

Kortfattet beskrivelse av løsningen

- Skulle noe vært gjort annerledes - Allt (Eirikur) :P

Løsningen består av en mobilapplikasjon til Android, med minimum SDK 4.4(KitKat), og target SDK 6 (Marshmallow). Løsningen har 2 Activities: MapsActivity og HighscoreActivity. En bruker skal samle Pokemons ved å skrive inn en kode (eller scanne en NFC-tag) som finnes på en viss lokasjon i Oslo. Dette er gjort ved å implementere Google Maps sitt API, Løsningen benytter seg av en SQLite database for å lagre informasjon av hvilke Pokemons brukeren har samlet.

SQLite

For å koble seg til en SQLite-database, må man først opprette en database og tabell(er) databasen skal ha. Dette gjøres ved å extend SQLiteOpenHelper-klassen, og override metodene til superklassen (oncreate og onUpdate). Disse *må* overrides for å ta i bruk klassen. SQLite er en lightweight database, som passer godt til mobilt bruk fordi den ikke krever mye batteribruk. Dataen leses via SQL-spørringer mot databasen, og man må dermed hente ut data via SQL-spørringer som returnerer et Cursor objekt. For å lese dataene, må man navigere seg gjennom Cursor-objektet ved å lese resultatene linje for linje.

Når man anvender sig av sqlite måste man passa på att ha en context för att få jobba med adaptorn. Ett annat hänsyn man borde ta är att skydda sig mot sql injections genom prepared statements.

AsyncTask

AsyncTask er en klasse for å håndtere tråder i en Android mobilapplikasjon. Objektet gjør det mulig å gi informasjon til UI-tråden i programmet, uten at man må opprette tråder eller handlerne for å kommunisere til UI-tråden. Fordi alt av User Interface i Android bare kan styres via hovedtråden (main UI-tråd), er en AsyncTask en måte å få oppretta tråder på som kan gi info til denne UI-tråden, og samtidig utføre operasjoner i bakgrunnstråder.

En AsyncTask har fire hovedfunksjoner:

1. onPreExecute()

Metoden sender noe som vises og kjøres i UI-tråden ved oppstart (for eksempel en melding om at arbeid nå skal gjøres, eller en progress bar).

2. doInBackground(Params...)

Hovedmetoden som må implementeres hver gang man oppretter en AsyncTask. Den tar imot eventuelle parametre som skal brukes/bearbeides, for eksempel en List<String>. Her utføres bakgrunnsoperasjonen som AsyncTasken er ment å utføre, og kan være temmelig omfattende (for eksempel et kall til en database som i vår løsning der vi henter ut alle navn fra en gitt tabell i en database).

3. onProgressUpdate(Progress)

Dersom doInBackground-metoden har gjort et kall på publishProgress(Progress), kjøres denne metoden av UI-tråden og gir brukeren en indikasjon på at prosessen er i gang (for eksempel kan man vise en progressbar eller vise en melding i et TextView).

4. onPostExecute(Result)

Denne metoden kjøres i UI-tråden etter arbeidet er utført i doInBackground-metoden, og tar imot parameteret Result som er resultatet av doInBackground-metoden.