GitHub Username: madfree

Qwizard

Description

Want to test your knowledge about common – and not so common trivia?
Qwizard is a quiz game that tests your knowledge in different categories and different difficulty levels. See, how good you are, compared to others. Share your knowledge providing new trivia.

Find out, how much you know – play Qwizard.

Intended User

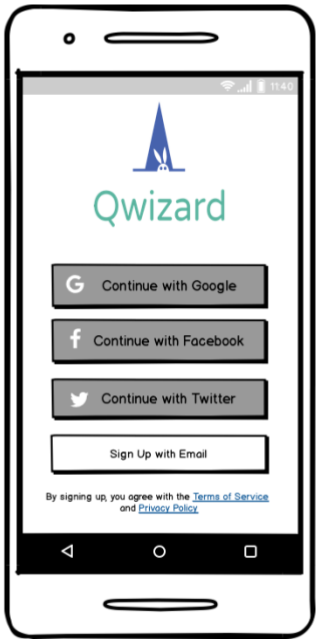
Anyone who is looking for some distraction and wants to learn facts in a playful way. The language of the app is English, to target as many users as possible.

Features

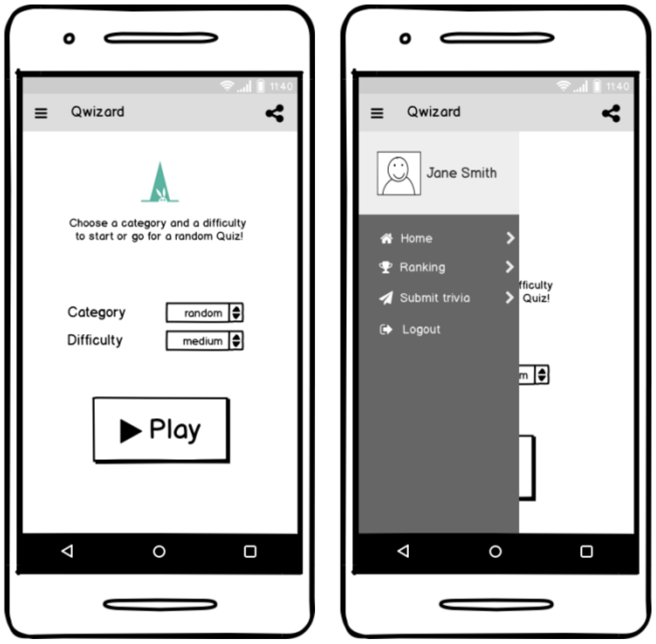
1. Take quizzes in different categories
2. Take quizzes on different levels of difficulty
3. Compare your results to others
4. Add new trivia questions to a growing database

User Interface Mocks

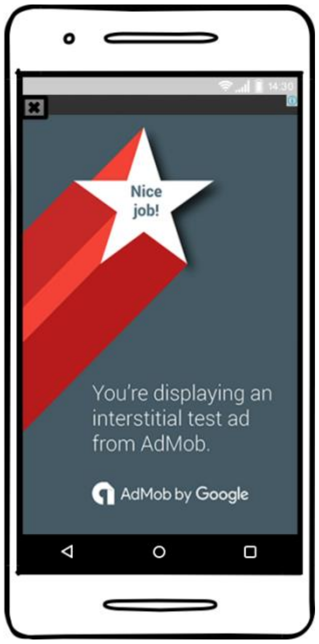
Screen 1 - Signup / Login



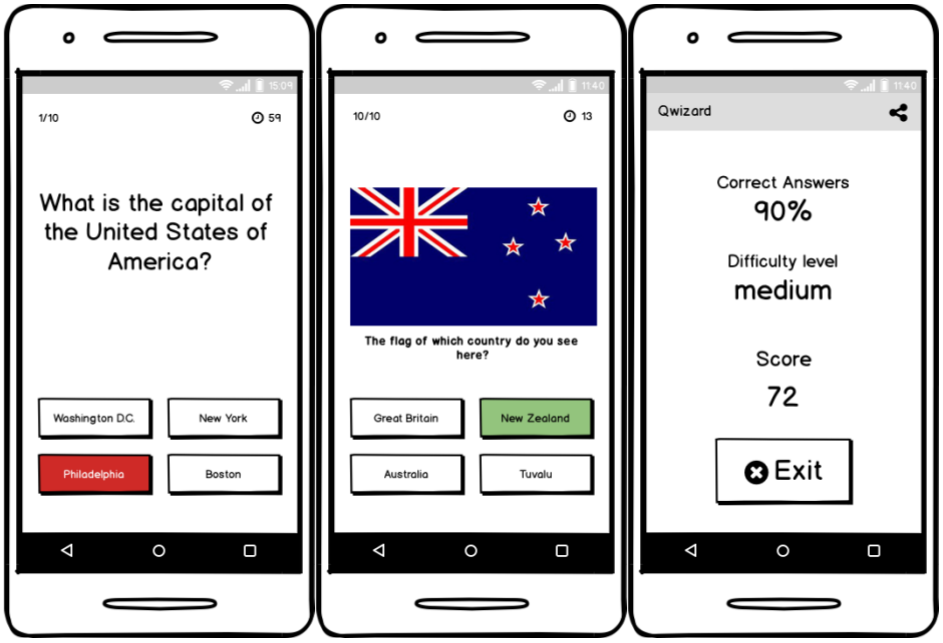
Screen 2: Home Screen



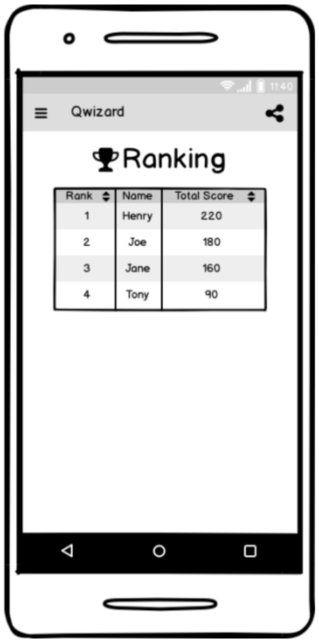
Screen 3: Intersititial Ad



Screen 4-6

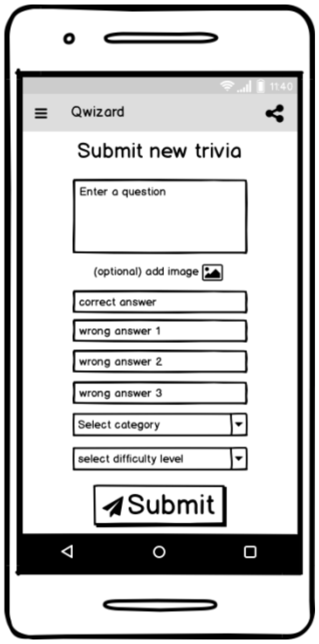


Screen 7: Ranking

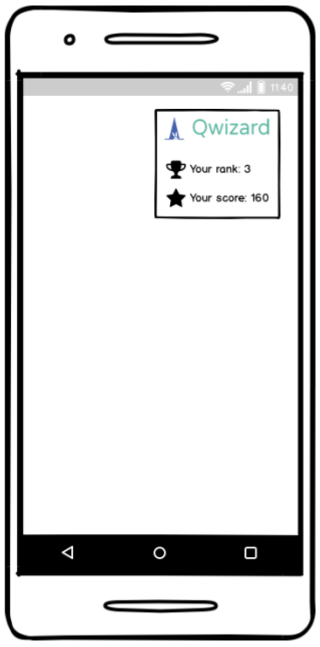


Rank	Name	Total Score
1	Henry	220
2	Joe	180
3	Jane	160
4	Tony	90

Screen 8: Submit new trivia



Screen 9: Widget



Key Considerations

How will your app handle data persistence?

Qwizard uses Firebase Realtime Database and Firebase Storage to handle the trivia data.

Describe any edge or corner cases in the UX.

This app is intended to make heavy use of Firebase / Google Services and is solely written in Java.

The user will login and authenticate upon app start via Google, Facebook, Twitter or his email address to create a user/player account. Firebase Authentication / Firebase UI will be used for it. The data will be set to persist, so the user can access the app, even when he/she is offline.

At the home screen the user can start a quiz either choosing a category and difficulty or choosing “random” to be surprised. A quiz of 10 trivia will be fetched from the Firebase Realtime Database / Firebase Storage (for images) according to the user’s choice. In case the user is offline or the connection is bad, an error message will be shown instead of starting the quiz.

A burger menu icon in the app bar provides access to the other menu options. The user can navigate to the ranking, submit new trivia or log out (disconnect his account).

Before the start of each quiz a google admob interstitial will be shown for monetization purposes.
For the quiz, the player has a total of one minute to answer 10 questions, showing either a question or an image with a corresponding question. The player has to choose one answer out of four shown. At the end of the quiz a result screen is showing, calculating the score, based on percentage of correct answers and difficulty level.

The app will provide a widget showing the user's current rank and score. **To update this Widget asynchronously, WorkManager will be used to run scheduled synchronization, as this is now the recommended way according to documentation, since JobDispatcher and IntentService are deprecated.**

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

The app will be built with Android Studios most recent (as of May 2020) version 3.6.3 and the most recent gradle version 5.6.4

- Firebase UI for authentication / login (version 6.2.0)
- Picasso or Glide to handle saving and loading images (Picasso version 2.71828 or Glide version 4.11.0)
- Timber for logging (version 4.7.1)
- Material Components, if needed (version 1.1.0)
- Dagger 2, if needed (version 2.27)
- WorkManager, for asynchronous tasks (version 2.3.4) → since JobDispatcher or IntentService are deprecated

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

- Firebase Authentication for user sign up / log in (version 19.3.1)
- Firebase Realtime Database to handle the trivia data (version 19.3.0)
- Firebase Storage to handle images for the trivia (version 19.1.1)
- Firebase Cloud Messaging for notifications (version 20.1.6)
- Firebase Analytics for insights into app usage (version 17.4.0)
- Google AdMob for monetization (version 19.1.0)

Next Steps: Required Tasks

1. Set up empty Android project and most recent gradle dependencies / Androidx
2. Set up firebase project and connect to it

3. Implement Firebase Authentication and SignUp-Activity
4. Build MainActivity UI: Implement app bar with menu, button, spinner for categories and difficulties
5. Set up trivia pojo / model
6. Build the QuizActivity holding the list of trivias and logic for the quiz
7. Set up security rules for Realtime Database
8. Build the RankingActivity, to show the list of players and scores
9. Set Up Firebase Storage and connect app
10. Add Firebase Storage security rules
11. Build the AddNewTriviaActivity to submit new questions and answers and test it
13. Connect firebase cloud messaging and implement notification (daily reminder)
14. Build the widget

Task 1: Android Project Setup

- Create an new android project, choosing the “empty activity” template
- Connect the project to the Capstone Github repository
- Update gradle and basic dependencies to newest versions
- Add dependencies for Timber, Firebase Authentication, Firebase UI , Firebase Analytics, Firebase Realtime Database, Firebase Storage, Firebase Cloud Messaging, Google Play Services (Mobile Ads SDK)

Task 2: Set up Firebase project

- Create new Firebase project via the Firebase console
- Add Analytics to project within the creation process
- Link the Firebase project with the Qwizard app
    - Get the debug signing certificate (SHA-1)
    - Download the configuration file (google-services.json)
    - Add configuration file to the app
- Run the app to test if the connection is established

Task 3: Implement Authentication

- Build user authentication via Firebase Authentication using Firebase UI

Task 4: Build Home Screen

- Build UI for MainActivity / HomeScreen
- Add Drawer Menu

- Add a share button

Task 4: Add functionality to submit trivia

- Build UI for the SubmitTriviaActivity, to submit the questions and answers
- Create a trivia class that holds/parses the trivia data
- Update the security rules for the Firebase Realtime Database and Firebase Storage
- Write methods for pushing data to the Firebase Realtime Database and images to Firebase Storage
- Test functionality and add a core stack of trivia to the database

Task 5: Build Quiz UI and functionality

- Build UI for the individual quiz questions and answers within the quiz
- Build methods to get trivia from the database according to chosen category and difficulty, run the quiz and show a result screen
- Implement Google Admob interstitial (test) ad

Task 5: Build Ranking UI and widget

- Build UI for the RankingActivity showing the placement and scores of the players
- Add Widget, which displays user's score and rank

Task 6: Test and polish

- Testing all functionalities
- Optimize UI according to material design guidelines
- Add transitions
- Provide RTL-layout support
- Move all remaining strings to strings.xml
- Add missing content descriptions
- Add an app icon

Task 7: Generate and Deploy

- Create Google Play Account
- Generate APK with signing configuration, keystore and password
- Create Google Play image and text
- Push the APK on the store