for alle påstandene og deretter subtraherer man dette fra 25. Den totale poengsummen finner man ved å addere positivt og negativt resultat og deretter multiplisere med 2,5. Da får man et resultat mellom 0 og 100 (Chinn, 2022). Tar man gjennomsnittet av resultatene fra alle de individuelle undersøkelsene får man gjennomsnittet for alle undersøkelsene.

Del to av undersøkelsen fokuserte på hvordan applikasjonen påvirket øvingskulturen innad i koret. Denne delen inneholdt spørsmål utformet for å avdekke hvordan Chorium påvirket organisering og kommunikasjon, samt hvordan den påvirket effektivitet og engasjement rundt øving. En fullstendig liste over spørsmålene som ble brukt i brukerundersøkelsen kan bli funnet i vedlegg A.

5.2 Evalueringsresultat

5.2.1 Analyse av resultater fra brukerundersøkelse

Analysene av brukerundersøkelsene utført etter brukertesting av iterasjon 1 indikerer en betydelig positiv reaksjon til applikasjonen. Fra den første delen av undersøkelsen, som benyttet System Usability Scale (SUS), fikk applikasjonen en imponerende poengsum på 86,6 av 100. Dette tilsvarer en bokstavkarakter A, og tyder på høy brukertilfredshet og indikerer at opplevelsen var overveiende positiv.

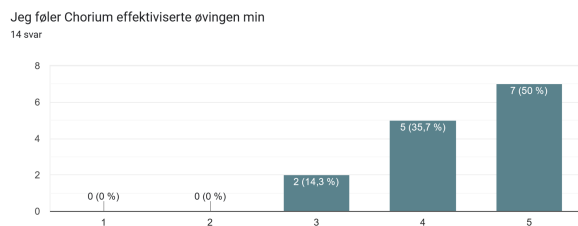


Figur 20: SUS resultat

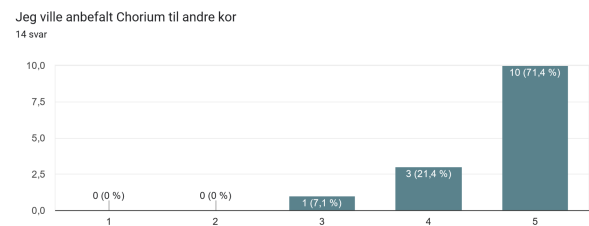
Den andre delen av undersøkelsen, som fokuserte på applikasjonens effekt på øvingsrutiner, viste en forbedring i effektiviteten av øvinger, med et gjennomsnittlig svar på 4.37 på påstanden om Chorium hadde bidratt til mer effektive øvinger. I tillegg rapporterte et flertall av brukere, med en gjennomsnittlig poengsum på 4.64 at de ville anbefalt Chorium til andre kor, noe som understreker applikasjonens potensial og

nytteverdi. Brukerne virket også fornøyd med hvor enkelt det var å finne lydfiler, og kvaliteten på avspillingsfunksjonen.

Til tross for de positive tilbakemeldingene, belyste også undersøkelsen noen områder for forbedring. Spesielt ble applikasjonens kommunikasjonsverktøy, som i iterasjon én består av kunngjøringer, vurdert til kun 2,5 av 5, noe som tyder på at disse verktøyene ikke helt

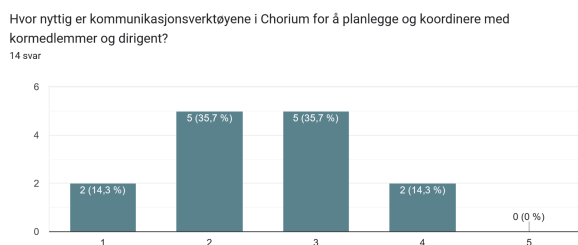


Figur 21: Effekt på øving

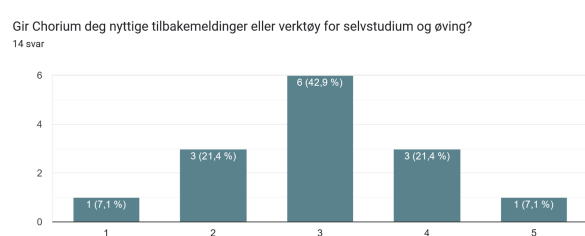


Figur 22: Ville anbefalt til andre kor

møtte brukernes behov eller mangler funksjonalitet for at det skal lønne seg å bruke dette istedenfor kommunikasjon på andre plattformer. En annen påstand som mottok lavere uenighet var om applikasjonen ga nyttige tilbakemeldinger for selvstudium. Dette indikerer et område med potensial for videre utvikling.



Figur 24: Nyttighet av kommunikasjonsverktøy



Figur 23: Nyttighet av tilbakemeldinger på øving

Til sist ga vi mulighet for åpne tilbakemeldinger i en tekstboks, hvor brukerne kunne dele mer spesifikke meninger om applikasjonen. En del brukere påpekte et teknisk problem hvor applikasjonen hadde en tendens til å krasje under spesifikke hendelser knyttet til avspilling av lydfiler. Disse tilbakemeldingene har vært uvurderlige i utviklingen av iterasjon to, da brukerne direkte informerer om problemer som er nødvendig å fikse for en optimal brukeropplevelse. En full oversikt over resultatene fra undersøkelsen kan bli funnet i vedlegg B.

5.2.2 Iterasjon 2

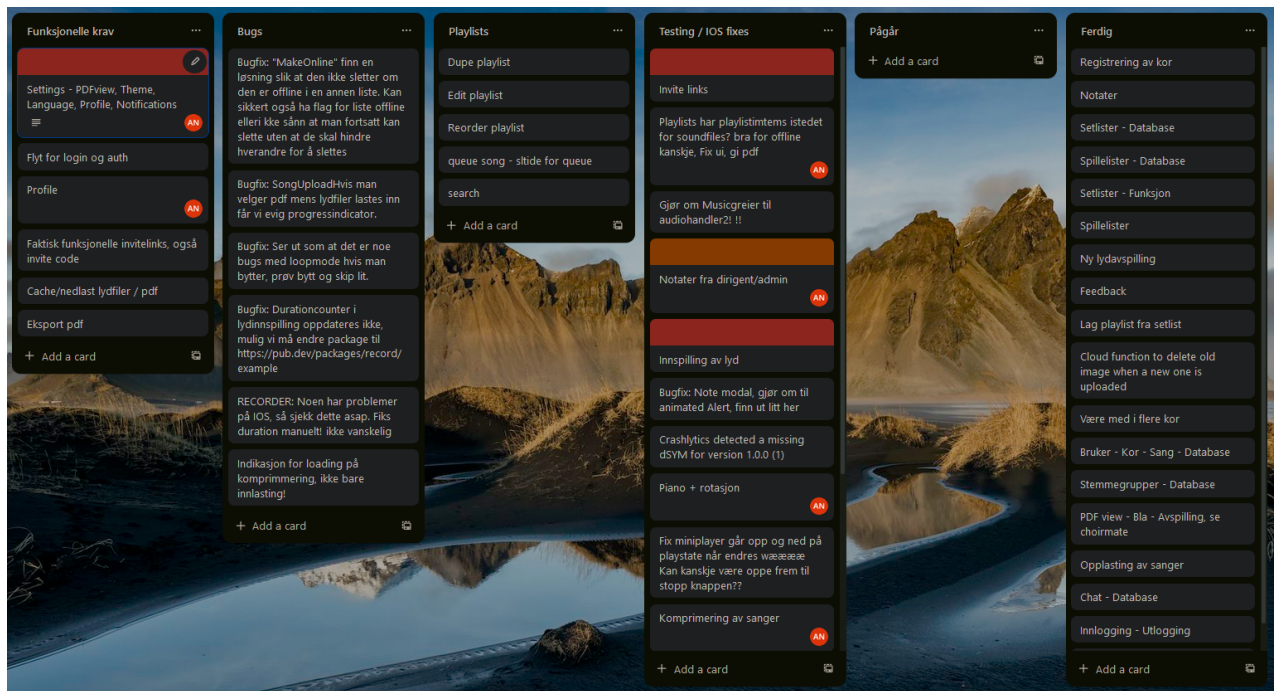
Planen for prosjektet var å inkludere enda en runde med brukertesting for å evaluere forbedringene som kom i iterasjon 2, basert på tilbakemeldingene fra første brukerundersøkelse. Målet var å bygge videre funksjonalitet og se en forbedring i brukertilfredshet ved å implementere endringer foreslått etter første iterasjon.

Dessverre ble den andre runden med brukertesting ikke gjennomført som planlagt grunnet tidsbegrensninger mot slutten av prosjektperioden. I stedet prioriterte vi heller å utvikle en funksjonell og forbedret iterasjon, fremfor å utgi det vi hadde kun for å få en ny runde brukertesting selv om endringene var minimale. Selv om dette hindret oss i å samle direkte sammenlignbare data fra en påfølgende og forbedret iterasjon, har den initielle brukerundersøkelsen gitt verdifulle innsikter i hvordan vi kan forbedre applikasjonen videre.

5.3 Prosjektgjennomføring

I evalueringen av prosjektet, ble funksjonelle tester dessverre nedprioritert. Gitt prosjektets begrensede omfang og ressurser, ble fokus lagt på å utvikle et fungerende brukergrensesnitt, noe som resulterte i at systematisk testing av kode ikke var et stort fokus. Selv om man ideelt sett ville benyttet automatiserte tester, for eksempel gjennom CI/CD verktøy for kontinuerlig kvalitetssjekker, som GitHub Actions, ble det i stedet utført manuelle tester på både emulatorer og fysiske enheter gjennom hele utviklingsprosessen. Flutters innebygde verktøy for logging ble også flittig brukt for feilsøking av applikasjonen. Denne tilnærmingen har både sine styrker og svakheter, hvor hovedfordelen er en fleksibel og rask prototyping, mens den åpenbare risikoen er potensialet for oversette feil som kan påvirke applikasjonens stabilitet og kvalitet.

Som utviklingsmetodikk ble det brukt en smidig tilnærming, med problemer delt opp og på et KanBan Brett. En slik tilnærming fungerer bra for en mindre utviklingsgruppe hvor arbeidsflyt er mer dynamisk. Det var også en meget passende metodikk for et prosjekt som utvikler iterativt, hvor funksjonalitet bygges på litt og litt, og gis ut i mindre iterasjoner, fremfor én stor utgivelse av det ferdige produktet. Iterative utgivelser gjorde også at prosjektet kunne ta i mot gode tilbakemeldinger underveis, for å så implementere funksjonalitet basert på tilbakemeldinger.



Figur 25: Skjerm bilde av Trello-brett

6. DISKUSJON

I dette kapittelet analyseres prosjektresultatene i lys av de opprinnelige målene som ble etablert i Kapittel 1. Vi vil reflektere over hvordan Chorium har påvirket effektivitet og organisering av korvirksomhet. Videre vil vi diskutere valgene vi har tatt underveis, fra metodikk til teknologiske valg, og hvordan disse har påvirket prosjektets utfall. Til sist vil vi reflektere over prosjektets bredere implikasjoner og potensial for utvidelse.

6.1 Analyse av prosjektresultater

I kapittel 1 ble det definert noen mål for prosjektet og applikasjonen. Målet var å få på plass den grunnleggende funksjonaliteten i applikasjonen som muliggjorde enklere organisering av et kor. Videre ble det stilt noen forskningsspørsmål rundt korvirksomhet og øving, og hvordan en applikasjon med samlet funksjonalitet kunne påvirke dette. Disse spørsmålene var:

1. Kan en dedikert applikasjon øke effektiviteten for øving og organisering av kor sammenlignet med nåværende metoder?
2. I hvilken grad øker applikasjonen tilgjengeligheten av øvingsmateriale?
3. Hvordan påvirker applikasjonen kormedlemmenes motivasjon for individuell øving?
4. Hvordan bidrar applikasjonen til bedre organisering og planlegging av øvinger og andre arrangementer?
5. På hvilken måte forbedrer applikasjonen kommunikasjon mellom ledere og kormedlemmer?

Effekt på øving

Gjennom brukertesting og undersøkelser har vi fått verdifulle tilbakemeldinger på hvordan korvirksomheten ble påvirket av Chorium. På spørsmål om Chorium effektiviserte øvingen til brukerne var gjennomsnittlig svar 4,36 av 5, noe som tyder på at ja, en slik samlet løsning kan hjelpe veldig med effektivitet på øving. Når det gjelder tilgjengelighet av øvingsmateriale er det vanskelig å måle tilgjengeligheten ved bruk av andre lagringstjenester versus Chorium, men brukerundersøkelsen viser en gjennomsnittlig poengsum på 4.28 når brukerne er spurt om hvor enkelt det er å finne og organisere musikkstykker.

Effekt på kommunikasjon

Videre ser man fra undersøkelsen at kommunikasjonsverktøyene i Chorium ikke helt har den påvirkningen som vi håpet på i begynnelsen. Kommunikasjon fra ledere til medlemmer rundt arrangementer, kunngjøringer og tilbakemeldinger er en viktig del av korvirksomheten, og det er nødvendig at denne funksjonaliteten tilbyr akkurat det et kor trenger. Disse implementasjonene er kanskje de som ikke har kommet så langt som vi ønsket i løpet av prosjektperioden, og derfor kan det virke som at de ikke er helt der de burde være i forhold til hva kor trenger, og ikke minst i forhold til den tekniske funksjonaliteten. Arrangementer er et sted hvor det har vært en del problemer gjennom utvikling, og det har vært vanskelig å få til en feilfri implementasjon. Kunngjøringer er også en del av applikasjonen som har mye mer potensial enn det som nå er implementert. For øyeblikket er det kun slik at en person kan legge ut en kunngjøring, som dermed vises i en feed. Det er ønskelig å legge til mer funksjonalitet her for å forbedre kommunikasjonen og det sosiale elementet ved kor, dette inkluderer kommentarer og likes.

Kvaliteten på tilbakemeldingene

Selv om brukertesting av Chorium indikerte en høy grad av tilfredshet blant testerne, må vi anerkjenne at utvalget var relativt lite og ikke helt representativt for mer generelle korvirksomheter. Med kun 14 svar på undersøkelsen fra to kor, hvorav flere var bekjente eller fra det samme lokale kormiljøet, er det mulig at svarene var positivt forskjøvet. Dette reiser spørsmål om hvorvidt den store tilfredsheten ville holdt seg i et bredere og mer variert utvalg av brukere.

For å adressere denne potensielle skjevheten, ville det være gunstig å utvide brukerbasen for å inkludere en større og mer variert gruppe deltakere. Dette ville gi en mer nyansert forståelse av appens effektivitet og tilfredshet. Det er også viktig å reflektere over de begrensede ressursene som var tilgjengelig for prosjektet, samt uforutsette risikoer som frafall av faktiske tilbakemeldinger fra testingen.

Web-app

Under brukertesting mottok vi tilbakemeldinger om behovet for en web-løsning for å forenkle administrative oppgaver som opplasting av filer. Selv om dette ikke var essensielt for den grunnleggende funksjonaliteten i appen, var det et sterkt ønske om å inkludere dette i iterasjon 2. Dessverre, på grunn av tidsbegrensninger og tidligere teknologiske valg i utviklingen, var ikke en slik integrasjon mulig.

Spesielt valget av visse plugins og teknologiske løsninger, som lagring av bilder for cache på lokalt filsystem, skapte kompatibilitetsproblemer med en webversjon. For å gjøre

filhåndteringen kompatibel med web, ville store deler av implementasjonen måttet gjøres om, noe som ville kreve mer tid enn det som var tilgjengelig. Dette var ikke forutsett i prosjektplanleggingen, og det understreker viktigheten av å vurdere alle plattformers kompatibiliteter tidlig i designprosessen, for det er ikke nødvendigvis slik at alt er kompatibelt selv om man utvikler med et rammeverk som tillater å bygge til flere plattformer fra samme kildekode.

Læringen fra denne utfordringen er at fremtidige prosjekter vil dra nytte av å ta hensyn til kompatibilitet med plattformene fra starten av, spesielt når tilgang fra flere enheter er ønskelig.

Kodekvalitet

Et område for forbedringspotensial er opprettholdelse av kodekvalitet. Som nevnt i forrige kapittel, ble automatiserte tester og enhetstester nedprioritert til fordel for kontinuerlig manuell testing via emulatorer og fysiske mobile enheter. Selv om denne tilnærmingen er god for rask og umiddelbar testing av brukergrensesnitt, innebar det også at viktige bakgrunnsprosesser, særlig de som omhandler dataflyt og kommunikasjon med databaser, ikke fikk nok oppmerksomheten. Disse komponentene som håndterer brukerdata er kritiske for applikasjonens funksjonalitet og det er avgjørende at disse er robuste og nøyaktige.

Videre refleksjon rundt testingen fremhever en potensiell risiko for at viktige feil i bakgrunnsprosesser kunne forbli uoppdaget, noe som kunne påvirket funksjonalitet på lengre sikt. Mens Flutter består mye av utvikling av brukergrensesnitt gjennom Widgets, som oftest testes best gjennom manuell interaksjon, ville kanskje en mer balansert tilnærming som inkluderte automatiserte enhetstester styrket applikasjonens generelle pålitelighet.

6.2 Vurdering av tilnærminger og valg

Valget av de grunnleggende teknologiene, Flutter og Firebase, bidro til effektiv utvikling, selv om valget av spesifikke plugins førte til problemer når det var tid for å bygge videre på de grunnleggende funksjonelle kravene. En iterativ smidig utviklingsmetodikk gjør det enklere å rydde opp i slike feil valg, ved å tillate en oppdatert versjon i neste iterasjon, om man så har nok tid til å gjennomføre disse endringene. Problemer og bugs kan legges inn på KanBan brettet, og deretter kan utviklerne prioritere om disse skal jobbes på, eller om det er andre ting som har høyere prioritering.

6.3 Helhetlig systemperspektiv

Som det ble påpekt i introduksjonen av problemstillingen, og gjennom personlige erfaring, har det eksistert et merkbart gap i tilgjengeligheten av et helhetlig system for korvirksomhet. Dette synspunktet støttes også av tilbakemeldinger fra andre korgrupper, hvor det kommer frem at det er et behov for funksjonalitetene som Chorium tilbyr. Chorium introduserer ikke akkurat nye teknologiske innovasjoner, men fungerer mer som en “unified platform” som tilbyr funksjoner som for øyeblikket er spredt over flere forskjellige tjenester.

Disse eksisterende tjenestene, som ofte er utviklet av store teknologiselskaper som Google og Meta, har fordelene av store ressurser og avanserte implementeringer. Dette stiller Chorium overfor en utfordring: Kan en samlet løsning som fokuserer på spesifikke behov innen korvirksomhet veie opp for det som måtte mangle av mer avanserte implementeringer med bredere funksjonalitet?

Hovedargumentet for Chorium ligger i dens forenkling og sentralisering av nødvendige aktiviteter innad i kor, som øving og kommunikasjon. Selv om Choriums teknologi kanskje ikke kan sammenlignes med kompleksiteten og funksjonaliteten til mer etablerte tjenester, er dens fokus og brukervennlighet designet for å tilby en løsning som er tilpasset målgruppens spesifikke behov. Ved å tilby en samlet integrasjon av nødvendige funksjoner, sikter Chorium mot å forenkle hvordan kor opererer, redusere behovet for å sjonglere flere plattformer, og forbedre brukeropplevelsen ved å fokusere på målgruppens faktiske behov.

Et annet spørsmål man kan stille seg er om dette er et økonomisk forsvarlig prosjekt.

6.4 Oppsummering

Prosjektgjennomføringen har vært preget av både oppturer og nedturer, men det ferdige produktet har vist seg å være av interesse for målgruppen, ifølge brukerundersøkelsen. Selv om brukerbasen har vært begrenset innenfor studentmiljø, gir tilbakemeldingene et innblikk i appens relevans. Dette reiser spørsmål om appens potensial for å tiltrekke seg et bredere publikum og for å kunne spres til andre geografiske områder og kulturer utenfor det opprinnelige miljøet. Hvis disse tilbakemeldingene kan sees på som en indikasjon for generelle behov i kormiljøer, kan appen ha potensial for videre adopsjon og utbredelse.

7. KONKLUSJON OG VIDERE ARBEID

7.1 Konklusjon

Prosjektet har med suksess utviklet en brukervennlig applikasjon som sentraliserer øvingsfiler, kommunikasjon og kalenderfunksjoner for kor, og dermed effektivt møter de identifiserte behovene innen korvirksomhet. Gjennom en systematisk tilnærming har vi kombinert praktisk utvikling med teoretisk innsikt, hvilket har resultert i en robust og skalerbar løsning.

Utviklingsarbeidet har vist at en dedikert applikasjon for korets virksomhet betydelig forenkler tilgangen til nødvendige filer, noe som i sin tur gjør det lettere for medlemmene å engasjere seg i individuell øving. Denne forbedrede tilgangen til øvingsmateriale forventes å øke motivasjonen blant kormedlemmene til å øve mer på egen hånd.

Selv om vi har oppnådd betydelige fremskritt innenfor filhåndtering og individuell øving, har evalueringen identifisert forbedringspotensial når det gjelder organisering og kommunikasjon. Disse funksjonene var i en tidlig utviklingsfase, og de utfordringene som ble identifisert, gir oss verdifulle lærdommer om viktigheten av tidlig brukerinnsikt og tilbakemelding i utviklingsprosessen. Vi har innsett at færre, men mer målrettede iterasjoner kunne ha gitt mer spesifikke tilbakemeldinger, og dermed gjort det mulig å utføre flere iterasjoner innenfor prosjektperioden.

Gjennom vår systematiske og iterative tilnærming har vi kontinuerlig samlet inn og analysert tilbakemeldinger fra brukerne, noe som har muliggjort løpende forbedringer av applikasjonen. Denne prosessen har vist seg å være avgjørende for å utvikle en løsning som er både funksjonell og brukervennlig. Tilbakemeldinger indikerer at brukerne setter pris på den enkle tilgangen til øvingsmateriale, som har gjort det enklere å øve selvstendig.

7.2 Videre arbeid

Videre ønsker vi å fullføre iterasjon 2 slik det først var planlagt. I denne iterasjonen har brukergrensesnittet fått et stort løft, sammen med forbedringer til kalenderen og kunngjøringene. Dette vil gi en mer intuitiv og brukervennlig opplevelse, og forbedre planlegging og kommunikasjon innen koret.

Det vil også bli tatt ekstra hensyn til prinsippene bak DSR. Under prosjektet har vi sett stor verdi av denne metodikken, og vi planlegger å utnytte den i større grad fremover. Spesielt vil vi fokusere på å redusere antall iterasjoner for å få mer målrettet tilbakemelding på spesifikke funksjoner, noe som kan føre til flere iterasjoner innenfor prosjektperioden.

Vi vil også utforske hvordan vi kan ytterligere utnytte teori fra musikkens verden i utviklingen vår. I en bok publisert av Routledge (McPherson et al., 2017) utforsker Gary E. McPherson hvordan man kan anvende selvregulert læring i sammenheng med musikk. Her kan man trekke ut konkrete eksempler på hva som kan forbedre egen læring. Disse kan da videre bli overført til funksjonalitet i applikasjonen for å gi brukeren en bedre læringsopplevelse.

Integrering av Selvregulert Læring

- **Målsetting**
 - **Langsiktige og kortsiktige mål:** Det er viktig å ha klare mål for øvingen, både langsiktige mål med setlister til hver opptreden og kortsiktige mål som daglig øving. For eksempel kan “streaks” og små oppfordringer fra appen bidra til å implementere målsettingen inn i øvingen, noe som kan forbedre brukernes engasjement og progresjon.
- **Selvovervåkning**
 - **Progresjonssporing:** For å se tydelig progresjon kan det være nyttig å se hvordan man ligger an. Her kan vi bygge videre på muligheten til å se progresjonen sin på sanger og setlister. Videre kan “streaks” fungere som en motivasjonsfaktor, ved å oppmuntre til daglig bruk og synliggjøre kontinuerlig fremgang.
- **Selvrefleksjon**
 - **Opptak og avspilling:** Det kan være en idé å la brukeren ta egne lydopptak mens de synger sammen med lydfilene, for deretter å høre på opptakene i etterkant. Dette er en praksis som allerede brukes i korøvinger, men det er rom for å intensivere bruken gjennom applikasjonen.
- **Motivasjon**
 - **Forbedret motivasjon gjennom streaks:** En studie fra University of Pennsylvania (Silverman, 2019) har vist at streaks har en positiv innvirkning på motivasjonen. Denne konklusjonen støttes av en analyse av Duolingo, som viser at lange streaks har en enda større innvirkning på brukernes motivasjon til å fortsette øvingen (Huynh and Lida, 2017). Ved å integrere slike streak-funksjoner kan vi øke brukerens engasjement og motivasjon til regelmessig øving.

Ved å utforske og potensielt integrere disse prinsippene fra selvregulert læring, sikter vi mot å forbedre brukeropplevelsen og øke effektiviteten i egenøvingen ytterligere. Dette

vil ikke bare bidra til en mer engasjerende øvingsprosess, men også fremme bedre læringsutbytte for kormedlemmene.

8. REFERANSER

- IBM (u.å.) 'What is mobile application development?', Tilgjengelig fra: <https://www.ibm.com/topics/mobile-application-development> (Hentet: 25. Februar 2024).
- Gamble, M. (2022) 'What is Native Mobile Development? (Benefits, Tools, Resources)', Couchbase, Tilgjengelig fra: <https://www.couchbase.com/blog/native-mobile-development/> (Hentet: 11. Mai 2024).
- Griffith, C. (u.å.) 'What is Hybrid Mobile App Development?', Ionic, Tilgjengelig fra: <https://ionic.io/resources/articles/what-is-hybrid-app-development> (Hentet: 11. Mai 2024).
- Dutta, S. (u.å.) 'Everything You Need to Know About Hybrid Mobile App Development', Turing, Tilgjengelig fra: <https://www.turing.com/resources/hybrid-mobile-app-development> (Hentet: 11. Mai 2024).
- Obum (2022) 'Why You Should Use Flutter for Your Projects', freeCodeCamp, Tilgjengelig fra: <https://www.freecodecamp.org/news/why-you-should-use-flutter/> (Hentet: 25. Februar 2024).
- Stevenson, D. (2018) 'What is Firebase? The complete story, abridged.', Medium, Tilgjengelig fra: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0> (Hentet: 11. Mai 2024).
- Spinellis, D. (2012) 'Git', *IEEE Software*, vol. 29, no.3, pp. 100-101, doi: 10.1109/MS.2012.61.
- Flutter (u.å.) 'Build and release an iOS app', Flutter, Tilgjengelig fra: <https://docs.flutter.dev/deployment/ios> (Hentet: 11. Mai 2024).
- Trello (u.å.) 'What is Trello?', Tilgjengelig fra: <https://support.atlassian.com/trello/docs/what-is-trello/> (Hentet: 12. Mai 2024)
- Mistry, K. (2023) 'Understanding Flutter Widgets: A Beginner's Guide to Building User Interfaces', Medium, Tilgjengelig fra: <https://medium.com/@kavyamistry0612/understanding-flutter-widgets-a-beginners-guide-to-building-user-interfaces-eed1acf17070> (Hentet: 11. Mai 2024).
- Chinn, A. (2022) 'What's the System Usability Scale (SUS) & How Can You Use It?', HubSpot, Tilgjengelig fra: <https://blog.hubspot.com/service/system-usability-scale-sus> (Hentet: 19. April 2024).
- Codacy (2023) '7 Metrics for Measuring Code Quality', Tilgjengelig fra: <https://blog.codacy.com/code-quality-metrics> (Hentet: 11. Mai 2024).
- OpenAI (2024) 'What is ChatGPT?', Tilgjengelig fra: <https://help.openai.com/en/articles/6783457-what-is-chatgpt> (Hentet: 11. Mai 2024).

Kalliamvakou, E. (2022) 'Research: quantifying GitHub Copilot's impact on developer productivity and happiness', GitHub Blog, Tilgjengelig fra: <https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/> (Hentet: 11. Mai 2024).

Kalliamvakou, E. (2024) 'A developer's second brain: Reducing Complexity through partnership with AI', GitHub Blog, Tilgjengelig fra: <https://github.blog/2024-01-17-a-developers-second-brain-reducing-complexity-through-partnership-with-ai> (Hentet: 11. Mai 2024).

Perera, R. (2023) 'What is Figma? (And How to Use Figma for Beginners)', Theme Junkie, Tilgjengelig fra: <https://www.theme-junkie.com/what-is-figma/> (Hentet: 11. mai 2024).

Banús, O. (2023) 'Responsive UX design guide: From mobile to desktop', LogRocket, Tilgjengelig fra: <https://blog.logrocket.com/ux-design/responsive-design-guide/> (Hentet: 11. Mai 2024).

Blavo, K. (2020) 'Aesthetics vs Functionality: Striking a Balance in UX Design', Medium, Tilgjengelig fra: https://medium.com/@kikelomo_blavo/aesthetics-vs-functionality-in-user-experience-design-82e7f0462f31 (Hentet: 12. Mai 2024).

UUTilsynet (u.å.) 'Kvifor universell utforming av ikt?', Tilgjengelig fra: <https://www.uutilsynet.no/veiledning/kvifor-universell-utforming-av-ikt> (Hentet: 11. Mai 2024).

Malt, U. og Grønmo, S. (2020) 'Likert-skala', Store Norske Leksikon, Tilgjengelig fra: <https://snl.no/Likert-skala> (Hentet: 11. Mai 2024).

Rehkopf, M. (u.å.) 'What is a kanban board?', Atlassian, Tilgjengelig fra: <https://www.atlassian.com/agile/kanban/boards> (Hentet: 11. Mai 2024).

A. Mehr, Samuel (2018) 'Form and Function in Human Song', Tilgjengelig fra: [https://www.cell.com/current-biology/fulltext/S0960-9822\(17\)31675-5](https://www.cell.com/current-biology/fulltext/S0960-9822(17)31675-5) (Hentet: 11. Mai 2024).

Silverman, Jackie (2019) 'Consumer Streaks', Tilgjengelig fra: <https://repository.upenn.edu/server/api/core/bitstreams/71df9803-66fb-43b2-9693-13177f634da5/content> (Hentet 11. Mai 2024).

Huynh, Duy og Iida, Hiroyuki (2017) 'An analysis of winning streak's effects in language course of "Duolingo"', Tilgjengelig fra: https://www.researchgate.net/publication/321123886_An_Analysis_of_Winning_Streak%27s_Effects_in_Language_Course_of_Duolingo (Hentet: 11. Mai 2024).

Brocke, Jan vom (2020) 'Introduction to Design Science Research', Tilgjengelig fra: https://www.researchgate.net/publication/345430098_Introduction_to_Design_Science_Research (Hentet 12. Mai 2024).

Firebase (2024) 'Firebase Crashlytics', Tilgjengelig fra: <https://firebase.google.com/docs/crashlytics> (Hentet: 12. Mai 2024).

Firebase (2024) 'Firebase Security Rules' Tilgjengelig fra: <https://firebase.google.com/docs/rules> (Hentet: 12. Mai 2024).

Google Analytics (2024) 'Start learning about Google Analytic', Tilgjengelig fra: <https://developers.google.com/analytics/learn> (Hentet 12. Mai 2024).

Google Maps Platform (2024) 'Google Maps Platform Documentation' Tilgjengelig fra: <https://developers.google.com/maps/documentation> (Hentet 12. Mai 2024).

Google Play (2024) 'Release with confidence', Tilgjengelig fra: <https://play.google.com/console/about/guides/releasewithconfidence/> (Hentet 12. Mai 2024).

Syncfusion (2024) 'Flutter Calendar Widget', Tilgjengelig fra: <https://www.syncfusion.com/flutter-widgets/flutter-calendar> (Hentet 12. Mai 2024).

McPherson, G.E. (2017). Self-Regulated Learning in Music Practice and Performance. Funnet i: Handbook of Self-Regulation of Learning and Performance. Routledge.

8.1 Illustrasjoner

Davies, A. (u.å.) 'What is Agile Methodology?', DevTeam.Space, Tilgjengelig fra: <https://www.devteam.space/blog/what-is-an-agile-methodology/> (Hentet: 12. Mai 2024).

ClearSurvey (u.å.) 'What is a Likert Scale?', Tilgjengelig fra: <https://clearsurvey.com/what-is-a-likert-scale/> (Hentet: 12. Mai 2024).

Peppers Ken. (2008) 'A Design Science Research Methodology for Information Systems Research', Tilgjengelig fra: <https://www.semanticscholar.org/paper/A-Design-Science-Research-Methodology-for-Systems-Peppers/97448a54ec6d1164ea340672c86d0b43449dd9c7> (Hentet: 12. Mai 2024).

9. VEDLEGG

Vedlegg 1: Prosjekthåndbok

Vedlegg 2: Visjonsdokument

Vedlegg 3: Kravdokument

Vedlegg 4: Systemdokumentasjon

Vedlegg 5: Kildekode

Vedlegg 6: Videodemo

Vedlegg A: Brukerundersøkelse oppsett

For hver påstand svarer brukeren med et tall på en skala fra 1 til 5, hvor 1 betyr helt uenig/veldig unyttig og 5 betyr helt enig/veldig nyttig.

Del 1 (System Usability Scale):

Jeg tror jeg ville brukt Chorium regelmessig.

Jeg syntes applikasjonen var unødvendig komplisert.

Jeg syntes Chorium var enkel å bruke.

Jeg tror jeg ville trengt hjelp av en teknisk person for å klare å bruke applikasjonen.

Jeg syntes de ulike funksjonene i applikasjonen var godt integrert.

Jeg tenker det var for mye inkonsistens i Chorium.

Jeg kan forestille meg at de fleste brukere vil lære seg å bruke Chorium veldig raskt.

Jeg syntes Chorium var veldig kronglete å bruke.

Jeg følte meg veldig trygg på å bruke Chorium.

Jeg trengte å lære meg mye nytt før jeg kunne komme i gang med å bruke Chorium.

Del 2 (Øvingskultur):

Jeg føler Chorium effektiviserte øvingen min.

Jeg øver mer på grunn av Chorium.

Jeg ville anbefalt Chorium til andre kor.

Hvor nyttig er kommunikasjonsverktøyene i Chorium for å planlegge og koordinere med kormedlemmer og dirigent?

Hvor enkelt er det å finne og organisere noter og musikkstykker i Chorium?

Gir Chorium deg nyttige tilbakemeldinger eller verktøy for selvstudium og øving?

Hvordan opplever du lyd kvaliteten og brukervennligheten til avspillingsfunksjoner i Chorium?

Valgfritt:

Har du noen andre tanker eller tilbakemeldinger?

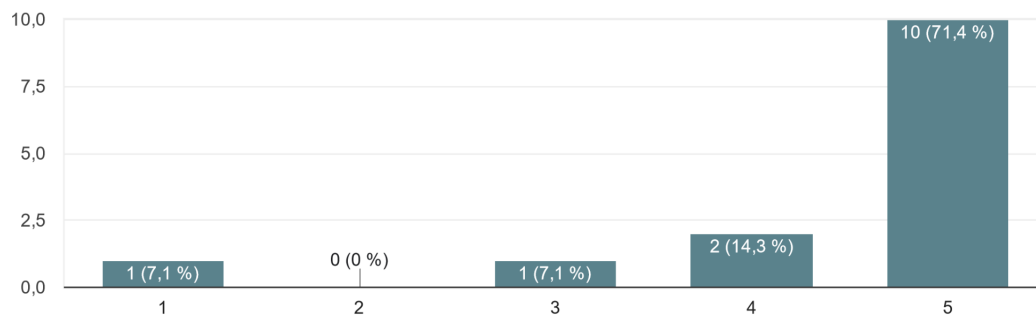
Vedlegg B: Resultater fra brukerundersøkelse iterasjon 1

14 svar.

Del 1 (System Usability Scale):

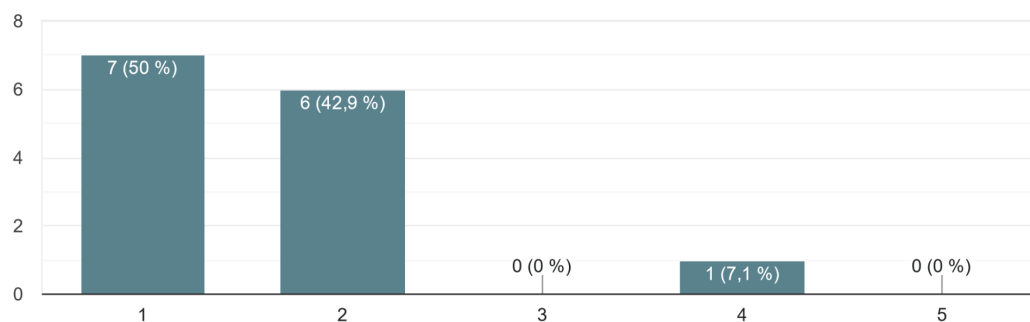
Jeg tror jeg ville brukt Chorium regelmessig

14 svar



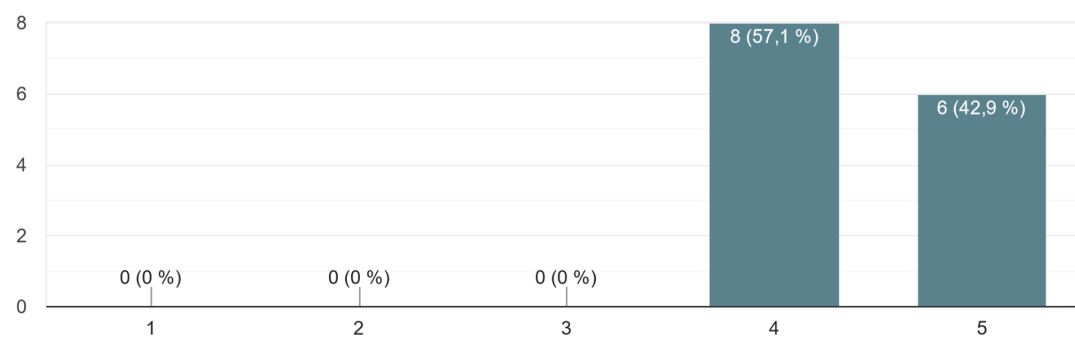
Jeg syntes applikasjonen var unødvendig komplisert

14 svar



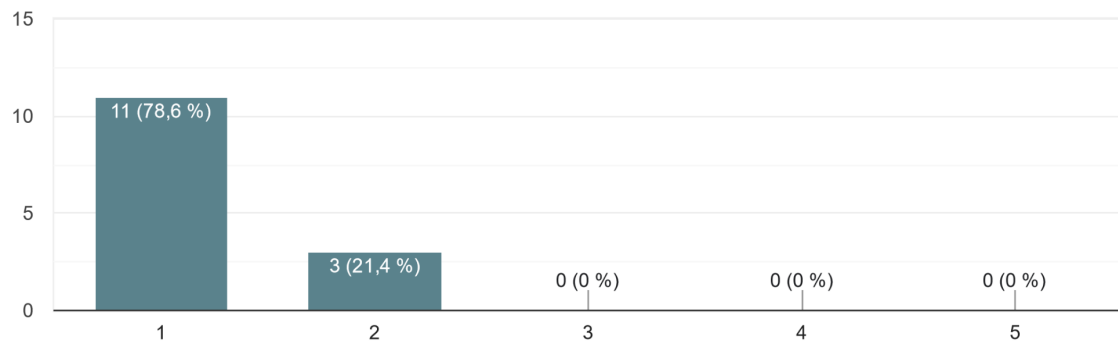
Jeg syntes Chorium var enkel å bruke

14 svar



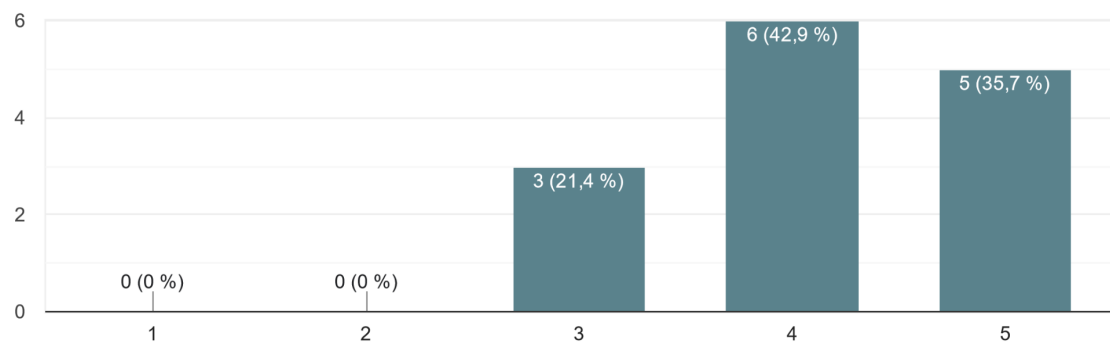
Jeg tror jeg ville trengt hjelp av en teknisk person for å klare å bruke applikasjonen

14 svar



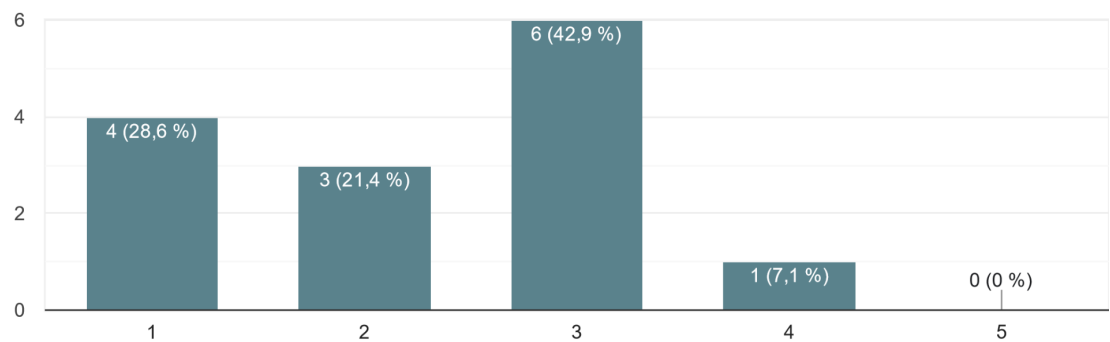
Jeg syntes de ulike funksjonene i applikasjonen var godt integrert

14 svar



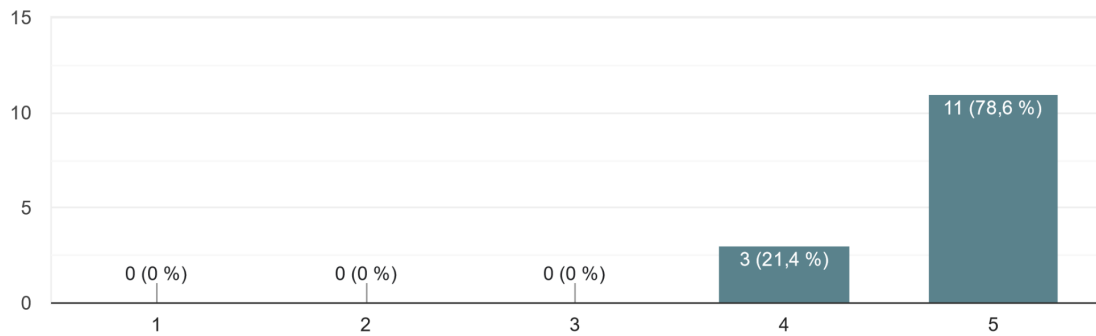
Jeg tenker det var for mye inkonsistens i Chorium

14 svar



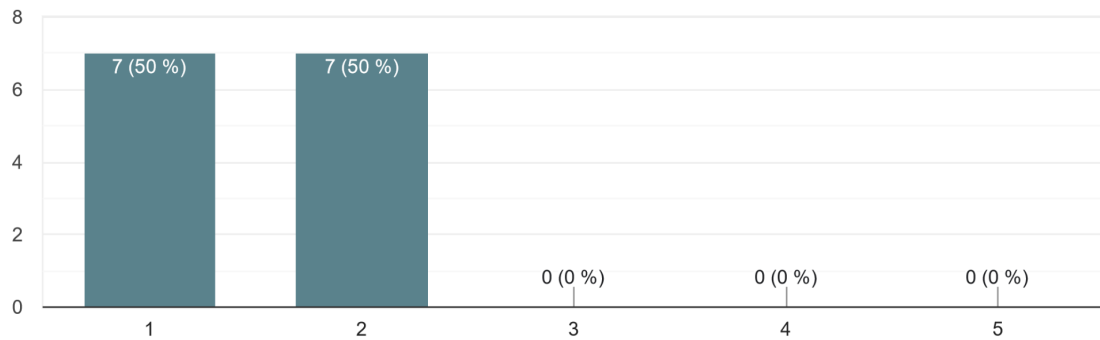
Jeg kan forestille meg at de fleste brukere vil lære seg å bruke Chorium veldig raskt.

14 svar



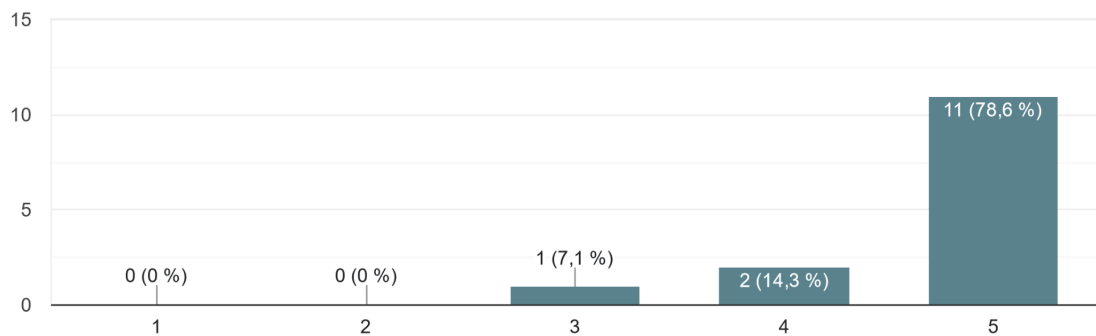
Jeg syntes Chorium var veldig kronglete å bruke.

14 svar



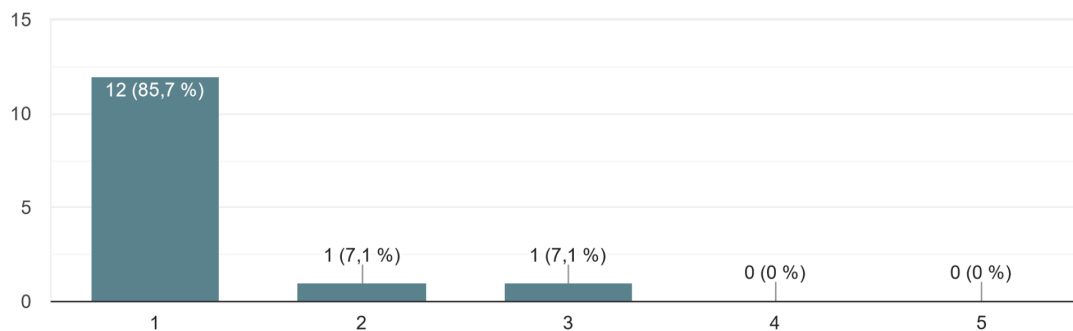
Jeg følte meg veldig trygg på å bruke Chorium

14 svar



Jeg trengte å lære meg mye nytt før jeg kunne komme i gang med å bruke Chorium.

14 svar

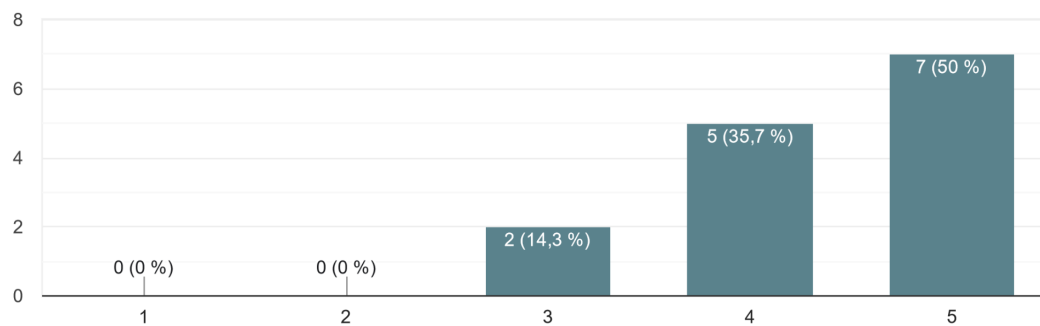


Resultat: 86.6 / 100

Del 2 - Øvingskultur

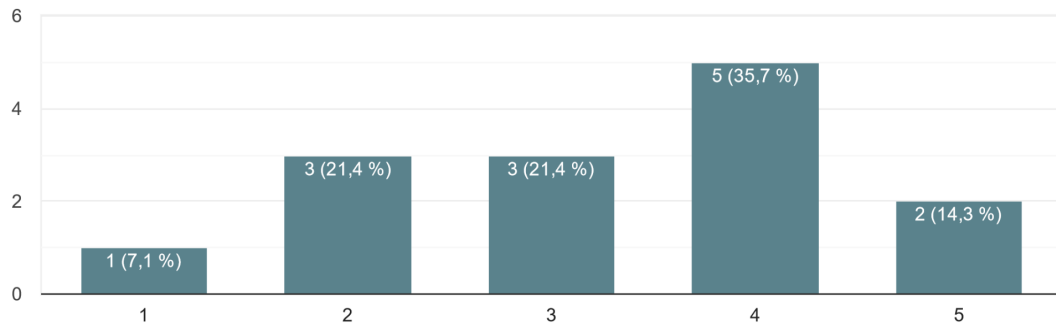
Jeg føler Chorium effektiviserte øvingen min

14 svar



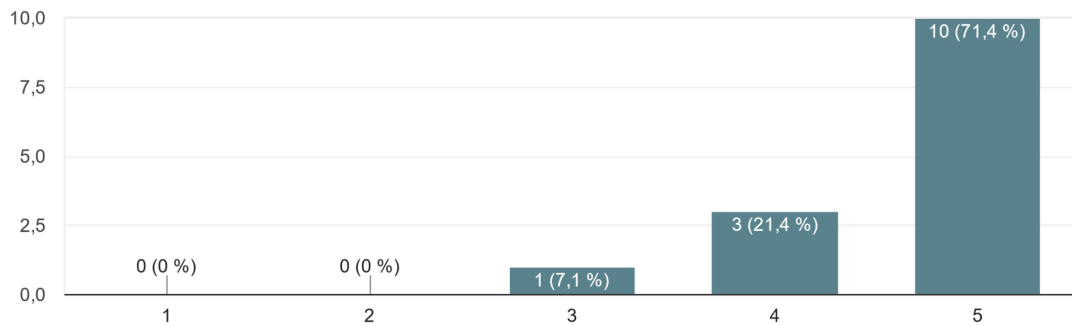
Jeg øver mer på grunn av Chorium

14 svar



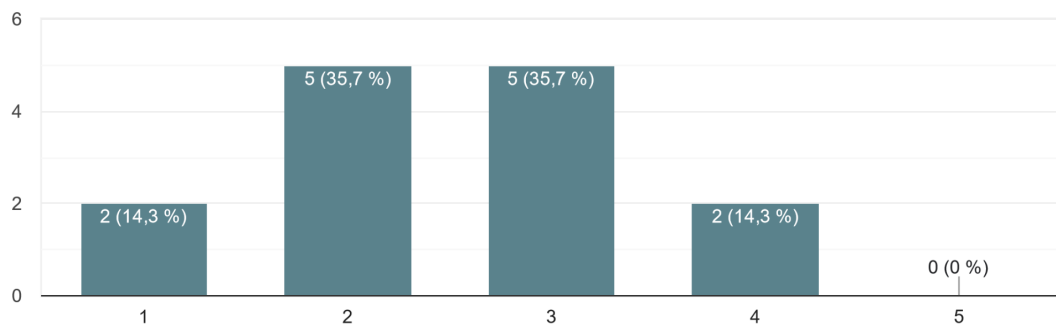
Jeg ville anbefalt Chorium til andre kor

14 svar



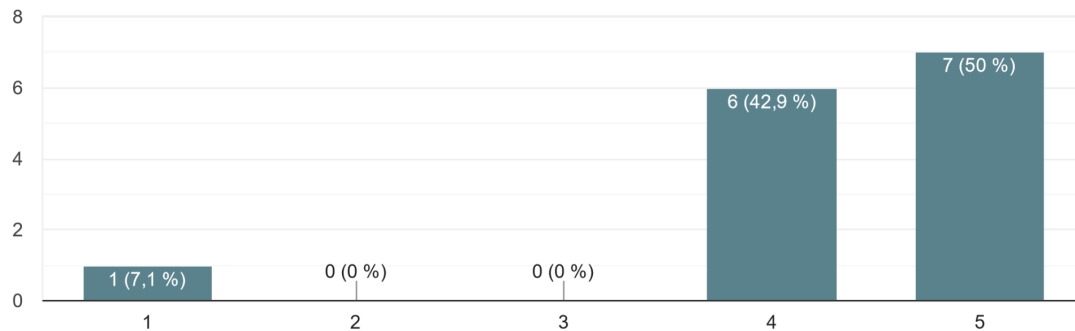
Hvor nyttig er kommunikasjonsverktøyene i Chorium for å planlegge og koordinere med kormedlemmer og dirigent?

14 svar



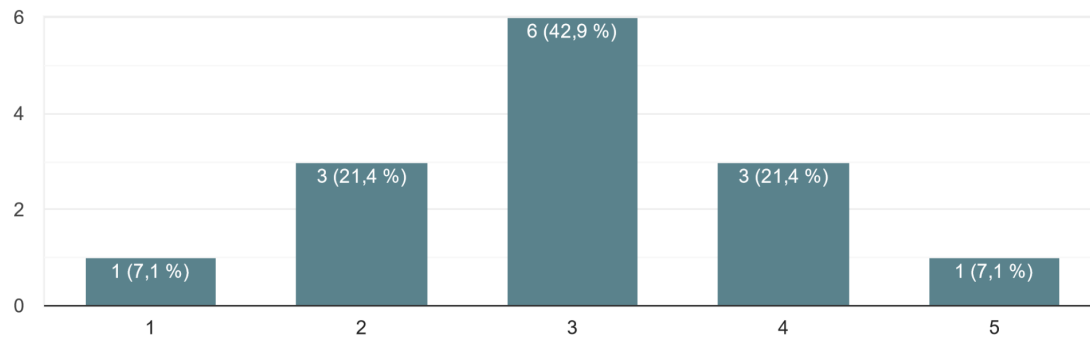
Hvor enkelt er det å finne og organisere noter og musikkstykker i Chorium?

14 svar



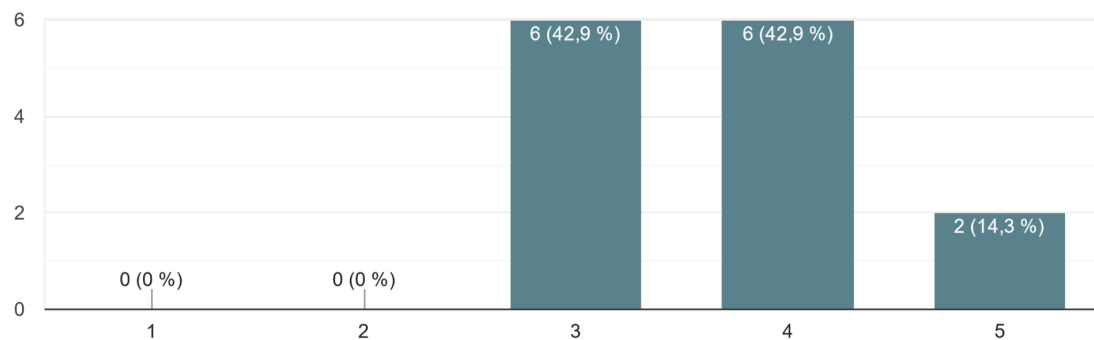
Gir Chorium deg nyttige tilbakemeldinger eller verktøy for selvstudium og øving?

14 svar



Hvordan opplever du lyd kvaliteten og brukervennligheten til avspillingsfunksjoner i Chorium?

14 svar



Tekstboks for tilbakemeldinger utenfor spørsmålene:

- "Veldig bra app. Bruker den masse! Er bare en tilbakemelding jeg har som kan være litt irriterende og det er at man noen ganger må ut og inn i appen for å spille av en ny lydfil."
- "Ved avspilling av lydfilene er det ikke automatisk når du går innpå en ny låt(den siste låten du hørte på er alltid valg). At en fysisk må endre hvilke lydfil hver gang det skal lyttes til ny fil(om du bytter låter). Kanskje om det var tutti som kom opp som lydfil forslag til avspilling ved avspilling når man trykket på «ny» låt avspilling. Eller automatisk kom opp som forslag ved stemmegruppen som en tilhører. Veldig godt verktøy til kor og øving! Og kjekt med farger som matcher kor man er med i og egen profil og en kan ha bilder til div innpå appen :-)"
- "Veldig fornøyd! Det eneste er at noen ganger får jeg ikke spilt av en ny lydfil etter å ha spilt av en annen, inne på en annen sang. Da må jeg nullstille appen, hver gang jeg bytter sang."
- "Lydavspilleren «crasher» veldig ofte. Hvis du hører på en sang og bytter vil den ofte stoppe. Samme skjer noen ganger når du vil starte sangen på ny. Da må man avslutte appen og gå inn på ny, men så fungerer det igjen. :)"
- "Veldig bra app når den fungerer, men må ofte gå ut og inn av appen fordi hele skjermen blir grå når jeg trykker på sanger, og noen ganger vil ikke lydfilet dirigenten laster opp spille av."
- "Noen ganger åpner ikke appen opp «sanger» og må restartes. Den er vanskelig å spole i låter ofte og det er ikke alltid den henger med når man bytter lydfil. Mulighet for å sakke ned lydfilene hadde vært flott! Jeg syns det er litt kronglete å legge inn lydfilet (men det er fordi jeg liker å gjøre det på PC heller enn telefon)"
- "Er en bug når man går inn på sanger, må lukke appen så gå inn igjen"
- "vil gjerne kunne ha notene som en rull i stedet for å bla side til side for jeg mister hvor jeg var og må zoome inn igjen for å finne fram (: kanskje et alternativ mellom rull og bla?"