

TDP028 App-beskrivning

Viktor Rösler

18 december 2021

Outdated App-beskrivning finns i readme.md

1 Svampkartan

Min app Svampkartan låter användaren markera på en karta var dom har hittat svampar. Till varje svampmarkör ska användaren kunna lägga till en bild och skriva kommentarer. Datum och tid på när svampen lagts till på kartan ska också sparas. Poängen med appen är att hjälpa svampplockare komma ihåg var de har hittat vilka svampar.

Appen ska ha stöd inloggning och en vänlista. Användaren ska kunna dela sina svampfynd med sina vänner, och få sina vänner svampmarkörer på sin egen svampkarta.

En extra funktionalitet jag skulle vilja få till är identifiering av svampar via kameran. Det skulle fungera så att användaren tar bilder genom appen på en svamp de hittat, och då ska appen använda image recognition för att identifiera vilken art svampen är. Med ett knapptryck ska användaren sedan kunna lägga till svampen på kartan.

Bildigenkänningen ska implementeras med hjälp av ett neuralt nätverk genom Tensorflow (Lite) som tränas på (delar av) följande dataset med svampbilder: https://github.com/visipedia/fgvcx_fungi_comp

Om automatisk klassificering av svampar implementeras ska appen varna användaren att klassificeringen inte kommer att vara korrekt varje gång, och att det är användarens eget ansvar att bekräfta vilken svampart det rör sig om.

Appen ska ha inställningar för att ändra språk, och för att byta mellan dark- och lightmode.

2 Betygsmål

Jag siktar på att få en 5:a (17p+) för projektet. Nedan följer de krav jag planerar att uppfylla.

2.1 Milstolpar

- V35 - Appbeskrivning (tänkt funktionalitet och wireframe-navigationsdiagram), samt konkurrensanalys inlämnade i repo. (0.5p)
- V36 - En enkel app (antingen Hello world, eller början på egen app) med textruta, och en knapp inlämnad i repo. (0.5p)
- V37 - Två skärmar, med navigering emellan, inlämnad i repo. (0.5p)
- V38 - Visa faktiskt innehåll på skärmarna, dvs. enkel backend (T.ex. Firestore) inkopplad. Kod inlämnad i repo. (0.5p)
- V39 - Veckoplanering av resterande projektarbetet, med kravlista (funktionella krav) på vad som kommer att implementeras, och vad som ska uppnås varje vecka, är inlämnade i repo. (0.5p)
- V47 - Halvtids-screencast på ca. 2 minuter, där du demar en tidig version av din app (ungefär motsvarande en Minimal Viable Product), inlämnat i repo. (0.5p)

3p totalt.

2.2 Tekniska Krav

Minst 5p

- Modulär kod, enligt någon vald princip. Referera till best practices, t.ex. ange i kommentarer eller i Readme-filen länk till websida eller annan källa du har valt att följa. (2p)
- Bra avvägning mellan användning av Activity och Fragments. Best practices säger att man gärna ska använda Fragments där det går, då dessa är mer lättviktiga än Activities. Activities bör användas när det gäller byten mellan olika användaraktiviteter, t.ex. lista inkorg byts mot skriva mail. (1p)
- Hantering av stora och små skärmar med olika layouter. Användaren ska kunna byta från mobil till tablet utan att tappa i upplevelse. Bl.a. är det viktigt att information på skärmen förblir detsamma, samt att layout:en ser bra ut i båda lägen. Användarupplevelsen kan enkelt testas genom att rotera mobilen eller emulatorn före uppstart. (1p)

- Hantering av skärmrotation. om enheten roteras ska skärminnehållet förbli stabilt. Samma innehåll visas som före rotationen, men nu med annan layout. Användarupplevelsen ska alltså bevaras, även om mobilen eller tablet:en roteras under körning. (2p)
- Hantering av användar-input (klickar) på rätt nivå. Oftast innebär detta att klickar hanteras av förälder-Activity:n, även om klicken först uppfångas av ett Fragment (knappen, eller dylikt sitter ju oftast i Fragmentet). Notera att det kan finnas fall där det är snyggare att hantera klickar på en lägre nivå, t.ex. om data ska hämtas och Fragment-innehållet ska uppdateras från en remote databas. (1p)
- Hantering av bakåtknapp, så att man aldrig hamnar konstigt vid tryck på bakåtknappen. Detta sköts oftast automatisk av Android, men kan i vissa fall kräva hantering av vad som läggs på backstack. (1p)
- Hantera, dvs. hålla koll på inloggningsstatus hos olika användare, så att bara inloggade användare kommer åt att använda vissa features, eller att inloggning krävs för att få se viss information i appen. (1p)
- Eget tekniskt krav: implementera identifiering av svampar via kameran med hjälp av machine learning (image recognition) (?p)

9p+ totalt.

2.3 API Krav

Minst 5p

Ska göras:

- Notifications. Kan skicka notifications till användaren. Notifications möjliggör för appar att visa att något relevant finns. Implementera notifications för din app och testa att visa information även när appen inte är igång. (1p)
- Multispråk-stöd. Använd Androids system för att göra så att Appen identifierar minst 2 språk och använder t.ex. Engelska som default och Svenska för användare med Svenska språkinställningar. (1p)
- Någon typ av UX inställning, t.ex. möjlighet att ändra till dark mode (1p)
- Använder Google maps på meningsfullt sätt i appen. (2p)
- Kontohantering, t.ex. lagra poäng, vänlista, eller dylikt som ska sparas över tid för olika användare (1p)
- Tredjepartsinloggning med Google eller Facebook. (1p)

7p totalt.

3 Sketch

