GitHub Username: teisen

Country Population Quiz

Description

Country Population Quiz shows you a short list of countries. Your objective is to sort the list by most populated to least populated by area.

Intended User

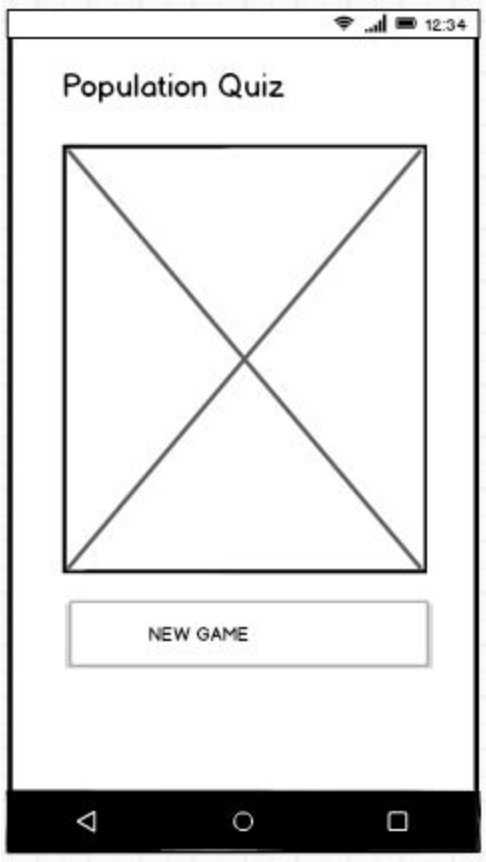
This app is intended for people who are want to learn more about the world's countries.

Features

- Connects to a web api to retrieve countries, populations and area
- Quizzes the user about their knowledge of country populations by area

User Interface Mocks

Screen 1: Start Screen



The start screen welcomes the user to the App. It has a big bold graphic as well as a button to start the game.

Screen 2: Game Question

The User is presented with a list of Countries. Each country also lists the name of the capital. The user is supposed to drag the items until they are in the correct population order. When done, the user needs to press the DONE button.

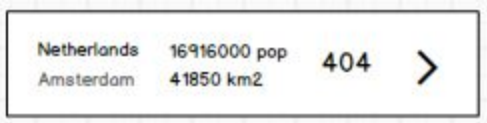
Screen 3: Correct Solution

When the user ordered the countries correctly then the user is shown the population / area values for each country. The user can click next for the next quiz. Screen 2 is shown with a new set of countries.

Screen 4: Wrong answer

When the user answers wrong this is indicated in the header and possibly with a colored background. The user then needs to reorder the list and click on DONE again. Either Screen 3 or Screen 4 is shown.

Screen 5: Widget



The Widget show a random entry from the Database. It shows the Countries Name, Capital, total population, area in square kilometers are well as the population per square km. It has a button to show the next random entry

Key Considerations

How will your app handle data persistence?

I will build a Database layer that offers access to the SQLite Database. The Database will have a single table with 4 columns. Id, Country, Population, Area.This Data is accessible through a ContentProvider.
The App uses a SyncAdapter to periodically sync the country data from the Web.

Describe any corner cases in the UX.

User hits the back button: is the user is in a game, he gets returned back to the Welcome Screen.

Describe any libraries you'll be using and share your reasoning for including them.

I will use Admob to serve ads and Google Analytics to learn more about the usage of the App.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Setup API contract if necessary
- Configure libraries: AdMob & Analytics

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for SortActivity
- Include AdMob

Task 3: Implement Database Layer

Create the Database Layer implementation.

- Define a Schema
- Define a Contract
- Create DB using a SQLHelper

Task 4: Implement SyncAdapter for the Dataload

Implement the WebAPI request to the countries API and insert the data into the DB. this only has to happen once after install of the application.

- Connect to Web API and get all the country info
- Delete all records if data exists
- Insert country info

Task 5: Implement Widget

Implement the Desktop Widget

- Make sure Widget can access the data
- Create UI fro Widget

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"