

Prosjekt plan gruppe 5  
PROG2052 - Integrasjonsprosjekt

*Thomas Øvstun Ditman*

*Harald Andreas Løkkeberg Hansen*

*Alin Mihai Napirca*

September 2022

# Innholdsfortegnelse

<b>Innholdsfortegnelse</b>	<b>2</b>
<b>1.0 Mål</b>	<b>3</b>
1.1 Resultatmål	3
1.2 Effektmål	3
1.3 Læringsmål	3
<b>2.0 Omfang</b>	<b>4</b>
2.1 Oppgavebeskrivelse	4
2.2 Avgrensning	4
2.3 Fagområde	4
<b>3.0 Prosjektorganisering</b>	<b>5</b>
3.1 Rolleansvar	5
3.2 Rutiner og regler i gruppa	5
<b>4.0 Planlegging</b>	<b>7</b>
4.1 Oppsett av scrum	7
4.2 Milepæler	7
4.3 Beskrive gruppa sin måte å følge modellen på	7
<b>5.0 Organisering av kvalitetssikring</b>	<b>8</b>
5.1 Plan for dokumentasjon, konfigurasjonsstyring, verktøybruk, kildekode	8
5.2 Tidlig risikoanalyse	9
<b>6.0 Fremdriftsplan for gjennomføring</b>	<b>12</b>
6.1 Gantt-skjema for hele perioden	12
6.2 Product backlog	12

# 1.0 Mål

## 1.1 Resultatmål

Hovedmålet med prosjektet er å lage en applikasjon som vil fungere som et galleri for ulike utviklere og utviklingstudioer å presentere alle sine programvarer på et sted. Applikasjonen skal også kunne sortere og presentere programvare basert på valgt land og gjøre utvikler-sorteringen mer generell. Prosjektet vil være utviklet gjennom Android Studio der hoved programmeringsspråkene vil være Kotlin og Java. Dette vil da innebære at applikasjonen kan kun brukes av Android brukere og hente programvare fra Google Play Store. Selvsagt vil en iOS implementasjon være det “neste” steget, men det er utenfor vår prosjektramme for denne gangen.

## 1.2 Effektmål

Formålet med vår oppgave er utvikle en smart løsning for et appgalleri som skal kunne benyttes av både små og store utviklere/utviklingstudioer til å presentere sin programvare. Vi ser for oss at dette kan være en god løsning for å utviklere til å kunne presentere hva de utvikler og hvilken programvarer de har utviklet, som også er lett tilgjengelig for alle sammen. Dette vil være nyttig ved for eksempel utstillinger der mange utviklere samler seg for å vise fram sine programvarer og sine nyeste lanseringer, eller bare for folk flest å kunne se og lære mer om hva ulike utviklere har å tilby.

## 1.3 Læringsmål

Læringsutbyttet for oppgaven er hovedsakelig anvendelsen av agil arbeidsmetodikk og bruken av Scrum, og å utvide på kunnskapen vi har samlet opp etter 2 år på bachelor studiet. Scrum er en populær og viktig rammeverk i IT verdenen og det er viktig å anvende den tidlig, samtidig er det viktig å kunne bruke kunnskapen vi har samlet på i løpet av studiet, men kunne bygge på det vi allerede vet som i dette prosjekt gjør ved å anvende Kotlin kode i vårt prosjekt og utvikle en mobil applikasjon.

## 2.0 Omfang

### 2.1 Oppgavebeskrivelse

Omfanget av dette prosjektet er å lage en applikasjon som skal fungere som et galleri der utviklere og utviklingsstudioer kan legge ut sin programvare til utstilling. Til grunn skal vi utvikle en mobilapplikasjon for Android-enheter som ulike brukere kan benytte for å søke opp ulike utviklere og se hva de har laget av applikasjoner. De skal kunne sortere etter utviklere, men også etter land og se hva ulike utviklere og utviklingsstudioer fra enkelte land har publisert på Google Play Store ved å bruke API-funksjonalitet.

### 2.2 Avgrensning

For dette prosjektet har vi lagt inn et par avgrensninger. Disse eksisterer for at prosjektet ikke får et for stort omfang, samtidig som det blir oppnåelig innenfor tidsrammen vi har fått.

1. Appen som blir laget fra dette prosjektet skal kun kunne brukes av Android brukere, om mer tid hadde vært tilgjengelig ville IOS kompatibilitet blitt en godt egnet egenskap å innarbeide.
2. I prosjektet antar vi at brukere av applikasjonen kan engelsk, som betyr at vi bare lager applikasjonen med engelsk som antatt bruker språk.
3. Applikasjonen bestemmer ikke hvilket apper som er laget av brukeren. Brukeren kan putte inn hvilket som helst applikasjon som er på Google play store<sup>1</sup>.

### 2.3 Fagområde

- IDG1362 Introduction to user-centered design / Designtenkning:
  - Lage prototyper
  - teste prototyper på potensielle brukere
  - personas.
- PROG1004 Programvareutvikling:
  - Utviklingsrammeverk og agil arbeidsmetodikk
  - Diagrammer & Wireframes
  - MVP
- PROG2005 Cloud Technologies:
  - REST APIer og cloud baserte tjenester
- PROG2007 Mobile Programming
  - Mobil app utvikling (Kotlin & Java)
- IDATG2204 Datamodellering og databasesystemer
  - Databaser

---

<sup>1</sup> <https://play.google.com/store/games?hl=no>

## 3.0 Prosjektorganisering

### 3.1 Rolleansvar

#### Scrum Master

- Alin Napirca
- Ansvarlig for prosjektet på høyt nivå. Er vi i rute? Har alle gruppemedlemmer noe å gjøre på? Har det oppstått problemer som krever innkalling til møter?
- Ansvarlig for å sette opp møter.

#### Utviklere

- Alin Napirca, Thomas Øvstun Ditman, Harald Andreas Løkkerberg Hansen.
- Ansvarlig for å utvikle applikasjonen, jobbe med sine oppgaver, stille opp på møter.

#### Veileder

- Tom Røise
- Gir veiledning en gang i uka med mindre gruppen ikke trenger det på det tidspunktet.

### 3.2 Rutiner og regler i gruppa

#### Møter

Møter innkalles etter behov. Tidlig i prosjektarbeidet vil møtene holdes fysisk som vi kan eventuelt flytte det over til digitale møter etter hvert, eller ved behov. Møtene skal brukes for å informere hverandre om ideer, hva som er gjort/hva som mangler å bli gjort, snakke om prosjektet og prosjekt dokumenter. Hvis alt er klart og alle medlemmer har noe å jobbe med, er det ikke behov for et møte, men hvis det oppstår «sperringer» som hindrer arbeid eller ting som må diskuteres, planlegges det et møte. Planen er å holde minst to møter hver uke, en i begynnelsen av uken for å planlegge sprinten (Sprint Backlog) og et i slutten av uken for å få en gjennomgang av den ukens sprint (Sprint Review). Vi starter med ukentlige sprinter i begynnelsen når arbeidsoppgavene endå er relativ små og gjennomførbare på liten tid før vi eventuelt går over til 2-ukers sprinter når utviklingen av programvaren er vel godt i gang.

#### Varsler

Varsler om møter går ut som direkte meldinger eller i gruppekanalen i Discord som alle gruppemedlemmene har tilgang til. Tid og sted avtales ut fra de enkelte medlemmenes timeplaner og muligheter. Generelt tilstreber vi å ha 2 møter i uken, men dette kan endres ved behov.

Dersom et medlem ikke kan møte opp til et møte eller støter på uregelmessigheter i sine oppgaver, er det deres ansvar å varsle prosjektleder. Prosjektlederen oppdaterer deretter de andre medlemmene om endringer basert på nyhetene.

## Gruppemedlemmenes' plikt rundt oppgaver

Gruppemedlemmene er selv ansvarlige for sine egne oppgaver og hvordan disse oppgavene passer inn i prosjektet. Hvis det oppstår problemer, må medlemmene som er tildelt de relevante oppgavene finne ut hvordan de skal gå frem sammen, sette opp et møte, dele arbeidsmengden og om rådgivning fra andre gruppemedlemmer/lærer.

## Innleveringer

Prosjektleder har ansvar for at alle oppgaver er i rute. Hvert medlem har ansvar for kvalitetskontroll av eget arbeid, dersom de trenger hjelp til det kan de henvende seg til andre medlemmer for å få hjelp. Arbeidet leveres til slutt av prosjektleder når alle på gruppa er fornøyd med arbeidet.

## Oppmøte og forberedelse

Medlemmene i gruppen har ulike timeplaner så møtetider kan variere fra uke til uke etter hva som passer for hver enkelt av oss. Alle ukedagene er åpne for møter, både fysiske og digitale.

Hvis forberedelse til et møte er nødvendig, vil medlemmene bli varslet på Discord.

## Tilstedeværelse og engasjement

Gruppa kan ikke forvente at medlemmene ofrer andre fag/møter/jobb for gruppens skyld, men hvis et medlem er tilgjengelig på møtetidspunktet forventes det at de møter opp fysisk/digital.

I møter forventes det at medlemmene er fokusert på oppgavene som skal gjøres, og at de ikke tuller og går på avveie fra møtene.

## Uenighet og kontraktsbrudd

Vi søker å håndtere alle problemer tidligst mulig (på sine røtter). Medlemmene oppfordres til å diskutere saker som dukker opp på en vennlig og konstruktiv måte. Hvis en sak strekker seg ut og uenighet oppstår på tvers av gruppa, vil prosjektleder gripe inn og ta en endelig beslutning.

## 4.0 Planlegging

### 4.1 Oppsett av scrum

For å følge scrum i dette prosjektet så har vi satt opp sprints som skal følges gjennom hele prosjektet. Hver scrum varer for øvrig en uke, men det er mulig at vi øker sprints til to uker i lengde om vi ser at det passer prosjektet bedre.

Hver sprint starter med et fysisk sprint planleggingsmøte, hvor vi går gjennom hvilket produkter i backloggen som skal tas tak i for den uken. Gjennom uken har vi enda et fysisk møte hvor vi arbeider med produktene som ble valgt i planleggingsmøtet. Ved slutten av uken så har vi et digitalt sprint review, hvor vi kort snakker om det vi har arbeidet med, og det vi har klart å gjennomføre. Utenom disse møtene så forventes det også at hvert teammedlem skal jobbe med sine valgte issues/oppgaver.

For å holde styr på hvilket issues som blir tatt opp av hver person så bruker vi kitemaker.co<sup>2</sup>. Dette lar oss enkelt se hvem som jobber på hvilken oppgave, og hvor langt gjennom prosessen denne oppgaven er.

Vi har også delt inn gruppen i roller som tilsvarer rollene i scrum modellen. Vi har satt en scrum master, sammen med et par utviklere.

### 4.2 Milepæler

1. 13.Sep: Første utkast av Prosjektrapport.
2. 30.Sep: Hoveddesign for applikasjon valgt.
3. 14.Okt: Første statusrapport, kravspek, Arkitekturskisse, og MVP.
4. 12.Nov: Veldig ferdig med kode.
5. 17.Nov: Andre statusrapport, tidlig utkast til prosjektrapport.
6. 8.Dec: Siste innlevering av rapport, kildekode, og demovideo.

### 4.3 Beskrive gruppa sin måte å følge modellen på

Scrum modellen er et arbeids rammeverk som tar i bruk mye kontakt med kunden, og gjennom bruk av adaptiv arbeidsgang for å kunne effektivt lage produktet som kunden er etter. Scrum modellen tar i bruk flere roller som teamet skal følge, i vårt tilfelle er ikke alle rollene relevante siden vi ikke produserer et produkt for en potensiell kunde. Dette gjør at de scrum relevante rollene er: Scrum-/teamleder, og utviklere. Scrumleder er ansvarlig for å gi ut oppgaver på starten av sprinten, under sprint planleggingsmøtet. Utviklere er ansvarlig for selve arbeidet på disse oppgavene.

For å følge scrum modellen har vi valgt å bruke kitemaker.co til å holde styr på oppgavene våre. Disse oppgavene er delt inn i 5 kolonner:

---

<sup>2</sup> kitemaker.co

- **Product backlog** Dette er samlingen av alle oppgavene som vil bli ferdig innen slutten av prosjektet
- **Todo** Oppgaver hentet fra backloggen som skal bli jobbet med i løpet av de neste par sprintene.
- **In progress** Oppgavene som har blitt valgt for arbeid under denne ukentlige sprinten
- **For Review** Oppgaver som er ment til å bli ferdige, men trenger et ekstra par øyne til å dobbeltsjekke
- **Complete** Oppgaver som er ferdige

## 5.0 Organisering av kvalitetssikring

### 5.1 Plan for dokumentasjon, konfigurasjonsstyring, verktøybruk, kildekode

Dokumenter som det jobbes aktivt med deles i Google Docs og Google Drive, eller som enkelte tekst filer, men alt må etter hvert i Google Docs for å gi alle på gruppa tilgang til de enkelte dokumentene. «Real-time» deling- funksjonaliteten og enkelheten til google-produkter gjør dem til den optimale tjenesten for oss. Ved hver iterasjon blir dokumentene lastet opp/pusket til GitLab for arkivering.

Google docs: Skrive rapporter

Google drive: Oppbevaring av google docs dokumenter

Gitlab: oppbevare kildekode og dokumentasjon.

Kitemaker: Verktøy for å holde styr på hvilke oppgaver som ikke er gjort, under arbeid, og ferdig.

Android studio: IDE for utvikling av mobilapplikasjoner

Discord: Verktøy for å få direkte kontakt med hverandre

Postman: Testing og utforskning av APIer

Firestore: Database tjeneste



## 5.2 Tidlig risikoanalyse

Risikoanalyse tabell:S				
Påvirkning				
Sannsynlighet		Lav(1)	Middels(2)	Høy(3)
	Lav(1)		(Nr. 2), (Nr. 3)	(Nr. 1), (Nr. 6)
	Middels(2)			(Nr. 4)
	Høy(3)		(Nr. 5)	(Nr. 7)

Tabell 1: Risikoanalyse tabell:

En tabell som viser til mulige gruppe hendelser, risikoer som er medlem relaterte.

Hendelsesbeskrivelser: GRUPPE			
Hendelser:		Risiko nivå	Begrunnelse
Nr. 1	Forfallsdato for levering repekteres ikke.	3	Fører til at masse tilbakelagt arbeid og muligens prosjekt feil.
Nr. 2	En gruppe medlem er ikke i stand til det å arbeid på grunn av personlige forhold	2	Vi er tre studenter med ulike timeplaner og det er mulighet for at prosjektarbeids timer overlapper med andre hendelser. Fører til potensielt masse arbeid som må gjøres på "overtid".
Nr. 3	En gruppe medlem blir syk	2	Selvom pandemien er for det meste "over" er dette noe som kan hende og føre til at prosjektet blir senket.
Nr. 4	Konflikt mellom gruppemedlemer	5	Lav sannsynlighet for at dette skjer, men det er fullt mulig ettersom vi har ikke jobbet med hverandre før og ble kjent med hverandre i dette faget. Konflikt kan føre til masse tid sløst på å diskutere problemer istedet for å løse de og masse sløsing av tid.

En tabell som viser til mulige prosjekt relaterte hendelser, risikoer som kan oppstå på grunn av oppgavens vanskelighetsgrad eller teknologi uregelmessigheter.

Hendelsesbeskrivelser: PROSJEKT/OPPGAVE			
Hendelser:		Risiko nivå	Begrunnelse
Nr. 5	Uventede programmeringsspråk vansker	5	2 av 3 på gruppa har ikke jobbet med Kotlin før og det kan føre til vansker å komme seg videre med prosjektarbeidet på grunn av dette. Det er stor sannsynlighet for at dette skjer og kan føre til at for masse tid blir brukt på å lære Kotlin enn på å komme seg videre med prosjektet.
Nr. 6	Applikasjoner/verktøy som brukes til arbeid blir utilgjengelige	3	Lav sannsynlighet med moderne verktøy, men muligheten finnes fortsatt. Kan føre til at gruppen må omorganisere arbeidet og kaste bort masse tid i prosessen.
Nr. 7	API svarer ikke	6	Prosjekt definerende, hvis dette ikke fungerer vil ikke vi ha mulighet å jobbe med prosjektet slik vi vill. Ofte vil ikke dette være et stort problem, men dette er en API vi fant selv og ikke noe vi fikk tildelt av skolen.

Tabell 2: Tabell for hendelsebeskrivelser

Hendelsesbeskrivelser: Risiko begrensningstiltak			
Hendelser:		Skadebegrensning	Resulterende risiko nivå
Nr. 1	Forfallsdato for levering respekteres ikke.	Med å følge Scrum arbeidsmetodikken; holde ukentlige statusmøter og definere vel organiserte spuirter for å sikre at arbeidet er fullført til gitt tid og at datoer respekteres. Dette vil redusere sannsynligheten for hendelsen skjer, men innvirkning forblir den samme, som er prosjektsvikt.	2
Nr. 2	En gruppe medlem er ikke i stand til det å arbeid på grunn av personlige forhold	Dette er ofte vanskelig å håndtere ettersom det er ofte utenfor vår kontroll, men vi skal forsøke vårt beste å planlegge rundt dette. Gode tiltak er å forsøke å gi beskjed god tid i forveien eller snarest mulig når en vet han ikke kan møte opp, holde notatene våre samlet slik at vi vil ikke være helt avhengig av en savnet gruppemeldem og organisere arbeidet slik at alle vet hva de andre jobber med slik at hvis en mangler så er ikke de andre hjelpeløse på oppgaven.	1
Nr. 3	En gruppe medlem blir syk	Går hånd i hånd med forrige punkt, men utgangspunktet er forskjellig. Igjen, gode tiltak er å forsøke å gi beskjed god tid i forveien eller snarest mulig når en vet han ikke	1

		kan møte opp, holde notatene våre samlet slik at vi vil ikke være helt avhengig av en savnet gruppemeldem og organisere arbeidet slik at alle vet hva de andre jobber med slik at hvis en mangler så er ikke de andre hjelpeløse på oppgaven.	
<b>Nr. 4</b>	Konflikt mellom gruppemedlemmer	Ettersom vi er relativt ukjente med hverandre blir dette et problem hvis noe slikt oppstår. Vi bør forsøke å holde en åpen dialog for å redusere risikoen for eskalering av konflikter noe som tillater medlemmer å se ting fra forskjellige perspektiver. Videre, ha en gruppeleder til ha det siste ordet, ettersom vi er en treergruppe og dette er den eneste måten å åpne majoritet på.	3
<b>Nr. 5</b>	Uventede programmeringsspråk vansker	2 av 3 gruppemedlemmer har aldri jobbet med Kotlin før og må bruke en stund for å vende seg inn syntaksen og programmerings paradigmen. Hoved tiltaket er at en av gruppemedlemmene har erfaring med Kotlin og kan hjelpe/veilede i tilfelle vi ikke kommer oss videre. Samtidig vil vi også jobbe litt utenfor pensum i PROG2007 - Mobilprogrammering slik at vi har nok kunnskap for å sette i gang programmerings delen.	3
<b>Nr. 6</b>	Applikasjoner/verktøy som brukes til arbeid blir utilgjengelige	Evaluerer av verktøyene brukt og "backup" verktøy er gode tiltak for å minimere påvirkningen av denne risikoen.	2
<b>Nr. 7</b>	API svarer ikke	Siden vi ikke har fått tildelt en "sikker" API av skolen er vi avhengig av at den API-en vi bruker er til å stole på. Likevel, siden muligheten for at API ikke fungerer lenger er tiltaket vi har valgt å bruke å finne en "backup" API. Dette kan likevel føre til arbeid som må gjøres om, er dette det beste vi kan gjøre. Sannsynligheten går da ned, men påvirkningen blir likevel stor.	4

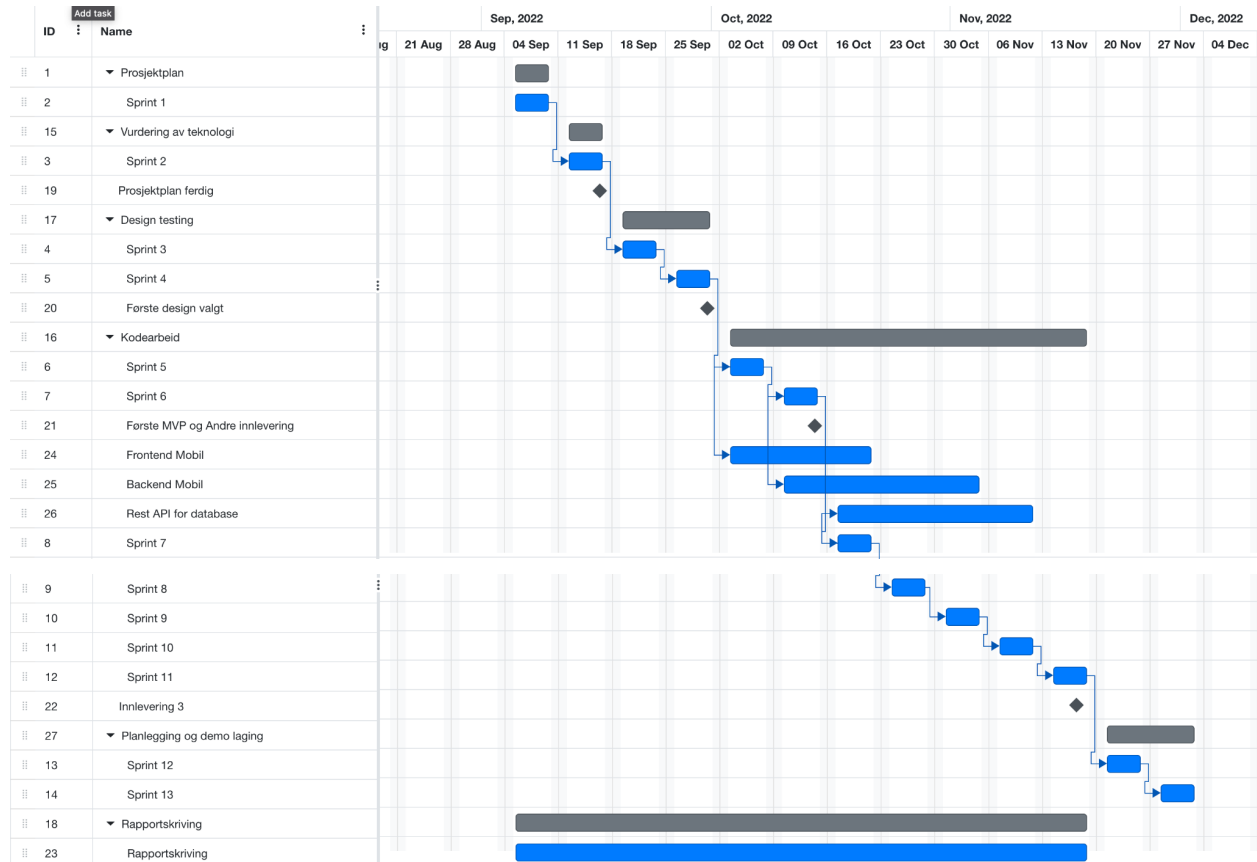
Tabell 3: Tabell for risikoreduserende tiltak

Risikoanalyse tabell:				
Påvirkning				
Sannsynlighet		Lav(1)	Middels(2)	Høy(3)
	Lav(1)	(Nr. 2), (Nr. 3)	(Nr. 1), (Nr. 6)	
	Middels(2)		(Nr. 5)	(Nr. 7)
	Høy(3)	(Nr. 4)		

Tabell 1: Risikoanalyse tabellen etter risikobegrensningstiltakene er tatt i bruk.

## 6.0 Fremdriftsplan for gjennomføring

### 6.1 Gantt-skjema for hele perioden



### 6.2 Product backlog

1. Lær mer om Kotlin
2. Lag wireframe modeller av applikasjon
3. Test modeller opp mot brukere
4. Formaliser bruker-UI
5. Hovedskjerm til applikasjon
6. Vis data som tilhører applikasjoner
7. Åpne applikasjon fra applikasjonen
8. Få app til å snakke med databasen
9. Lag Rest API for databasen
10. Uteplasser Rest API for databasen i skyen
11. Test applikasjon
12. Test Rest API
13. Lag demo
14. Skriv rapport