

Strikke Appen Rapport

Av Kim-Andre Engebretsen

| | |
|--|-----------|
| 1. Hva er det som er lagd | 2 |
| 2. Hvorfor | 2 |
| 3. Brukerveiledning | 2 |
| 3.1 Åpne et lagret prosjekt | 3 |
| 3.1.1 Åpne prosjekt og dele det | 4 |
| 3.2 Start et nytt prosjekt | 5 |
| 3.2.1 Generer strikkemønster med et bilde tatt med kameraet | 6 |
| 3.2.2 Generer strikkemønster med et bilde på enheten | 7 |
| 3.2.3 Mønsteret blir generert | 8 |
| 3.2.4 Hent prosjekt fra nett | 9 |
| 4. Appens oppbygging | 10 |
| 4.1 Oppbyggingen | 10 |
| 4.2 Klasser | 12 |
| 4.3 Biblioteker | 13 |
| 4.4 Design | 13 |
| 5. Arbeidsprossesen underveis | 14 |
| 5.1 Teknisk | 14 |
| 5.2 Design | 15 |
| 6. Skrytliste | 18 |
| 6.1 Generering av Strikkemønster | 18 |
| 6.2 Deling og nedlasting med Firebase Database. | 19 |
| 7. Hva fikk jeg hjelp til | 20 |
| 8. Hva ble ikke ferdig / kunne blitt gjort annerledes | 21 |
| 8.1 Hva ble ikke ferdig | 21 |
| 8.2 Hva kunne blitt gjort annerledes | 21 |
| 9. Videreutvikling | 22 |
| 10. Prototype | 22 |
| 11. Konklusjon | 26 |
| Referanser | 26 |
| Kilder | 26 |

1. Hva er det som er lagd

Det ferdige produktet: StrikkeAppen, er kort sagt en app for å lage strikkemønstre. Med appen kan brukere lage et strikkemønster ut av hva som helst enten det er et bilde tatt med kameraet eller det er et bildet som ligger på telefonens mapper. Deretter kan bruker velge å lagre mønsteret på sin telefon etter at det har blitt generert. Brukeren kan også velge bredde og høyde på strikkemønsteret, noe som er viktig for en som strikker siden da kan man beregne hvor mye garn man kommer til å bruke. Hvis brukeren synes strikkemønsteret de har laget har fungert bra til et prosjekt eller synes det er stilig så kan de dele strikkemønsteret (lagre det til et database) så andre brukere kan laste det ned til sin enhet og bruke det til sitt eget prosjekt.

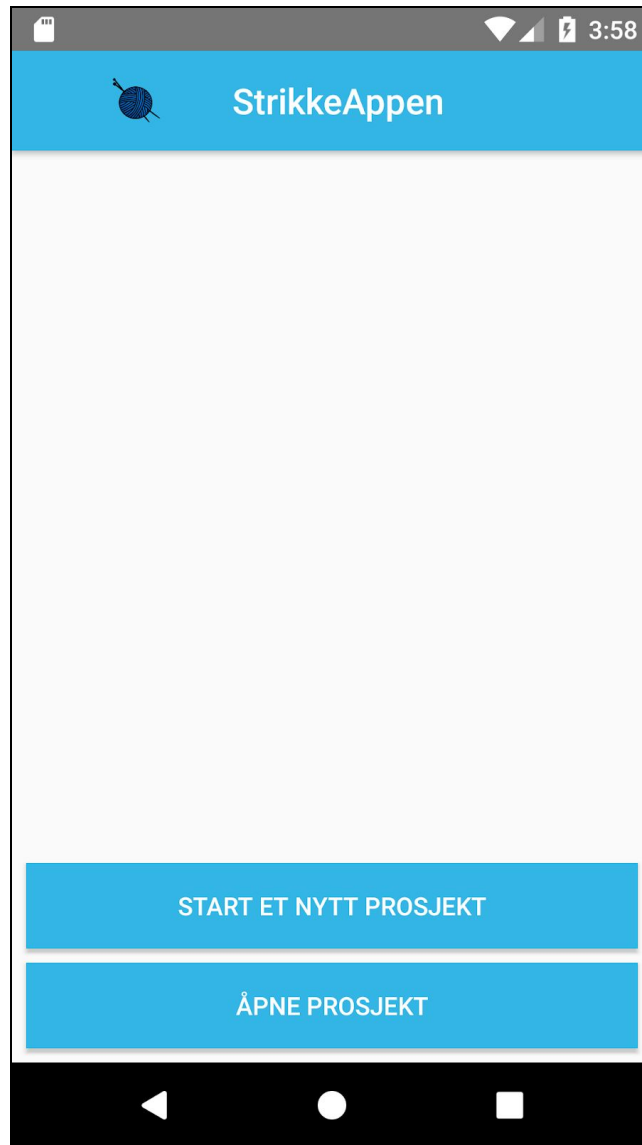
2. Hvorfor

Å lage et strikkemønster manuelt i dag er kjedelig, vanskelig og tidskrevende, og det finnes ikke mange lignende apper per dags dato som egner seg spesielt til å lage strikkemønstre for strikkere. Har så langt ikke funnet noen lignende apper som kan lage et strikkemønster utifra et bilde som også er bakgrunnen for forslaget til prosjektet. Forslaget kommer fra min søster som strikker daglig og som har strikket en god del gensere og luer, hun tenkte seg en app som kunne ta seg av prosessen med å tegne et strikkemønster utifra et bilde som enten er tatt med kamera eller lastet ned.

3. Brukerveiledning

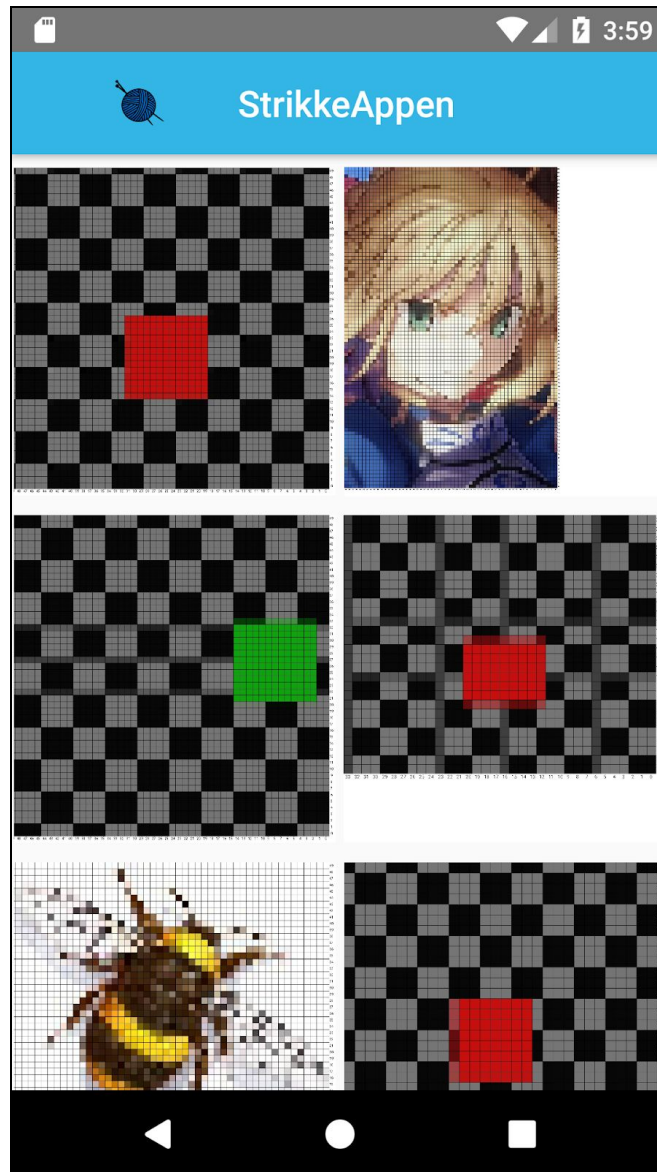
Brukeren åpner appen og får to valg:

1. Åpne et lagret prosjekt
2. Start et nytt prosjekt



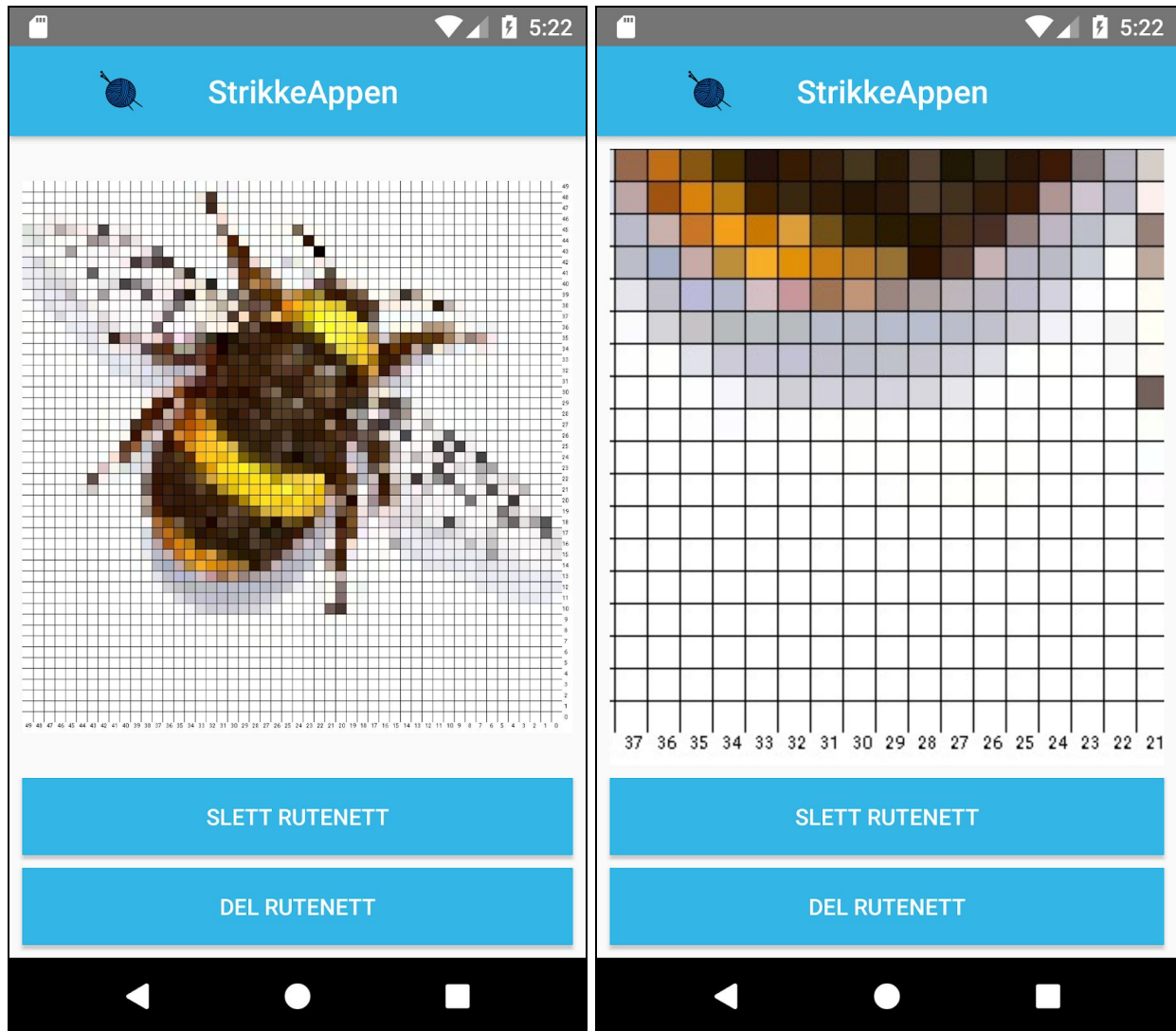
3.1 Åpne et lagret prosjekt

Brukeren får opp alle strikkemønstre som er lagret på enheten og kan velge hvilke de vil fortsette med som de ikke er ferdig med, eller som de skal bruke på nytt ved å trykke på det ønskede strikkemønsteret.



3.1.1 Åpne prosjekt og dele det

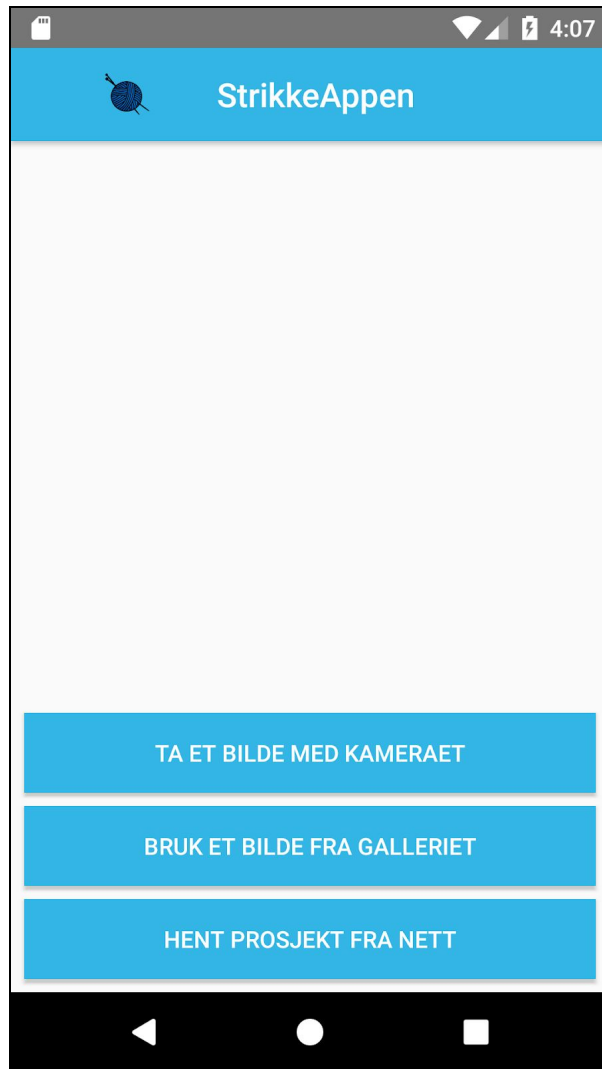
Når brukeren har trykket på et strikkemønster så får man opp strikkemønsteret i tillegg til to knapper. Ved å trykke på "Del rutenett" laster opp strikkemønsteret til en database så alle andre brukere kan få tilgang til mønsteret og laste det ned til sin enhet. Ved å trykke på "Slett rutenett" så slettes strikkemønsteret fra enheten. Her kan man også zoomme inn på strikkemønsteret slik at man kan bruke koordinatene til orientere seg rundt i mønsteret.



3.2 Start et nytt prosjekt

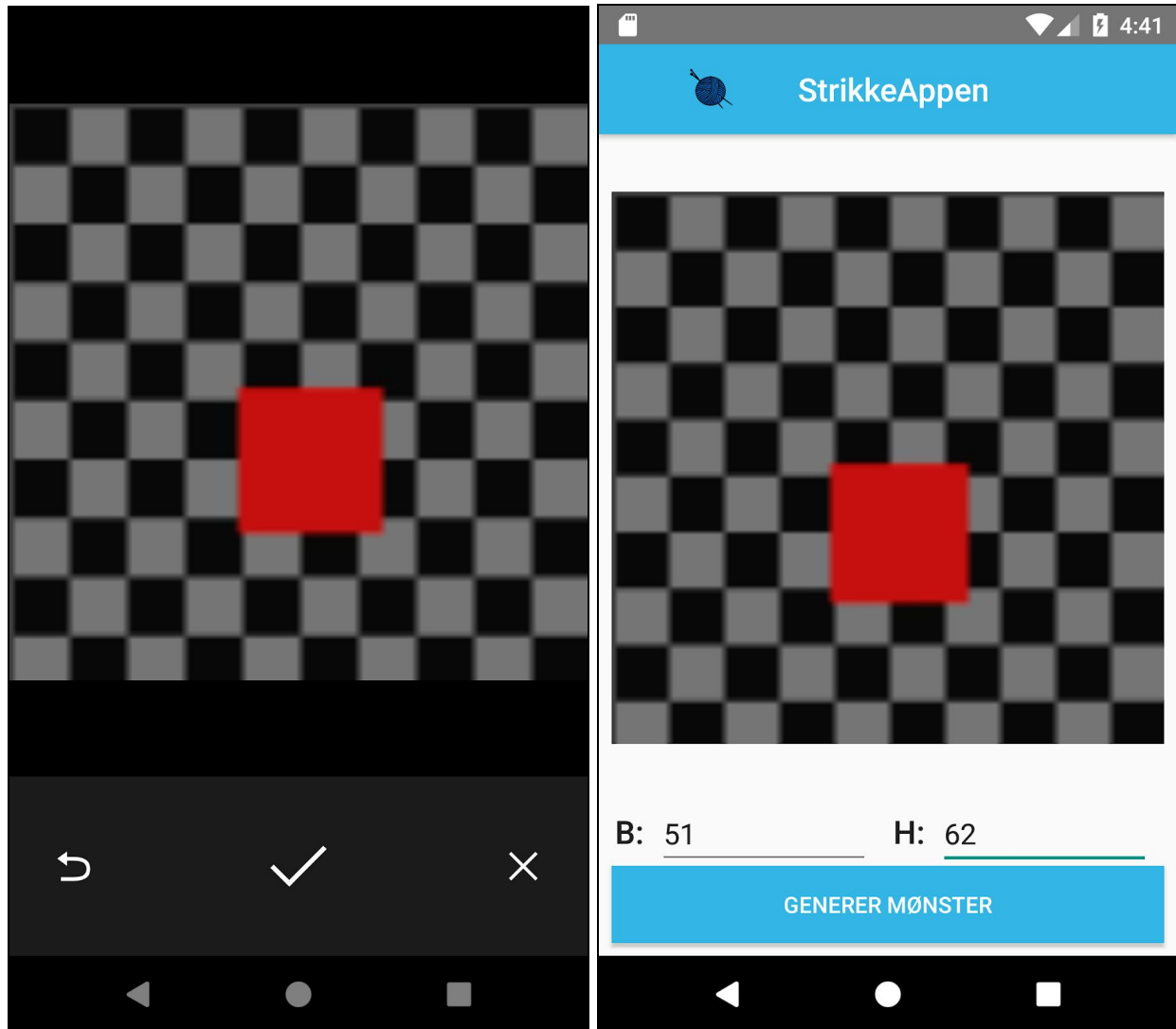
Hvis brukeren velger å starte et nytt prosjekt så får de 3 nye valg:

- 1. Ta et bilde med kameraet
- 2. Bruke et bilde fra galleriet / image mapper på telefonen
- 3. Hent prosjekt fra nett



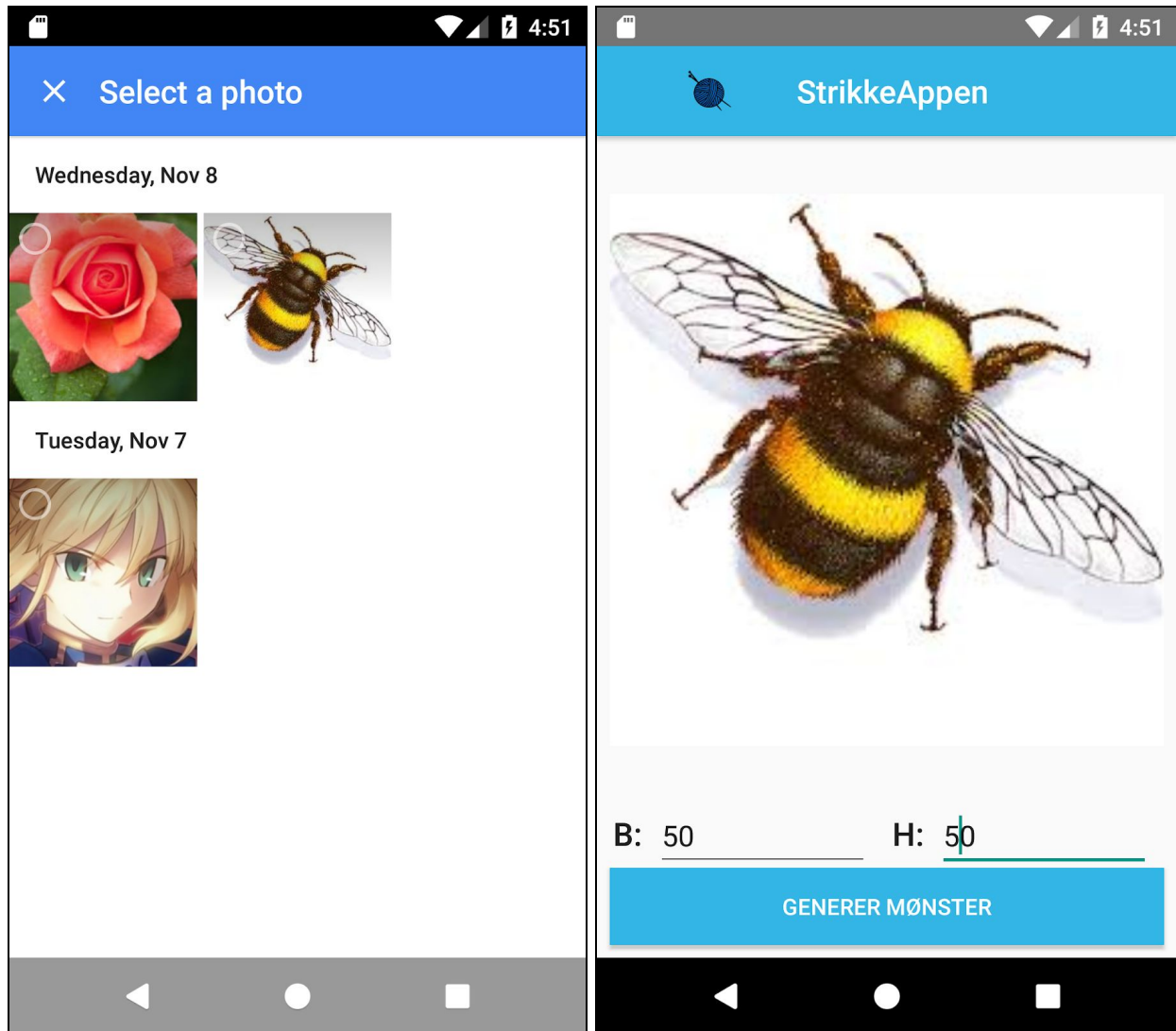
3.2.1 Generer strikkemønster med et bilde tatt med kameraet

Brukeren kan ta et bildet med kameraet som skal bli generert til et strikkemønster. Første gang appen kjøres så vil appen spørre om tillatelse til å bruke kameraet. Hvis ikke tillatelses vinduet kommer opp, forsøk å vente et par sekunder eller trykk på tilbake knappen for å se om den kommer opp.



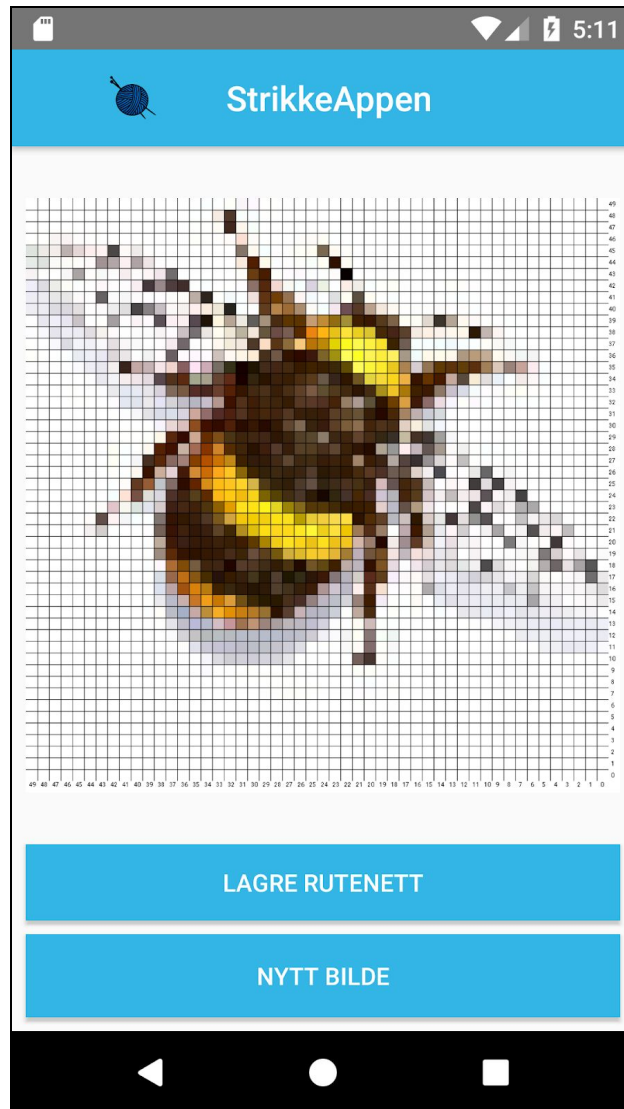
3.2.2 Generer strikkemønster med et bilde på enheten

Brukeren kan velge et bildet som ligger på enhetens mapper og generere et strikkemønster utifra det. Første gang appen kjøres så vil appen spørre om tillatelse til å bruke et bilder på enhetens mapper. Hvis ikke tillatelses vinduet kommer opp, forsøk å vente et par sekunder eller trykk på tilbake knappen for å se om den kommer opp.



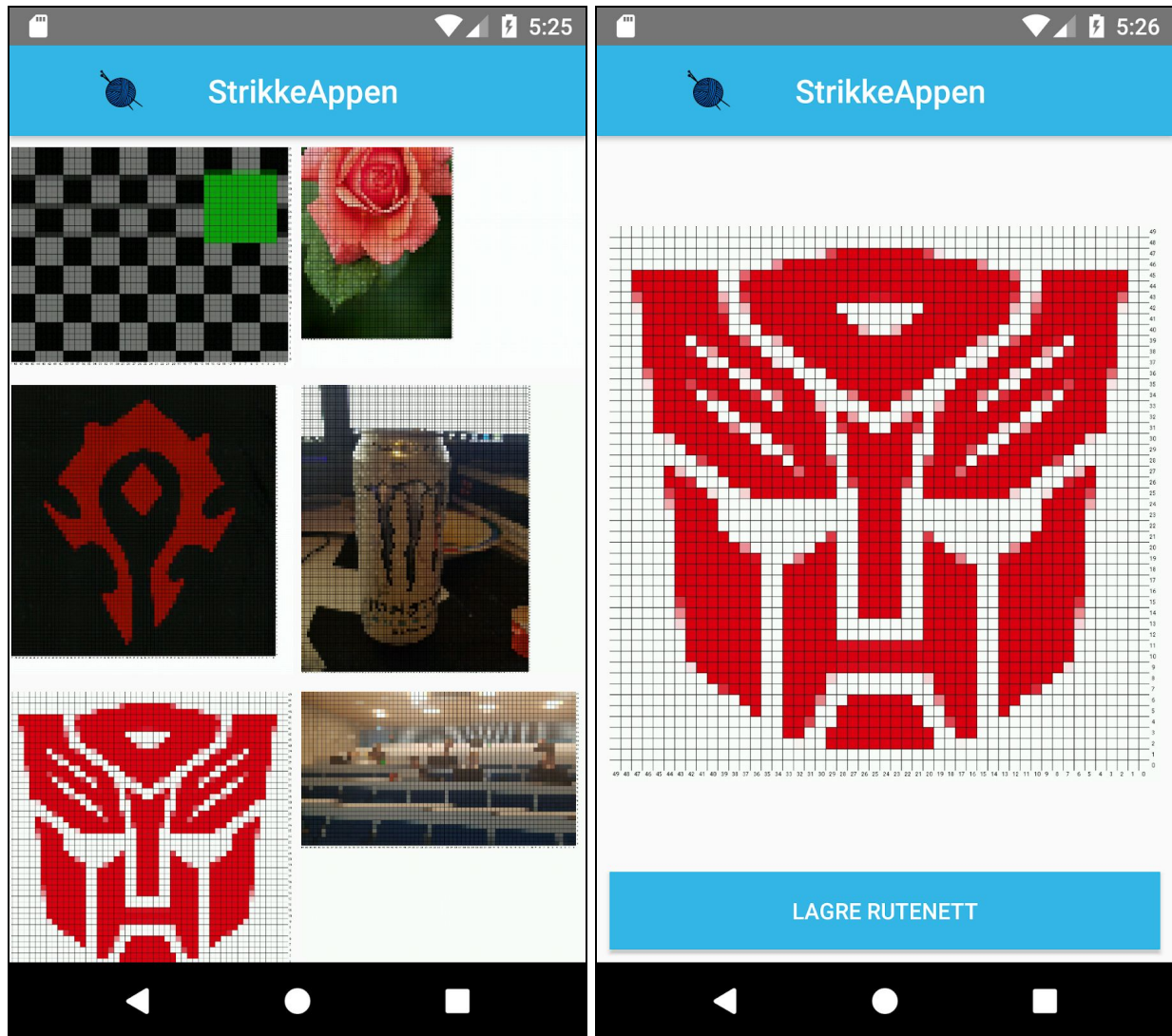
3.2.3 Mønsteret blir generert

Etter at brukeren har valgt et bilde eller tatt et bilde med kameraet og trykket på generer mønster, så kommer brukeren til neste aktivitet hvor man kan velge om man vil lagre strikkemønsteret eller velge et nytt bilde.



3.2.4 Hent prosjekt fra nett

Når brukeren trykker på "Hent prosjekt fra nett" så får brukeren opp en liste over alle strikkemønstre som er delt av andre. Brukeren kan trykke på et mønster hvor mønsteret kommer opp i en ny aktivitet med en knapp. Når brukeren trykker på knappen blir mønsteret lagret til enheten.



4. Appens oppbygging

4.1 Oppbyggingen

Se Klasser for referanse.

- Startup. Brukeren kommer hit ved oppstart av appen (Main Activity). Har 2 knapper:
 - Åpne prosjekt knapp → kobler brukeren videre til SavedProjects.
 - Her sjekkes det også om mappen som inneholder strikkemønstrene finnes eller er tom. Hvis det er tilfelle så si ifra til bruker.
- Start et nytt prosjekt knapp → kobler brukeren videre til NewProject.

- NewProject, Brukeren velger å starte et nytt prosjekt. Har 3 knapper:
 - Ta et bilde med kameraet knapp → kobler brukeren videre til CameraView med en boolean verdi lik True som initierer at kameraet skal brukes. Denne boolean verdien er false som default → brukeren tar deretter et bilde og velger høyde og bredde på strikkemønsteret som skal genereres → brukeren trykker på “Generer mønster” knappen og bildet blir generert til et rutenett med PixelGridView → brukeren kommer deretter til → NewProjectGenerated.
 - Bruk et bilde fra galleriet knapp → kobler brukeren videre til CameraView, brukeren velger et bilde fra enhetens mapper og velger høyde og bredde på strikkemønsteret som skal genereres → brukeren trykker på “Generer mønster” knappen og bildet blir generert til et rutenett med PixelGridView → brukeren kommer deretter til → NewProjectGenerated.
 - Hent prosjekt fra nett knapp → kobler brukeren videre til SharedProjects.
- SharedProjects → brukeren får opp alle strikkemønstre som ligger på Firebase databasen → brukeren kan trykke på et strikkemønster som kobler brukeren videre til → SaveProjectFromDatabase.
- SaveProjectFromDatabase → Strikkemønsteret som brukeren trykket på pluss en knapp (“Lagre rutenett”) blir vist. Ved å trykke på knappen lagres strikkemønsteret til enhetens SD kort. → brukeren kommer tilbake til StartUp.
- PixelGriedView, Bruker en 2D array med fargekoder til å lage et strikkemønster med canvas.
- SavedProjects
 - Brukeren kan trykke på et strikkemønster → når brukeren trykker på et strikkemønster kommer brukeren videre til OpenProject.
- NewProjectGenerated. Har 3 Knapper
 - Lagre rutenett → Det genererte strikkemønstre blir lagret til enhetens SD kort → brukeren blir deretter sendt tilbake til StartUp.
 - Nytt bilde → kobler brukeren tilbake til CameraView → brukeren tar deretter et nytt bilde eller velger et nytt bilde fra galleriet avhengig av boolean verdien i CameraView som styrer på om kameraet er brukt eller ikke → brukeren angir høyde og bredde → brukeren trykker på “Generer mønster” knappen og bildet blir generert til et rutenett med PixelGridView → brukeren deretter tilbake igjen til NewProjectGenerated.
- OpenProject

- Strikkemønsteret som brukeren trykket på pluss to knapper; "Slett rutenett" og "Del rutenett" blir vist. Ved å trykke på "Slett rutenett" knappen lagres slettes strikkemønsteret fra enhetens SD kort. Deretter sjekkes det om enheten inneholder noe flere strikkemønsteret, hvis ikke blir brukeren koblet til StartUp. Hvis det fortsatt er strikkemønsteret lagret så blir brukeren koblet tilbake til SavedProjects.→ brukeren kommer tilbake til StartUp.

4.2 Klasser

- StartUp
 - Med andre ord: appens main activity, det er her brukeren kommer når appen starter og er default. Inneholder 2 buttons for: Åpne prosjekt og Start et nytt prosjekt.
- NewProject
 - Brukeren kommer til en aktivitet hvor det er 3 buttons for:
 - Ta et bildet med kamera
 - Bruk et bilde fra Galleriet
 - Hent et prosjekt fra nett.
- SavedProjects
 - Viser alle prosjekt som brukeren har lagret. Brukeren kan trykke på et strikkemønster for å velge om de vil bruke det.
- OpenProject
 - Her skal bare et rutenett som brukeren har valgt i SavedProjects vises.
 - Brukeren kan dele strikkemønsteret ved å trykke på "Del rutenett" knappen.
 - Brukeren kan slette strikkemønsteret ved å trykke på "slett rutenett" knappen.
- SharedProjects
 - Viser alle prosjekt som andre brukere har delt, alle strikkemønstra som er lagret i firebase databasen listes opp. Brukeren kan trykke på et strikkemønster for å velge om de vil lagre det.
- SaveProjectFromDatabase
 - Her skal bare et rutenett som brukeren har valgt i SharedProjects vises.
 - Brukeren kan lagre strikkemønsteret ved å trykke på "Lagre rutenett" knappen.
- CameraView
 - Brukeren kommer til en aktivitet hvor enten bildet som er tatt med kameraet, eller bildet valgt på enhetens mapper vises. Her velger brukeren høyde og bredde på rutenettet.

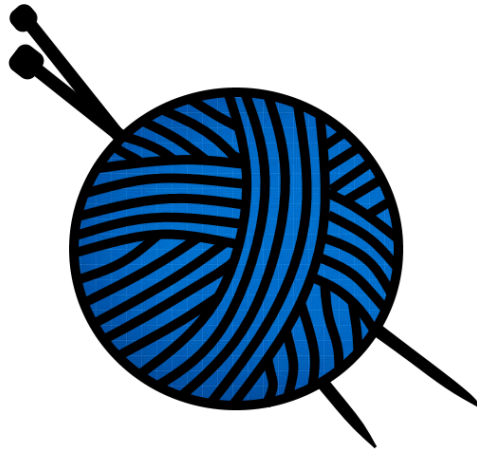
- Når brukeren har valgt en høyde og bredde kan de trykke på “Generer mønster” knappen som genererer et strikkemønster fra bildet som er valgt.
- PixelGridView
 - En klasse som bruker en 2D array med fargekoder til å tegne et strikkemønster med canvas.
- NewProjectGenerated
 - Brukeren får opp det genererte strikkemønstre med de paremetre som er angitt i CameraView. Brukeren får her 2 valg: Enten lagre strikkemønsteret til enheten eller velge et nytt bilde.
- ImageAdapter
 - Adapter som brukes til et gridview i SavedProjects, lister opp alle strikkemønstere som er lagret på enheten.
- SharedImagesAdapter
 - Adapter som brukes til et gridview i SharedProjects, lister opp alle strikkemønstere som finnes i firebase databasen.
- Uploads
 - Brukes til å lage objekter som holder på en URL.

4.3 Biblioteker

Det ble benyttet en rekke biblioteker til å lage appen, mer enn det jeg trudde jeg kom til å trenge. Det blir forøvrig benyttet jeg et ekstern bilbiotek kalt Chris Banes PhotoView som blir brukt til å få til zoominga på strikkemønstrene, kommer tilbake til dette i del 7: “Hva trengte jeg hjelp til”. Link til Chris Banes PhotoView finnes nederst i rapporten i kilder under referanser.

4.4 Design

Det ble ikke tenkt så mye på selve designen av appen og det ble valgt å utsette det til appens hovedfunksjoner var på plass og appen begynte å ta form. Det ble ikke satt av så mye tid på googles design patterns og standardiserte måter å designe en app siden den gjør ikke så mye mer enn å generere strikkemønstere, lagring og deling og dermed består kun av noen knapper og gridviews. Logoen til appen laget jeg selv i paint.net og toolbaren følger en standard design; nemlig at den skal være 4dp “elevated” som knappene er.



5. Arbeidsprossesen underveis

5.1 Teknisk

Det ble i starten tenkt at all lagring på appen skulle bli lagret på firebase storage, slik at det ikke var behov for å måtte lage en egen mappe til appen på mobilen. Men det ble bestemt at ingen brukere skal bli tvunget til å måtte logge inn eller må gjennom prosessen med å lage for eksempel en google+ konto fordi det er ikke alle som har en google+ konto.

En stor utfordring var å finne ut hvordan appen skulle klare å generere et strikkemønster, skulle det bli benyttet gridview? recyclerview? layoutview? Osv... Det ble gjort en god del research for hvordan jeg måtte da gå fram for å benytte for eksempel et gridview til å lage et strikkemønsteret men det så komplisert ut og jeg skjønnte ikke hvordan dette ville fungere når en bruker av appen skal lagre eller dele strikkemønsteret. Etter en veiledning med student assistent Adrian Jensby Sandaker så hadde han kommet over et innlegg på stackoverflow som benyttet canvas til å tegne et rutenett. Jeg fant ut canvas kan også bli benyttet av bitmap til å lagre bilde-filer så da ble det bestemt at jeg skulle bruke canvas til å få til strikkemønster genereringen.

I appen så går det an å dele strikkemønstre med andre, dette er da gjort ved at strikkemønsteret blir lastet opp og lagret på en database. Dette skulle egentlig gjøres i cameraview rett etter at brukeren har generert et rutenett og har valget mellom å lagre til enheten eller ta bilde på nytt, men jeg valgte å fjerne den muligheten fra cameraview og heller ha det som et valg når man åpner et strikkemønster som allerede er laget. Dermed må man først generere et mønster og lagre det til mobilen før man kan dele det, noe som gir mer mening.

Jeg visste tidlig i prosjektet at brukeren av appen må kunne zoome in og ut på strikkemønstra, det som da først ble tenkt på var å bruke et recyclerview. Til recyclerviewet så skulle det da være en knapp som når ble trykket så zoomet den inn ved å kutte øverste rad og kolonne lengst bort og forstørre rutene litt. Som egentlig er en god ide men veldig kronglete, og da kommer man til det problemet igjen; hvordan skal dette lagres og deles? Jeg valgte å droppe det og heller finne noe annet, det var da jeg kom over et eksternt bibliotek som heter Chris Bane's Photoview som er et eget imageview som man kan zoome inn på. Det ble da avgjort å gå videre med canvas ideen og ha de genererte strikkemønstra i et photoview.

Jeg hadde også et par problemer underveis med at bildet som skulle genereres ble rotert, måtte da finne en metode for å snu og rotere på bildet så det stemte.

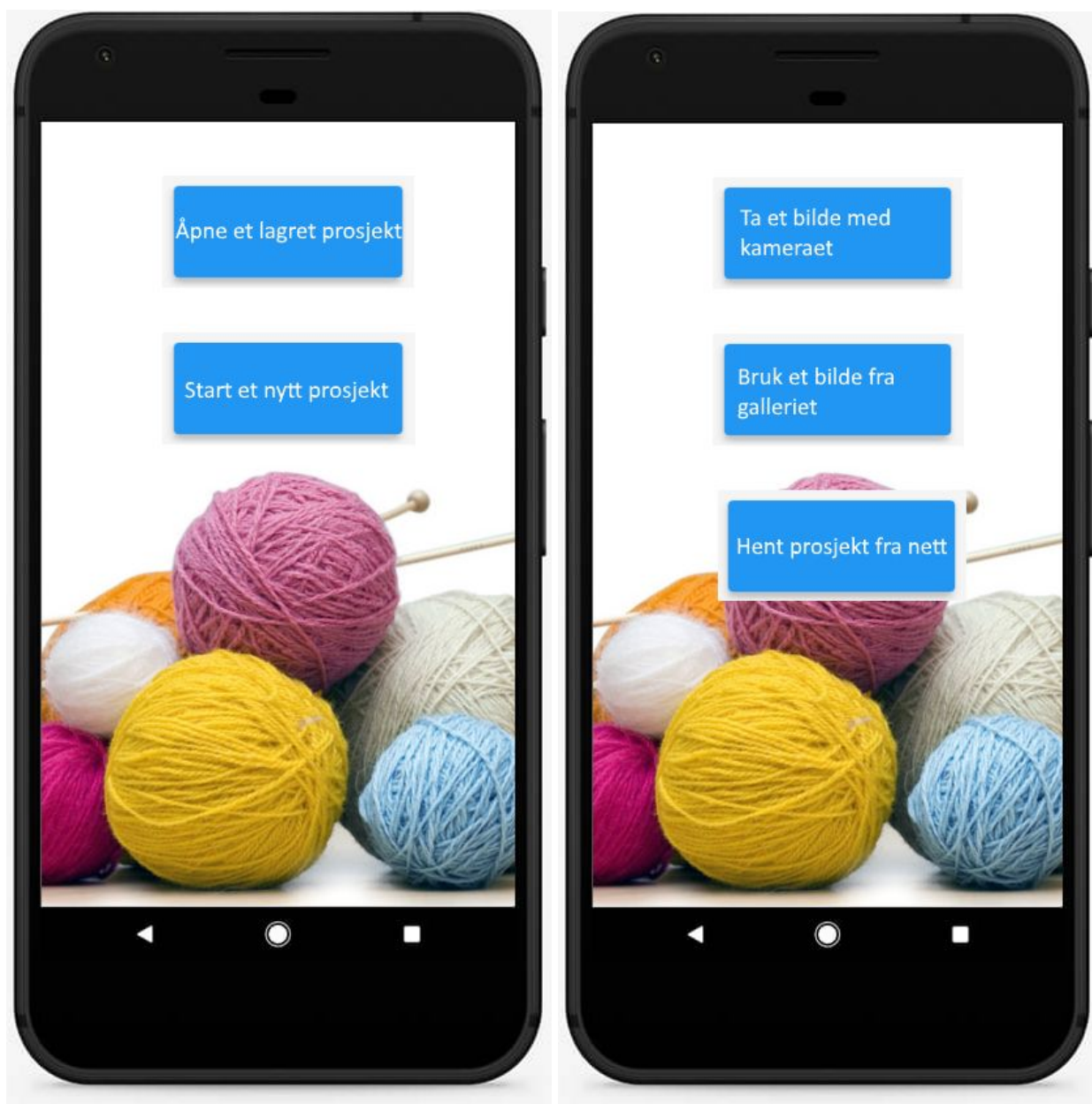
Etter presentasjonen den 24.11.2017 fant jeg ut at jeg ikke hadde så mye tid til å implementere flere funksjoner på appen, men jeg tenkte at det var lurt å ha en knapp som kan slette de strikkemønstre som er lagret på enheten. Dermed ble det implementert i appen at man kan slette strikkemønstre, dette ble gjort etter presentasjonen på skolen.

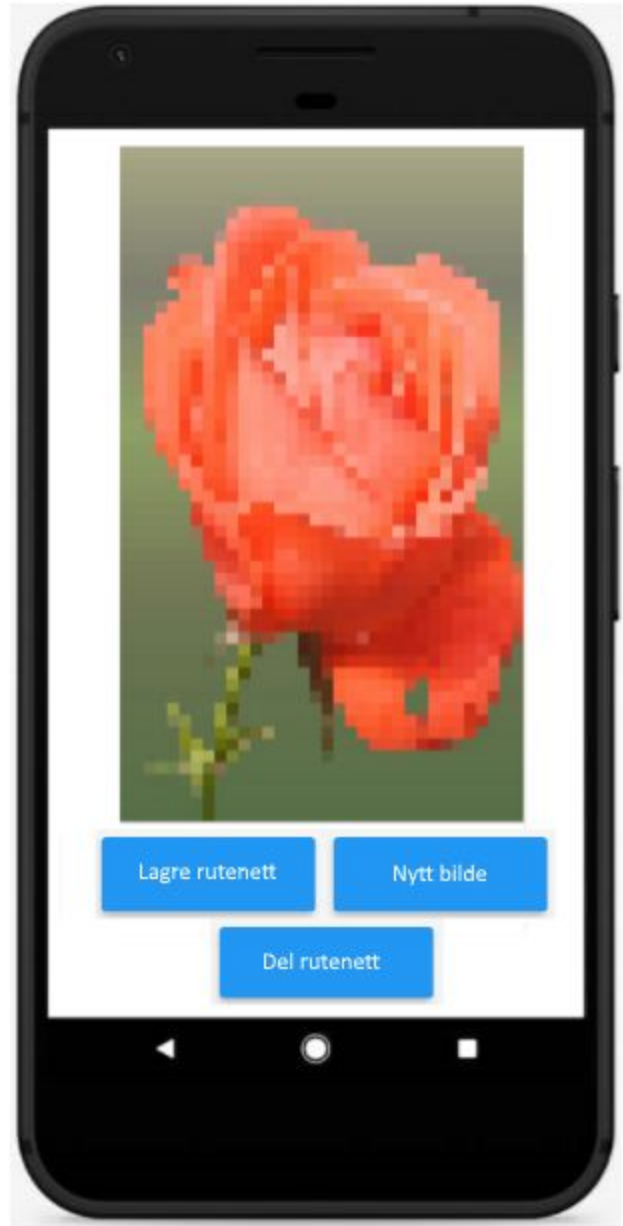
Det ble også justert på oppløsningen til strikkemønstrene, slik at oppløsningen ikke fulgte en standard verdi som jeg hadde satt men ble beregnet utifra høyden eller bredden på strikkemønstrene. Når jeg brukte en standard verdi så ble koordinatene for små jo større høyden og bredden på rutenettet ble, dette kommer jeg tilbake til i kapittel 6. Skryteliste. Det var da jeg i tillegg bestemte at bruker ikke kan lage et strikkemønster som har flere enn 100x100 ruter. Siden det er lite sannsynlig at noen strikker noe større, og jo flere ruter mønsteret skal innholde jo større blir bilde-filen. Et 100x100 strikkemønster vil ha en oppløsning på 2000x2000 som er stort nok i seg selv.

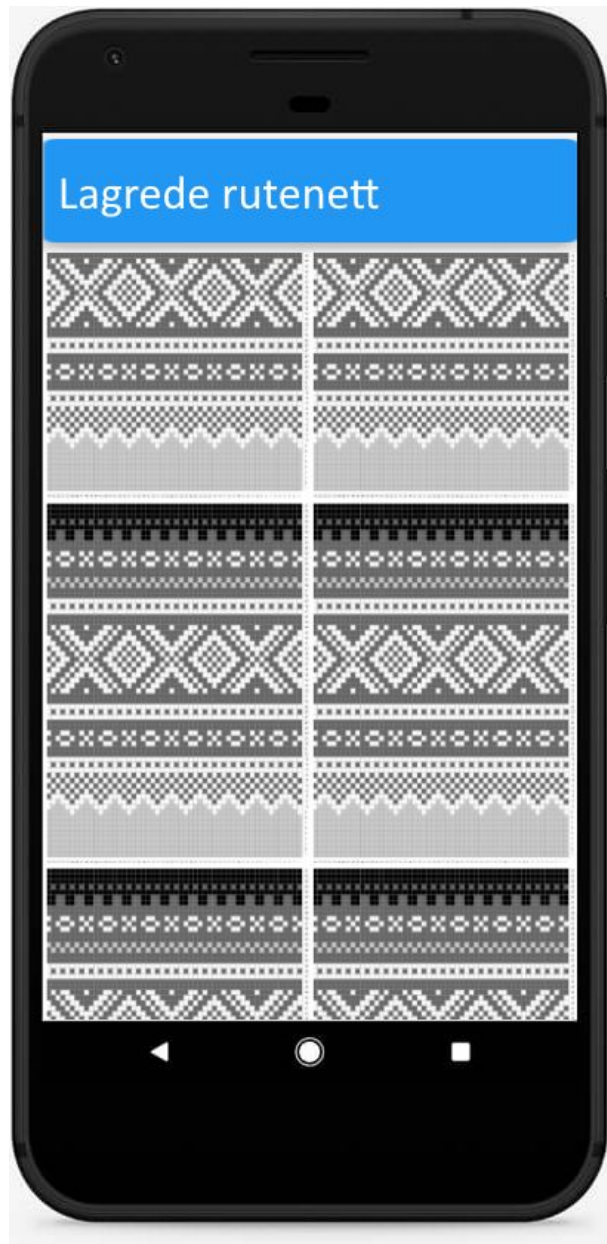
I sluttfasen med bug fiksing så ble det lagt til litt flag håndteringer slik at backstacken blir "clearet" etter at en bruker har slettet et strikkemønster og/eller etter at bruker har generert og lagret et strikkemønster. Det ble i tillegg lagt inn en permissionrequest når brukeren skal hente et strikkemønster som ligger på firebase databasen siden mønsteret blir lagret på enhetens mapper.

5.2 Design

Det ble gjort et par endringer på selve utseende av appen underveis, men noe ble det samme. Under finner man bilder som jeg lagde for "Detlajert prosjekt design" innleveringen, som man kan se sammenlignet med screenshotsa i brukerveiledningen så er det mye som ligner men det var ting jeg justerte på. Jeg valgte også å droppe å ha et bakgrunnsbilde.







6. Skryteliste

Her snakker jeg litt om de funksjonene som jeg vil skal vektlegges mer enn de andre i vurderingen. Det var disse funksjonene som tok lengst tid og som jeg er mest fornøyd med.

6.1 Generering av Strikkemønster

Etter at bruker har valgt et bilde og definert ønsket høyde og bredde i CameraView blir det laget et Bitmap som benytter bildet skalert til den nye høyden og bredden som brukern har satt. Deretter blir bitmappet flippet og rotert så den stemmer overens med bildet som bruker har

valgt. Dette bitmappet blir så sendt videre til NewProjectGenerated. I NewProjectGenerated blir det initiert en dobbel for-løkke som går gjennom hvert pixel på bitmappet og lagrer fargekoden til pixelen i en 2D array. Deretter initieres det et nytt PixelGridView. Før et strikkemønster blir generert så blir det definert hvor store rutene i strikkemønsteret skal være, dette har jeg satt til å være 20 ganget med bredden hvis den er større enn høyden, ellers 20 ganget med høyden. Det er fordi det blir også generert koordinater, og font størrelsen til disse koordinatene er lik rutens størrelse. Dette ble lagt inn senere i prosjektet, før så var alle genererte strikkemønstre standard 1000x1000 (selv størrelsen på bildet som inneholder det genererte strikkemønsteret) så hvis man for eksempel lagde et strikkemønster som var 80 i bredde og 80 i høyde så ble koordinatene for små til se hva de var. Etter at størrelsen på rutene er blitt initiert så blir antall kolonner og rader definert i PixelGridViewet før det benytter 2D arrayen med fargekoder til å lage strikkemønsteret i canvas. PixelGridViewet bruker en dobbel for-loop til å gå gjennom 2D arrayen og tegne ruter med fargekodene i arrayen. Deretter blir det tegnet linjer vannrett og loddrett før det blir lagt på koordinater langs den nederste raden og langs kolonnen helt til høyre. Dette blir gjort med canvas. Det ferdig genererte strikkemønsteret blir konvertert til et bitmap som blir brukt til å vises i et Photoview og/eller bli lagret som en png fil.

6.2 Deling og nedlasting med Firebase Database.

Firebase blir benyttet til database løsningen for dette prosjektet. Det var ganske rett fram i starten å lagre strikkemønstrene til databasen ved å følge en Firebase tutorial som man finner i Tools på Android Studio. Prøvde å laste opp strikkemønstrene til databasen og det gikk fint. Men problemet jeg støttet på var å liste opp alle strikkemønstre som finnes i databasen i et gridview UTEN å måtte laste ned alle til enheten først. Jeg fulgte en tutorial som løste akkurat dette problemet, kilde til denne tutorialen finnes under referanser nederst i rapporten. Det blir først laget en Uploads klasse som brukes til å lage objekter som holder på en URL. Disse objektene blir laget når man deler et strikkemønster og blir lagret i databasen. Siden min App ikke krever innlogging for å få tilgang til delte strikkemønstre og dele strikkemønstre så måtte jeg justere litt på reglene i Firebase slik at alle har tilgang til filene som ligger der. Når man da trykker på "Hent prosjekt fra nett" knappen så hentes ut alle download URLene fra Uploads objektene på databasen og blir lagret i en arrayList. ArrayListen blir brukt i et adapter som gridviewet i aktiviteten bruker, i adapteret så brukes download URLene sammen med Glide (et eksternt bibliotek) til å vise strikkemønstrene i imageviews som blir satt inn i gridviewet. Når Brukeren trykker på et imageview i gridviewet så blir man koblet til "SaveProjectFromDatabase" sammen med download URLen til strikkemønsteret som er vist i imageviewet. I "SaveProjectFromDatabase" aktiviteten så kan brukeren lagre strikkemønsteret til sin enhet ved å bruke download URLen som peker til strikkemønsteret i Storage.

7. Hva fikk jeg hjelp til

Når jeg satt fast eller slet med å finne ut åssen jeg skulle få til den funksjonen som jeg hadde tenkt meg så var det første jeg gjorde å google hvordan jeg skulle få det til, og som regel fantes det svar på spørsmålet. Mye av koden i prosjektet er “copy paste” som jeg har “tweaket” til å fungere i mitt prosjekt. Jeg brukte derimot ikke så mye av ressursene som lå på fagets hjemmeside (hiof canvas) eller studass for den saks skyld. Som nevnt tidligere så fikk jeg tips av studass med en link til et innlegg på stackoverflow for hvordan jeg skulle utføre mønster genereringen. Jeg sendte også en mail til faglærer for å spørre om hjelp til håndtering av regler i Firebase. Det er en god del linker til kilder som jeg har benyttet inni koden på prosjektet, jeg kommer også til å liste opp alle kildene jeg har brukt under referanser nederst i rapporten.

Her er en grovliste over hva jeg har trengt hjelp til / benyttet kode fra med et nummer som peker til linker som ligger under kilder under referanser nederst i rapporten:

1. Java, konvertere image til Buffered Image.
2. Hvordan fører jeg over verdier mellom activities i android
3. Flippe et bitmap
4. Hvorfor blir et bilde tatt med kamera rotert på noen enheter
5. 2D array grid i Canvas
6. Lagre bilder til Firebase
7. Hvordan zoome inn og ut på et bilde (Chris Bane's PhotoView)
8. Konvertere view til et Bitmap
9. Hvordan lagre et bitmap som en png fil til enheten
10. Hvordan få nåværende tid og dato
11. Hvordan slette en fil på enheten
12. Hvordan lage en “Ja” eller “Nei” dialogbox
13. Hvordan legge til bilder i et gridview
14. Hvordan liste opp bilder i et gridview med hjelp av et adapter
15. Hvordan laste ned en fil fra firebase og lagre det på enheten
16. Hvordan liste opp alle bilder fra firebase uten å måtte laste de ned (Tutorial)
17. Hvordan lage en array med bilder
18. Hvordan sette opp en toolbar
19. Hvordan løse Unable to decode stream: java.io.FileNotFoundException
20. Hvordan ta et bilde med kamera og vise det i activity
21. Hvordan hente et bilde fra galleriet og vise det i activity
22. Sjekke om input i textview er et nummer eller ikke
23. Hvordan “cleare” backstacken
24. Hvordan få dimensionen til skjermen i pixler

8. Hva ble ikke ferdig / kunne blitt gjort annerledes

8.1 Hva ble ikke ferdig

Når jeg begynte å planlegge hvordan appen skulle se ut så satte jeg opp en prioritetsliste over funksjoner som: Måtte være med, Bør være med og Kan være med:

Her er en oversikt over den listen:

Må:

- Kunne ha tilgang til kameraet.
- Kunne ha tilgang til galleriet / image mappe på enheten.
- Kunne klare å pixelere et bilde og sette hver og en pixel inn i et rutenett, bruker må også kunne zoome inn og ut i rutenettet ellers blir det umulig å bruke store og detaljerte rutenett.
- Kunne lagre rutenett, for senere bruk eller lignende
- Kunne ha tilgang til lagrede rutenett

Bør:

- La brukeren kunne settet høyde og bredde på rutenettet
- La brukeren velge spesifikt hvilke farge som skal være i rutenettet
- La brukere dele rutenett med andre brukere via nettet og lagre det andre har delt.

Kan:

- Siden det ikke finnes garn i alle mulige farge, så skal rutenettet bare inneholde farger som man kan få kjøpt i butikker.
- I stedet for pixeler så skal det være symboler i rutenettet. Hvert symbol skal representere en farge.
- Brukeren kan klippe ut en del (crop) av bildet som skal bli til et rutenett, for eksempel at brukeren skal kunne lage mønster av et objekt på bildet og ønsker ikke å ta med det andre.

Så langt så inneholder appen alle funksjonene som måtte være med og burde være. Det er da funksjonene som "Kan være med" som det ikke var tid nok til å fullføre.

8.2 Hva kunne blitt gjort annerledes

Foreløpig så fungerer appen slik at når man lagrer et strikkemønster så blir det lagret på enhetens SD kort. Ikke alle har SD kort på sin enhet. Så det jeg ville gjort annerledes er at i prosessen hvor man lagrer et strikkemønster så sjekker man om enheten har et SD kort installert. Hvis ikke enheten har et SD kort så skal det lagres på telefonens interne minne i stedet. Det burde også vært tester som sjekker at det faktisk er plass til å lagre strikkemønsteret på enten internminne eller SD kortet. Det burde også ha vært en sjekk i SharedProjects som sjekker om det faktisk er noen strikkemønstre i databasen.

9. Videreutvikling

Hadde jeg hatt mer tid så ville jeg ha implementert de funksjoner som jeg tenkte “Kan være med” i starten av prosjektet i tillegg til noen ekstra funksjoner. Dette tenker jeg rundt potensialet for videreutvikling av appen:

- Må kreve logg inn for deling og henting fra nett, av den grunn for å beskytte folk mot ufint innhold (NSFW osv...). Hvis folk kommer over et strikkemønster som er generert ifra et bilde med støtende innhold så skal det gå ann å rapportere personen som delte mønsteret. Hvis det viser seg å være den samme personen hver gang så kan man da utestenge personen.
- Brukeren kan velge at det skal være symboler i stedet for farger i rutene i strikkemønsteret. Mange strikkemønsteret bruker symboler i et rutenett hvor hvert symbol representerer en farge.
- Siden det ikke finnes garn i alle mulige farge, så skal strikkemønsteret bare inneholde farger som man kan få kjøpt i butikker. Tenker her da en databaseløsning, hvor databasen inneholder alle mulige farger som det er mulig å få tak i strikkekarn til.
- Brukeren kan klippe ut en del (crop) av bildet som skal bli til et strikkemønster, for eksempel at brukeren skal kunne lage mønster av et objekt på bildet og ønsker ikke å ta med det andre i bakgrunnen.
- Brukeren kan publisere et komplett prosjekt. Brukeren kan publisere prosjektet sitt som inneholder det ferdig strikkede produktet, strikkemønsteret som ble benyttet til prosjektet, hvilke farger som ble brukt, samt diverse kommentarer. Andre brukere kan da også “like” og “kommentere” hva de syntes om prosjektet.

10. Prototype

Jeg så for meg i starten av prosjektet at den største utfordringen ville være å få til å lage et strikkemønster fra et bilde. For å finne ut om dette i det hele tatt var mulig så prøvde jeg først å lage en prototype i Java. I java koden under så brukte jeg et GridLayout hvor jeg la inn pixler i celler og la griden inn i et JPanel. Jeg håpet da på tidlig i prosjektet å få til noe lignende i android, det var derfor jeg vurderte i starten å benytte android sitt gridview.

Java koden benyttet i prototypen:

```
package main;
```

```
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
```



```

import java.awt.image.BufferedImage;
import java.io.*;
import java.awt.*;

public class ImageSplit extends JFrame {
    JPanel p = new JPanel();
    /**
     *
     */
    private static final long serialVersionUID = 1L;

    public static void main(String[] args) throws NumberFormatException,
IOException {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        System.out.println("Bredde: ");
        int sizex = Integer.parseInt(br.readLine());
        System.out.println("Høyde: ");
        int sizey = Integer.parseInt(br.readLine());
        new ImageSplit(sizex, sizey);
    }

//https://stackoverflow.com/questions/13605248/java-converting-image-to-buffe
redimage
    public static BufferedImage ImageToBufferedConvert(Image img){
        if (img instanceof BufferedImage){
            return (BufferedImage) img;
        }
        BufferedImage bimage = new BufferedImage(img.getWidth(null),
img.getHeight(null), BufferedImage.TYPE_INT_ARGB);

        Graphics2D bGr = bimage.createGraphics();
        bGr.drawImage(img, 0, 0, null);
        bGr.dispose();

        return bimage;
    }

    public ImageSplit(int sizex, int sizey) throws IOException {
        super("ImageSplit");
        setSize(sizex*30, sizey*30);
        setResizable(true);
    }

```



```

        setDefaultCloseOperation(EXIT_ON_CLOSE);
        GridLayout grid = new GridLayout(sizex, sizey);
        grid.setHgap(1);
        grid.setVgap(1);
        p.setLayout(grid);

        File file = new File("pokemon.jpg");
        FileInputStream fis = new FileInputStream(file);
        BufferedImage image = ImageIO.read(fis);
        BufferedImage bImage =
ImageToBufferedConvert(image.getScaledInstance(sizex, sizey,
Image.SCALE_DEFAULT));

        BufferedImage imgs[][] = new BufferedImage[sizex][sizey];
        for (int y = 0; y < sizey; y++) {
            for (int x = 0; x < sizex; x++) {

                imgs[x][y] = bImage.getSubimage(x, y, 1, 1);
            }
        }
        System.out.println("Splitting done");

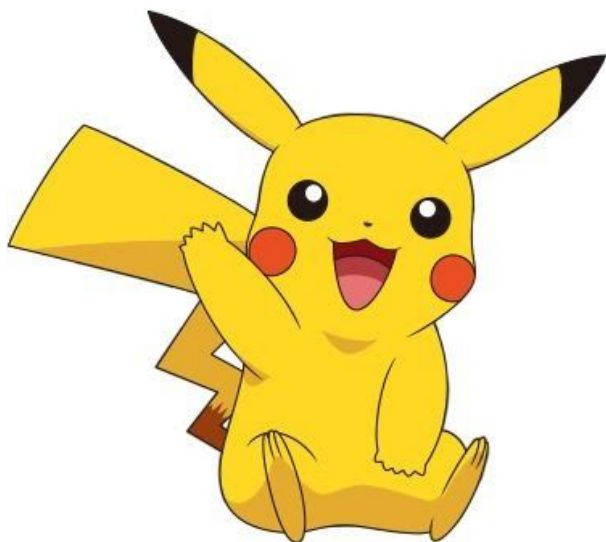
        int counterx = sizex;

        for (int y = 0; y < sizey; y++) {
            for (int x = 0; x <= sizex; x++) {
                if(x == sizex){
                    p.add(new JLabel("" + counterx));
                    counterx--;
                    break;
                }
                p.add(new JLabel(new
ImageIcon(imgs[x][y].getScaledInstance(30, 30, Image.SCALE_DEFAULT))));
            }
        }

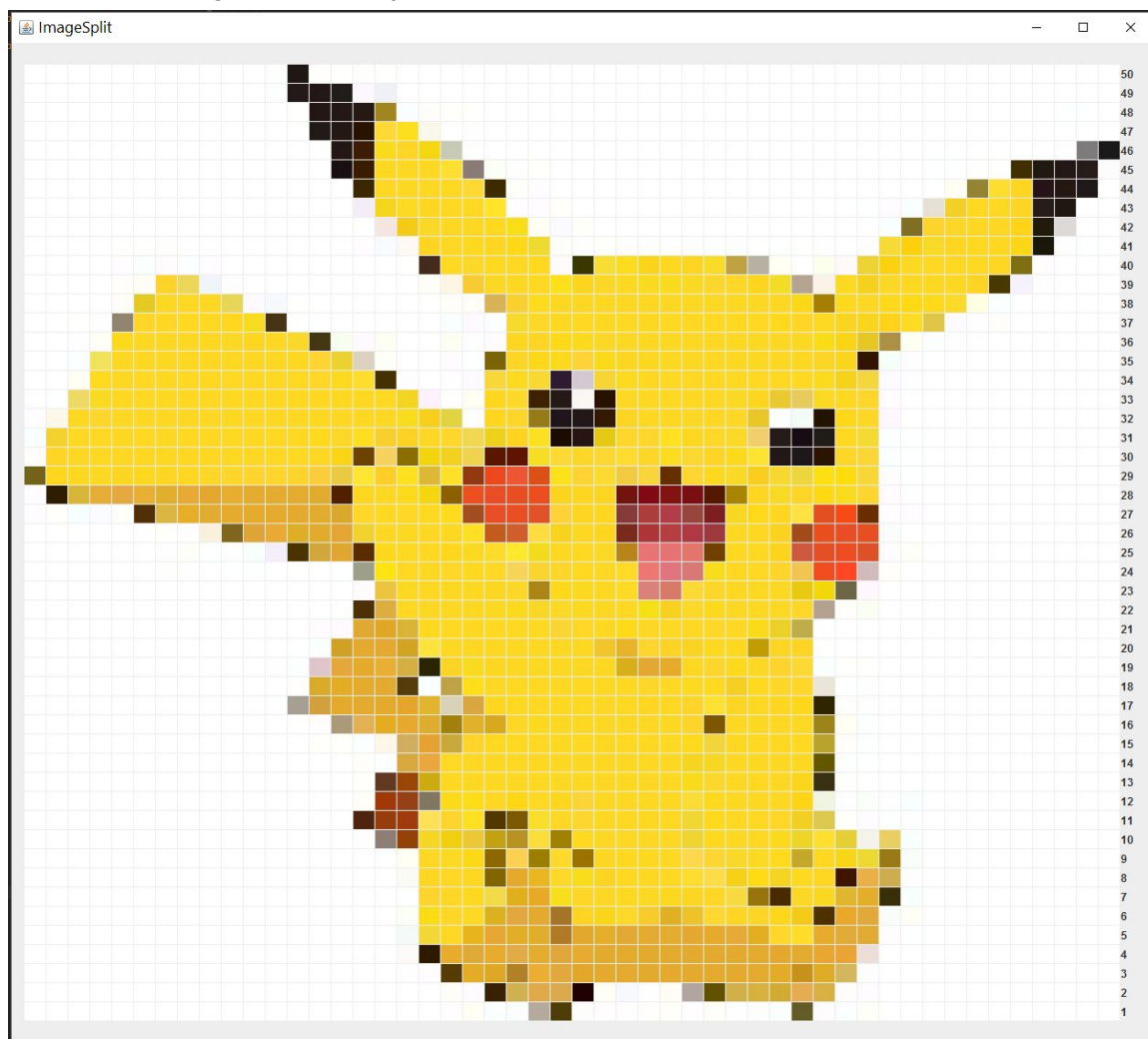
        add(p);
        setVisible(true);
    }
}

```

Bildet som ble benyttet til prototypen:



Griden som blir generert med javakoden:



Etter å ha fått til prototypen og så at det faktisk var mulig å få til prosjektets hovedfunksjon så valgte jeg å gå videre med ideen om å lage StrikkeAppen.

11. Konklusjon

I prosjektet har det blitt laget en app dedikert til å lage strikkemønstre for strikkere. I appen så finnes det funksjoner hvor brukeren kan ta et bilde med kamera eller velge et bilde fra enhetens mapper til å lage et strikkemønster. Man kan også lagre et strikkemønster til enhetens SD kort og dele det slik at andre brukere kan få tak i det og lagre det til sin egen enhet. I prosjektet pga begrenset med tid så har det blitt laget en prioritetsliste over hvilke funksjoner som må, bør og kan være med. I starten av prosjektet så var det lagt mest vekt på å få til hovedfunksjonen som var å lage et strikkemønster utifra et bilde og var basert på problemstillingen hvor strikkere vanligvis er nødt til å sette opp og tegne et strikkemønster selv. Som er en kjedelig og tidskrevende prosess. Med appen så er hensikten at strikkere skal slippe denne prosessen. Det ble gjort vurderinger hvorvidt det var mulig å gjennomføre prosjektet så det ble først gjort et forsøk på å lage en prototype i java. Prototypen demonstrerte at det var mulig å bryte opp et bilde og sette det sammen til et strikkemønster. Neste utfordring var å få til noe lignende i android studio, etter tips fra studass ble en lignende løsning lagd hvor det ble benyttet canvas til å tegne strikkemønsteret i stedet for å sette pixler inn i et gridview. For å få til deling, liste opp og lagre delte strikkemønstre så ble det benyttet Firebase. For videreutvikling har det blitt satt opp en liste over funksjoner som kan bli implementert for å gjøre appen til en mer fullverdig app.

Referanser

Kilder

Nummeret til hver link peker til listen i del 7: Hva fikk jeg hjelp til.

1. <https://stackoverflow.com/questions/13605248/java-converting-image-to-bufferedimage>
2. <https://stackoverflow.com/questions/2091465/how-do-i-pass-data-between-activities-in-a-android-application>
3. <https://acomputerengineer.wordpress.com/2016/06/07/flip-imagebitmap-horizontally-and-vertically-in-android/>
4. <https://stackoverflow.com/questions/14066038/why-does-an-image-captured-using-came-ra-intent-gets-rotated-on-some-devices-on-a>
5. <https://stackoverflow.com/questions/24842550/2d-array-grid-on-drawing-canvas>

6. <https://www.youtube.com/watch?v=mPOhnTnLcSY>
7. <https://github.com/chrisbanes/PhotoView>
8. <https://stackoverflow.com/questions/5536066/convert-view-to-bitmap-on-android>
9. <https://stackoverflow.com/questions/15662258/how-to-save-a-bitmap-on-internal-storage>
10. <https://stackoverflow.com/questions/5369682/get-current-time-and-date-on-android>
11. <https://stackoverflow.com/questions/5486529/delete-file-from-internal-storage>
12. <https://stackoverflow.com/questions/2478517/how-to-display-a-yes-no-dialog-box-on-and-roid>
13. <https://stackoverflow.com/questions/8515830/how-to-add-images-on-gridview>
14. <https://developer.android.com/guide/topics/ui/layout/gridview.html>
15. <https://stackoverflow.com/questions/39905719/how-to-download-a-file-from-firebase-storage-to-the-external-storage-of-android>
16. <https://www.simplifiedcoding.net/firebase-storage-example/>
17. <https://stackoverflow.com/questions/37335102/how-to-get-an-array-with-all-pictures>
18. <https://developer.android.com/training/appbar/setting-up.html>
19. <https://stackoverflow.com/questions/42571558/bitmapfactory-unable-to-decode-stream-java-io-filenotfoundexception>
20. <https://developer.android.com/training/camera/photobasics.html>
21. <https://stackoverflow.com/questions/38352148/get-image-from-the-gallery-and-show-in-imageview>
22. <https://stackoverflow.com/questions/17853360/check-entered-value-is-number-or-not>
23. <https://stackoverflow.com/questions/5794506/android-clear-the-back-stack>
24. <https://stackoverflow.com/questions/1016896/get-screen-dimensions-in-pixels>