GitHub Username: sanmatishah

QuizApp

Description

Knowledge is power. Test yours!

QuizApp is a multiple choice quiz that is fun, challenging & informative at the same time!

No matter whether you're alone, with friends, or with family, you will have so much FUN that you won't stop playing this quiz.

The quiz offers 3000+ verified questions spread over 3 levels of difficulty

Have fun and improve your knowledge!

Intended User

This app is for those who are interested in improving their knowledge in a fun & engaging way.

Features

The main features of the app are listed below:
- Simple & clean user interface.
- Users can choose the category of questions based on their topic of interest.
- Users can choose the number of questions (10 or 20) per round of quiz.
- Users can set different levels of difficulty – Easy, Medium or Hard
- User can have a widget to see the score of the last quiz played.

User Interface Mocks

Main Screen

This is the main screen of the app. It has the "Play Quiz" button which starts the quiz for the user. It also provides the "Settings" menu option to navigate to the settings screen.
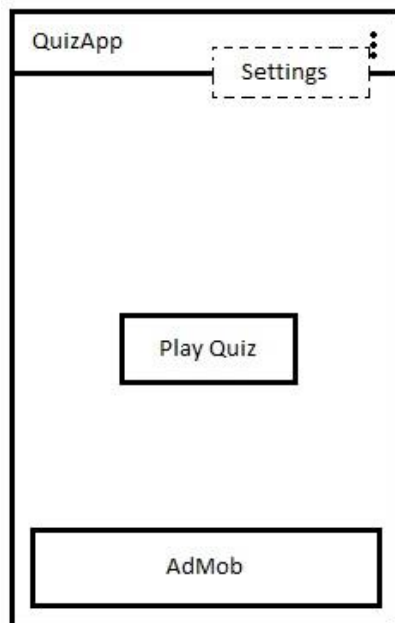
Question Screen

This is the layout of the question screen. When user clicks the "Play Quiz" button on "Main Screen", then the question & the multiple options of the answers are displayed to the user.

Mobile Portrait Layout for Question Screen

Question 9

> Q. This is a sample question. What do you think is the answer to this?

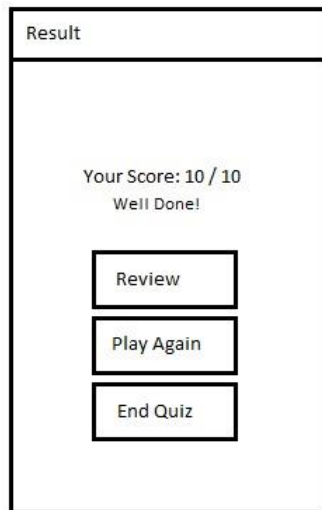| Answer A |
|---|
| Answer B |
| Answer C |
| Answer D |

Previous	Next

Mobile Landscape & Tablet Layout for Question Screen

Question 9

> Q. This is a sample question. What do you think is the answer to this?

Answer A	Answer B
Answer C	Answer D
Previous	Next

Result Screen

This is the result screen of the app. It displays the score to the user.
It has the options to:
   a) Revisit the quiz which the user just played,
   b) Start a new quiz with the same settings, or,
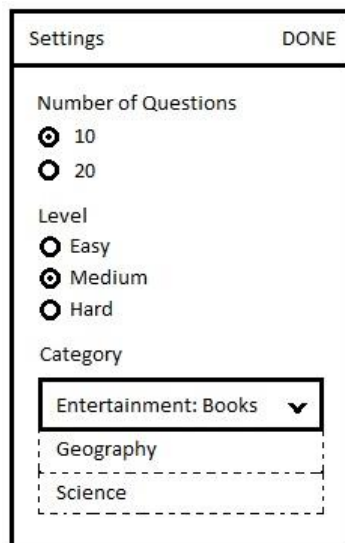   c) End quiz and move to the "Main Screen"

```
Result
───────────────────────────


            Your Score: 10 / 10
               Well Done!

            ┌─────────────────┐
            │     Review      │
            └─────────────────┘
            ┌─────────────────┐
            │   Play Again    │
            └─────────────────┘
            ┌─────────────────┐
            │    End Quiz     │
            └─────────────────┘

```

Settings Screen

This is the settings screen of the app. It provides the option to.
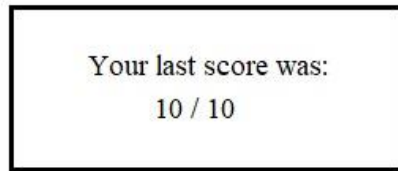   a) Set the number of questions – 10 or 20
   b) Set the level of difficulty – Easy, Medium, or Hard
   c) Choose the category based on user's topic of interest.

```
Settings                    DONE
───────────────────────────────

Number of Questions
  ◉  10
  ○  20

Level
  ○  Easy
  ◉  Medium
  ○  Hard

Category

┌──────────────────────────┐
│ Entertainment: Books   ▼ │
└──────────────────────────┘
┆  Geography               ┆
┆- - - - - - - - - - - - - ┆
┆  Science                 ┆
└──────────────────────────┘
```

Widget Screen

User can have a widget to see the score of the last quiz played.

```
Your last score was:
       10 / 10
```

Key Considerations

- App will be written solely in the Java Programming Language.
- App will utilize stable release versions for the libraries used, and also for,
    - Android Studio: version 3.3
    - Gradle: version 4.10.1
- App will keep all strings in a strings.xml file and enable RTL layout switching on all layout.
- App will include support for accessibility, which includes content descriptions and navigation using a D-pad.

How will your app handle data persistence?

The app will fetch the questions list from https://opentdb.com when the user clicks "Play Quiz" on the Main Screen using an AsyncTask.
These list of questions will be saved in SQLite Database using Room.

The settings opted by user, i.e. Number of questions, Category & Difficulty will be saved in preferences.

Describe any edge or corner cases in the UX.

If the network operation to fetch the quiz questions fails, then a Snackbar will be used to indicate the error condition to the user.

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

Include ButterKnife library (version 10.1.0) for view injection.

Describe how you will implement Google Play Services or other external services.

1. AdMob: To display ads on the main screen.
2. Analytics: To get deeper understanding of how users are using the app, and how many quiz rounds are played.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Create new Android project named QuizApp.
- Configure the required library dependencies in build.gradle

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for QuizActivity with ViewPager.
- Build UI for QuestionFragment to be displayed in ViewPager.
- Build UI for ResultActivity
- Build UI for SettingsActivity.

Task 3: Fetch Questions from https://opentdb.com

- Implement AsyncTask to fetch questions from https://opentdb.com
- Parse the JSON content fetched by the AsyncTask.

Task 4: Implement Room, LiveData & ViewModel for the list of questions

- Create an Entity for Question class to represent a table in the database.
- Create QuestionDatabase (extends from RoomDatabase) which serves as the access point for the list of questions persisted by the app.
- Create the QuestionDao interface to specify methods used for specifying the database.
- Implement LiveData to observe changes to the database
- Implement ViewModel to store and manage UI-related data in a lifecycle conscious way.

Task 5: Application logic

- Validating the user's selected answer against the correct answer.
- Getting the result score for the user at the end of the quiz.

Task 6: Build App Widget

- Implement UI for widget.
- Implement support for the widget to display the score from the last quiz played by the user.

Task 7: Implement GooglePlay services

- Incorporate AdMob service to display ads on the MainActivity in QuizApp.
- Incorporate Analytics service to get analytics information. Implement ONLY one instance of Analytics.