

PGR208 Android Programming

Konteeksamen

Vår 2025

Individuell eksamen.

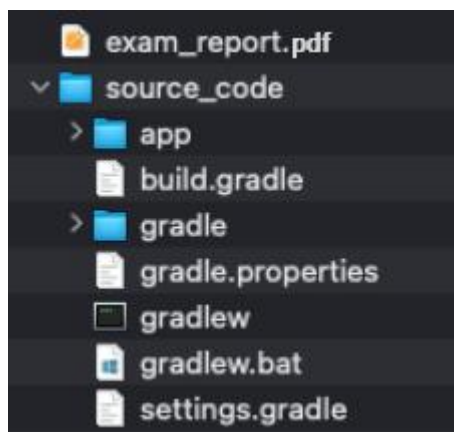
Hoveddelene av eksamenen

I denne eksamenen skal du gjøre bruk av teknikker du har lært i emnet PGR208 Android Programmering og på den måten vise hva du behersker. Eksamenen består av to hoveddeler:

1. App (80%)
2. Rapport (20%)

Instruksjoner for opplasting av besvarelse til WISEFlow:

1. Kjør Build -> Clean Project for å minske prosjektstørrelse
2. Slett local.properties-filen. Dette er for å sikre kandidatens anonymitet da local.properties-filen kan inneholde brukernavn. Se bilde under for eksempel på hvilke filer/mapper (øverste nivå) som skal være med i leveransen. Andre filer enn disse er ikke nødvendige for leveransen og skal ikke legges ved.
3. Legg alle deler av eksamenen i en felles mappe som zippes før opplasting til WISEFlow.



The app (80%): MyPokemon

You shall make an app where people can create new Pokemons and retrieve information about Pokemons from the PokeAPI (a free Web API).

The app shall consist of the following 3-4 screens:

Screen #1: Create New Pokemon

Here the user is presented with the possibility to create a new (made up) Pokemon. These should be stored in a local database.

Screen #2: Show Created Pokemons

Here you show a list of all saved Pokemons from Screen 1. Also handle the situation where the user comes to this screen without having created any new Pokemons yet.

Screen #3 (and #4 if you want): Pokemons from PokeAPI, and search in PokeAPI

Give the user the possibility to:

- show several Pokemons (with information and image) from the PokeAPI
- search Pokemon in the PokeAPI by id.

If you want to you can split this functionality over 2 screens.

This is an endpoint in the PokeAPI where you can get (HTTP) a specific Pokemon with id search: <https://pokeapi.co/api/v2/pokemon/1>.

See also next page to see example of a JSON result. Create an array containing at least 4 dummy objects to have as a fallback in case the API is not available/goes down.

Main page of PokeAPI: <https://pokeapi.co/>

The screenshot below shows an example of how JSON for one Pokémon may look – gotten from the referred to endpoint on the previous page. One does only need to include 4-5 properties for a higher grade.

JSON	Raw Data	Headers
Save	Copy	Collapse All Expand All (slow) Filter JSON
▶ abilities:	[...]	
base_experience:	64	
▶ cries:	{...}	
▶ forms:	[...]	
▶ game_indices:	[...]	
height:	7	
held_items:	[]	
id:	1	
is_default:	true	
location_area_encounters:	" https://pokeapi.co/api/v2/pokemon/1/encounters "	
▶ moves:	[...]	
name:	"bulbasaur"	
order:	1	
past_abilities:	[]	
past_types:	[]	
▼ species:		
name:	"bulbasaur"	
url:	" https://pokeapi.co/api/v2/pokemon-species/1/ "	
▼ sprites:		
▶ back_default:	" https://raw.githubusercontent.com/PokeAPI/sprites/pokemon/back/1.png "	
back_female:	null	
▶ back_shiny:	" https://raw.githubusercontent.com/PokeAPI/sprites/pokemon/back/shiny/1.png "	
back_shiny_female:	null	
▶ front_default:	" https://raw.githubusercontent.com/PokeAPI/sprites/pokemon/1.png "	
front_female:	null	
▶ front_shiny:	" https://raw.githubusercontent.com/PokeAPI/sprites/pokemon/shiny/1.png "	
front_shiny_female:	null	
▶ other:	{...}	
▶ versions:	{...}	
▶ stats:	[...]	
▶ types:	[...]	
weight:	69	

Generelle krav og retningslinjer

- Du skal gjøre bruk av teknikkene som er gjennomgått i emnet
- Kotlin er programmeringsspråket og Android Studio din IDE
- Du forventes å implementere god prosjektstruktur, koderyddighet, og god bruk av teknikker som vist i emnet.
- Du blir ikke vurdert direkte på vakkert design, men du blir vurdert på kode for å lage design med tanke på kompleksitet og omfang. Tenk også over hvordan en app bør utformes for å være brukervennlig – tenk over hvordan den vil oppleves av en bruker. I den sammenheng kan det være greit å tenke på å implementere feedback til bruker der det gir mening; eksempelvis etter handlinger bruker gjør osv.

Rapporten (20%)

En viktig del av denne eksamenen er å vise at du er bevisst og kan reflektere over teknologien og teknikkene du har lært i emnet.

Skriv en rapport på 400-500 ord som inkluderer følgende fire ting:

1. **Oversikt over funksjonalitet du har laget:** Lag en oversikt over hvilken funksjonalitet du har laget. Dette kan eksempelvis lages med en tabell med to kolonner: navn på funksjonalitet og kort beskrivelse av funksjonalitet.
2. **Skjermdump av alle skjermer i appen:** Skjermbilder av alle skjermer med en kommentar om grensesnittet og hva som skjer der/hva man kan gjøre der. Bør også inkludere skjermbilde for hvordan tilbakemeldinger til brukeren (for eksempel hvis klikker på søk uten å ha fylt inn noe tekst) ser ut.
3. **Beskrivelser og skjermbilder av hovedteknikker brukt:** Beskriv hovedteknikkene du har brukt i appen. Med beskrivelse menes minimum navn på teknikk og hva teknikken er for, og der hvor det passer seg, også hvorfor man bruker teknikken. Inkluder skjermbilder av kode og eventuelt modeller/diagrammer for å eksemplifisere teknikkene du omtaler.
4. **Kvalitet og struktur:** Skriv hva du har gjort for å sørge for at prosjektstrukturen er god, hvordan og hvorfor du har navngitt variabler, funksjoner, klasser osv. slik du har gjort, og eventuelt andre ting som angår struktur og kodekvalitet.

Rapporten skal leveres som en PDF-fil.

Assessment guide for students and assessors

The app: 80%

The screens will have weight according to the complexity in them, as some may be simpler than others.

Make sure you think about writing good, scalable code, and using the principles learnt in the course. Also make sure to use a variety of the most important coding techniques and methods which you have learnt in the course.

The coding of the UI also counts, not in the sense of being “beautiful” in itself, but rather that UI elements have been styled in a complex/amount-wise good enough manner. Feedback to the user after important actions or about situations is also important.

Some main elements/parts:

Screen #1-4

- **HTTP / API**
 - General implementation and usage of HTTP to retrieve data from the provided third-party API
- **Local storage**
 - General implementation and usage of storing data locally
- **UI/UX**
 - Overall UI and user experience implementation (i.e. error handling, and messages to user)
- **Navigation between screens**
 - Navigation and passing of data between screens where/if necessary
- **Architecture and project structure**

Report: 20%

Show awareness of the techniques, code, and functionality created. Be able to have an overview of techniques. Show understanding of the selected techniques. Be able to reflect on code quality and structure.