

Eksamen app-utvikling rapport

Forklaring av de overordnede valgene i forhold til brukergrensesnitt og navigering

Jeg valgte å løse oppgaven på den simpleste og mest kompakte måten jeg kunne, så mange eller de fleste valgene jeg har gjort innebærer å begrense både klasse-bruk, XML-sider og ren kode ned til minimum. Koden jeg har laget er hovedsakelig original og min egen, og der jeg har brukt kode fra nettet eller tatt inspirasjon, har jeg lagt det tydelig inn som kommentar i koden. Jeg valgte også å unngå fremmede bibliotek som ikke har vært inkludert i pensum eller på canvas, selv om jeg hadde noen som kunne gjort noen av oppgavene mye lettere og kortere.

Jeg oppfatta etter å ha lest oppgava at appen krevde bare 2 skjermer (hvis vi ikke regner med innstillinger), en for å vise fram tilsyn objekt, og en for å vise mer om tilsyn i tillegg til en liste med alle kravpunkt tilhørende tilsyn id. Derfor tenkte jeg at jeg ville designe to aktiviteter, og hver av dem ville fokusere på det av lister og komponenter som var logisk og ha med i aktiviteten. Jeg var ganske fast bestemt på å vise at jeg behersket bruken av fragmenter på eksamen, men jeg klarte ikke å finne noe sted i oppgava der det var logisk å bruke det. Jeg ville begrense mengden kode i MainActivity og heller splitte det opp på en kreativ måte med fragmenter. Men det førte bare til at variabler ble unødvendig utilgjengelige når jeg for eksempel prøvde å la et fragment ta hånd om tilsyn recycleviewet i MainActivity. Dette er hovedsakelig fordi variabler som for eksempel søkefelt og søkeknappene er direkte knyttet til recycleviewet. Så jeg bestemte for å ta meg til ro med 2 aktiviteter.

For å vise fram tilsyn objekt valgte jeg å sette opp et recycleview, siden det er noe jeg har god erfaring med etter obligatorisk oppgave 1 og 2. Trykker man på et kort åpner man neste aktivitet som viser fram et scrollview med tilsyn informasjon i tillegg til enda et recycleview med kravpunkt. Denne er satt til å være horisontal i et forsøk på litt bedre brukervennlighet, og

roterer man skjermen vil recycleviewet omformes til vertikalt igjen. Overordnede valg i forhold til design av brukergrensesnitt har vært å fokusere på å ha en god deling på display-delen (recycleview) og bruker-input delen (knapper og felt), så det er oversiktlig og lett å navigere. Jeg fant også ut at tilsyn "kortet" jeg designa i første aktivitet også kunne passe like godt inn for å vise fram kravpunkt i andre aktivitet, så jeg bestemte meg for å gjøre kortet (info_kort.xml) om til en template som kunne vise både Tilsyn og Kravpunkt, i stedet for å designe en for hver.

Forklaring av tankegang/fremgangsmåte for hver del-oppgave (a-e).

A.

Det første jeg gjør er å designe layout med søkefelt, knapper og et recycleview for å vise tilsyn. Jeg at jeg må hente ut strenger fra felte bruker har fylt ut/ikke fylt ut, og sende de videre til et volley kall som setter opp en tabell av type Tilsyn. Det var den generelle tankegangen.

På linje 164 i MainActivity starter volley metoden. Her legger jeg først opp alle strenger jeg potensielt trenger i spørringen, og sjekker de som må sjekkes før jeg setter dem. Før volleyen setter jeg opp spørre-strengen. Her legger jeg bare inn alle strenger uavhengig om de er tomme eller ikke. Om man legger inn et felt i en url spørring og verdien er tom, vil spørringa bare ignorere det og heller lete etter andre felt som har verdier. Dette er perfekt for oppgava siden bruker selv kan velge hvilke felt som skal fylles ut, og bare de som er fylt ut vil faktisk telle mot spørringa.

I Tilsyn klassa har jeg laget en metode som tar imot en streng med json objekt og formaterer det ned til Tilsyn objekt og legger det inn i lista av type tilsyn. Deretter oppdaterer jeg lista med adapter og recycleview. I TilsynListeAdapter setter jeg alle felt i info_kort.xml til felte som er lagra i hvert objekt av tilsyn lista (LinkedList<Tilsyn>). Til slutt bruker jeg bare en switch til å finne ut hvilket type smilefjes jeg vil vise siden karakter er lagret i Tilsyn som en streng.

B.

Denne oppgava starter fra onclick metoden i TilsynListeAdapter (linje 115) der jeg legger inn tilsynid feltet fra datasettet inn i en bundle sammen med navn, og sender det til/starter aktivitet TilsynAktivitet. Id'en jeg sender i bundle er lagret i et TextView som er satt til "visibility=gone" (i info_kort.xml linje 100), siden jeg måtte på en måte lagre informasjonen jeg trengte via XML på grunn av måten adapteren er satt opp på. Det er sikkert en bedre måte å gjøre det på, men det påvirker heldigvis ikke layouten. Med tilsyn id'en overført til neste aktivitet TilsynAktivitet har jeg alt jeg trenger til neste skjerm.

I TilsynAktivitet valgte jeg å bruke 2 volley kall (en for tilsyns-informasjon(linje 95), en for kravpunkt objekt(linje 112)), ved hjelp av lambda kall hvor jeg implementerer Volley grensesnittet i metoden. Jeg vurderte også å bare sende den fulle tilsyn-objekt strengen via en bundle, og opprette den i TilsynAktivitet uten å kalle på enda en volley, men det kom med begrensninger. Siden jeg sender bundle via onclick metoden i TilsynListeAdapter, har jeg bare tilsyn-objekt felte som blir brukt akkurat til å vise fram tilsyn korta, altså 5 felt. Jeg syns ikke det var veldig logisk å vise akkurat samme informasjon om tilsyn i neste skjerm som på kortet i første, så jeg bestemte meg for å gjøre et nytt volley kall og hente mye mer informasjon. Det førte naturligvis til en enorm setup metode med mange getString() metoder på json objektet (linje 148). Jeg valgte å bruke et TextView i et scrollview her i stedet for et TextView per felt for mindre kode. For å hente ut kravpunkt objekt gjorde jeg bare akkurat det samme som i A, bare at jeg i stedet brukte tilsynid etter endpoint (linje 112).

C.

Jeg valgte å sette opp både filter for årstall, filter for karakter og sortering. Disse ble satt opp på samme måte som søkefelte der jeg bare henter ut hvilket "item" spinneren ligger på, og sender det med i spørringa til volleyen (linje 164 MainActivity). Sortering er satt opp sånn at lista kan sorteres på både navn og poststed. Dette krevde at jeg måtte lage to lambda metoder som implementerte comparator grensesnittet og sammenligna med eget felt. En vanlig implementasjon ville ligget i klasse-deklarasjonen med en compareTo() metode i klassen, men hadde vært begrensa til å sammenligne på bare et felt.

Swipe funksjonen er implementert i MainActivity (linje 298), og lytter etter swipes fra bruker. Her lagrer vi først tilsyn-objektet og posisjon til kortet som blir swipa. Så har jeg brukt en dialog (linje 317) til å spørre om brukeren vil slette det eller ikke, og bruker bare LinkedList.remove() for å slette objektet fra lista. Deretter legger jeg opp en snackbar for å gi brukeren et alternativ om å gjenopprette, som bruker lagrede tilsyn objekt og posisjon til å legge det inn igjen på samme plass.

Noe jeg også gjør både i MainActivity og TilsynAktivitet, er å lagre recycleview listene i onSaveInstanceState med putSerializable. På den måten kan jeg ikke bare gjenopprette recycleviewet når bruker roterer skjermen og slepper å kjøre volley på nytt, men scrolleposisjonen blir også gjenoppretta, så bruker ender opp akkurat der bruker var før rotasjon.

D.

I denne oppgava valgte jeg å hente kode fra GPS_Demo som er lagt på canvas. Jeg så at den koden fant postnummer til bruker, så jeg fant ut hvilken del av koden som gjorde akkurat det, og implementerte det inn i min egen (metode linje 383 MainActivity, metode linje 359 MainActivity). Deretter sender jeg bare postnummer inn i spørringa (linje 188 MainActivity), hvis bruker har trykket på knappen "finn tilsyn i området". Om ikke, er det bare en tom streng.

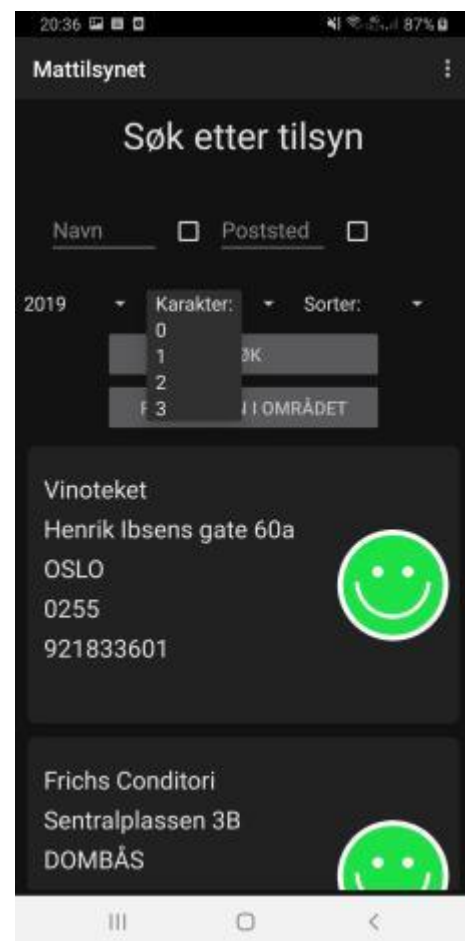
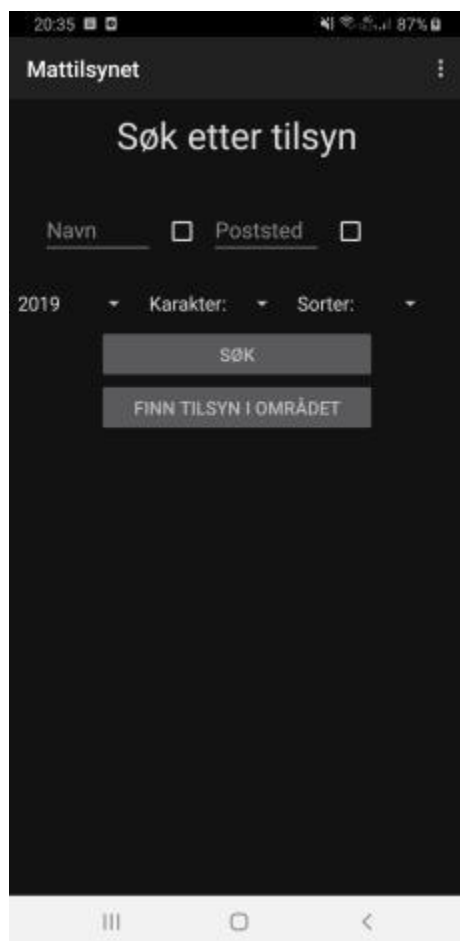
E.

Her valgte jeg å bruke bibloteket preference for å sette opp instillinger.xml. I MainActivity lytter metoden (linje 263) etter klikk på meny i toolbar, og hvis bruker trykker på "instillinger", kjøres InstillingerAktivitet. Her legger jeg opp et fragment som holder på instillinger.xml, og lytter etter forandringer på årstall filter. Preference bibloteket gjør lagring av informasjon automatisk mellom app-kjøringer, så jeg trenger ikke å tenke på det når det kommer til innstillingene.

I instillinger.xml satte jeg opp noen felt og ga dem nøkler som jeg henter i MainActivity for å legge inn favoritt navn, poststed og årstall. I Resume (linje 431 MainActivity) henter jeg årstall i innstillinger ved hjelp av minePreferanser objektet, og setter spinner til det. På navn og

poststed felt har jeg lagt inn en checkbox og metoden paaCheckBoxTrykk (linje 238) som lytter etter trykk, og fyller inn felte med innstillinger om de er satt.

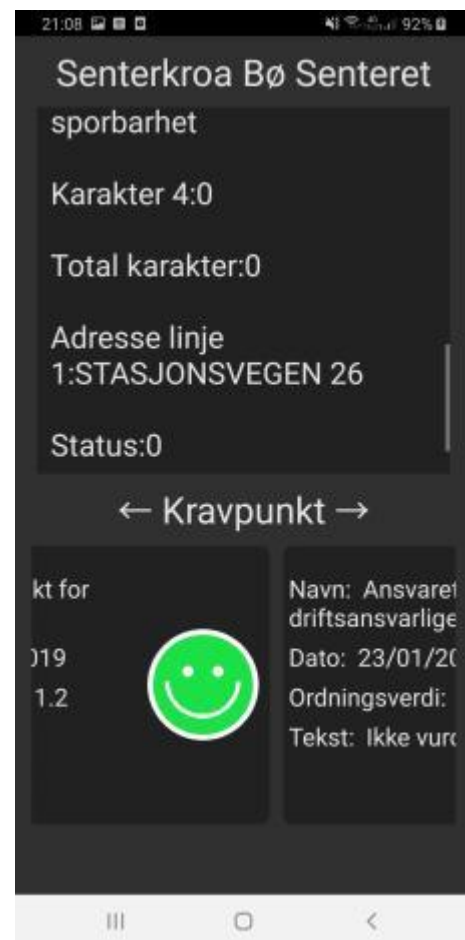
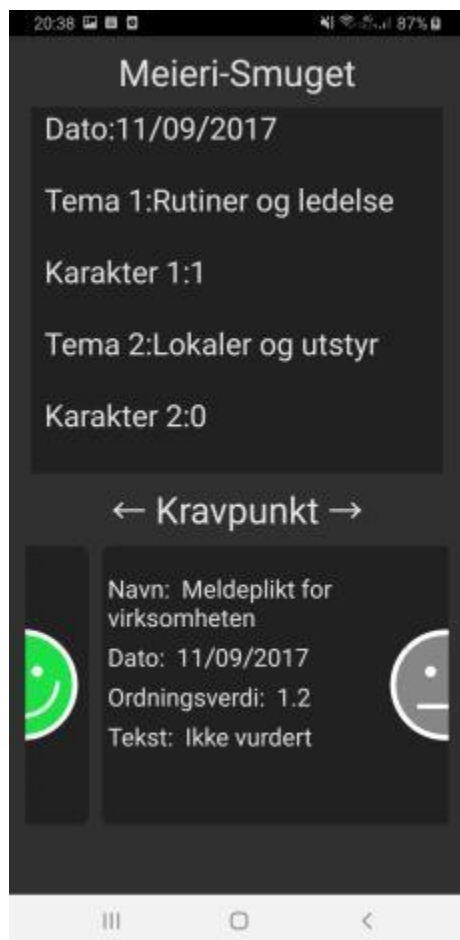
[Brukerveiledning med skjermbilder fra appen](#)



Skjermbilde 1

I første skjermbilde er det to felt som bruker kan fylle ut, navn og poststed, i tillegg til 3 lister der bruker kan velge å filtrere på årstall, karakter og sortering. Hvis vi trykker "søk" uten å fylle inn noe, vil de første 100 tilsyn objekta komme opp som kort med informasjon og et smilefjes

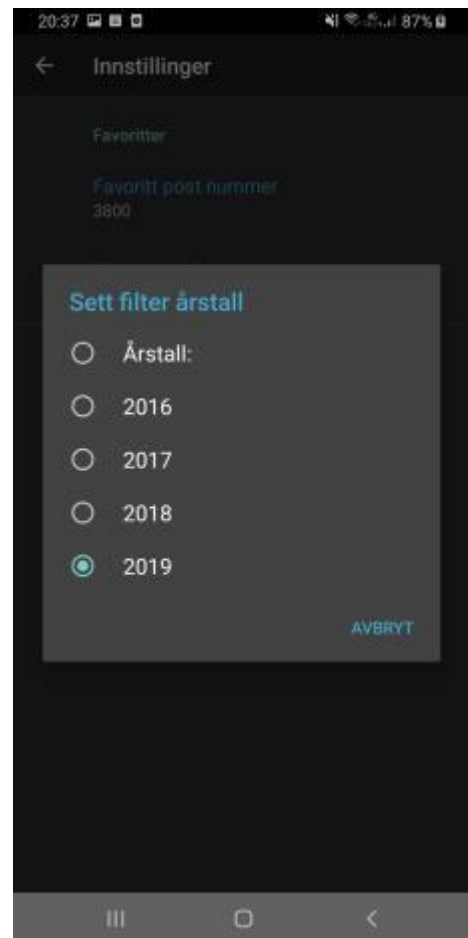
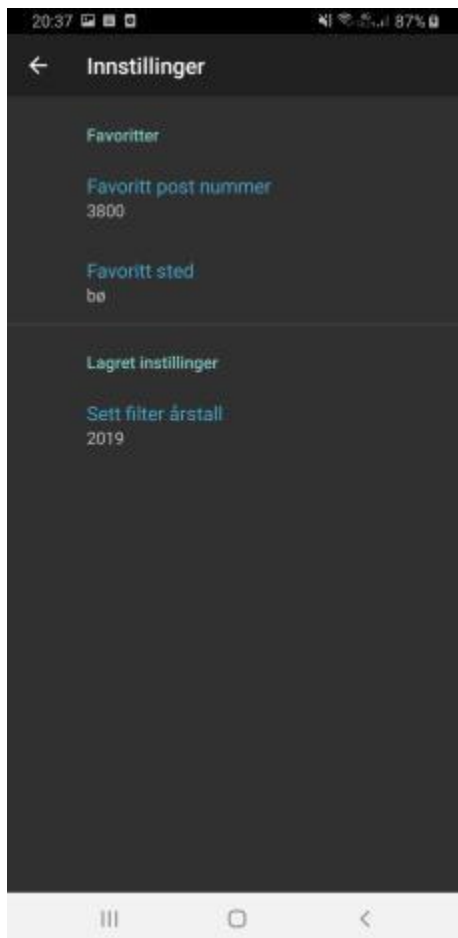
som beskriver karakter. Disse kortene kan scrolles gjennom, sveipes vekk, og trykkes på for å gå videre til neste skjermbilde og se hvilke kravpunkt tilsynet har. Navn og poststed kan også fylles ut automatisk ved å trykke på boksene ved siden av, men bare om bruker har allerede vært i innstillinger og lagt til preferanser. Dette kan gjøres om bruker trykker på prikkene oppe i høyre hjørne og får opp et valg om å gå inn i innstillinger.



Skjermbilde 2

Her kan bruker se mer informasjon rundt tilsyn-kortet som ble trykket på i forrige skjermbilde. Det er mulig å scrolle nedover for å se mer informasjon i topp-vinduet. I bunn har vi alle kravpunkt som tilhører tilsyn kortet som ble trykket på. Bruker kan scrolle disse bortover for å

finne fram til riktig kort. For å gå tilbake til skjermbilde 1, må bruker bruke den innebygde/standard gå tilbake android knappen.



Skjermbilde 3

Her kan bruker legge inn preferanser som favoritt post nummer, favoritt sted og filter årstall. Post nummer og sted blir mulig å utfylles i skjermbilde 1 ved trykk på boks, om man fyller dem ut her. Om bruker bruker setter et filter årstall, vil skjermbilde 1 automatisk sette filter til det

årstallet, og bare søke på tilsyn som skjedde det året. Man kan gå tilbake til skjermbilde 1 ved hjelp av både pila oppe til venstre og innebygd android pil.

Kildebruk

- All kode som ikke er min egen eller fra pensum/canvas er allerede linka til og dokumentert i programkoden.
- Andre inspirasjonskilder:
 - <https://developer.android.com/guide/topics/ui/controls/checkbox>
 - <https://alvinalexander.com/android/android-tutorial-preferencescreen-preferenceactivity-preferencefragment>
 - <https://stackoverflow.com/questions/12793069/android-onsaveinstancestate-not-being-called-from-activity>
 - <https://medium.com/hootsuite-engineering/handling-orientation-changes-on-android-41a6b62cb43f>
 - <https://stackoverflow.com/questions/4432774/how-do-i-make-2-comparable-methods-in-only-one-class>
 - <https://stackoverflow.com/questions/30180052/you-need-to-use-a-theme-appcompat-theme-or-descendant-with-this-activity-chan>
 - <https://stackoverflow.com/questions/4823442/android-list-preferences-have-summary-as-selected-value>
 - <https://stackoverflow.com/questions/17643340/android-back-button-doesnt-return-to-previous-activity>